RISC-V OpenSBI

v1.6

Generated by Doxygen 1.9.8

1 RISC-V Open Source Supervisor Binary Interface (OpenSBI) 1

2 OpenSBI Contribution Guideline 7

3 OpenSBI Platform Support Guideline 9

4 OpenSBI Platform Requirements 11

5 OpenSBI Library Usage 13

6 OpenSBI Domain Support 15

7 OpenSBI Device Tree Configuration Guideline 21

8 Writing tests for OpenSBI 23

9 OpenSBI Platform Firmwares 25

10 OpenSBI Firmware with Dynamic Information (FW_DYNAMIC) 27

11 OpenSBI Firmware with Jump Address (FW_JUMP) 29

12 OpenSBI Firmware with Payload (FW_PAYLOAD) 31

13 Linux as a direct payload to OpenSBI 33

14 U-Boot as a payload to OpenSBI 35

15 Andes AE350 SoC Platform 37

16 Ariane FPGA SoC Platform 41

17 OpenPiton FPGA SoC Platform 43

18 Generic Platform 45

19 Nuclei UX600 Platform 47

20 OpenSBI Supported Platforms 49

21 QEMU RISC-V Virt Machine Platform 51

22 Renesas RZ/Five SoC (R9A07G043F) Platform 53

23 Shakti C-class SoC Platform 57

24 SiFive FU540 SoC Platform 59

25 Spike Simulator Platform 63

26 T-HEAD C9xx Series Processors 65

27 Data Structure Index 67
27.1 Data Structures . . . 67

28 File Index 71
28.1 File List . . . 71

29 Data Structure Documentation 77
29.1 __aligned Struct Reference . . . 77
29.1.1 Field Documentation . . . 77
29.1.1.1 next . . . 77
29.1.1.2 owner . . . 77
29.2 aclint_mswi_data Struct Reference . . . 77
29.2.1 Field Documentation . . . 77
29.2.1.1 addr . . . 77
29.2.1.2 first_hartid . . . 77
29.2.1.3 hart_count . . . 78
29.2.1.4 size . . . 78
29.3 aclint_mtimer_data Struct Reference . . . 78
29.3.1 Field Documentation . . . 78
29.3.1.1 first_hartid . . . 78
29.3.1.2 hart_count . . . 78
29.3.1.3 has_64bit_mmio . . . 78
29.3.1.4 has_shared_mtime . . . 78
29.3.1.5 mtime_addr . . . 78
29.3.1.6 mtime_freq . . . 78
29.3.1.7 mtime_size . . . 78
29.3.1.8 mtimecmp_addr . . . 79
29.3.1.9 mtimecmp_size . . . 79
29.3.1.10 time_delta_computed . . . 79
29.3.1.11 time_delta_reference . . . 79
29.3.1.12 time_rd . . . 79
29.3.1.13 time_wr . . . 79
29.4 aplic_data Struct Reference . . . 79
29.4.1 Field Documentation . . . 79
29.4.1.1 addr . . . 79
29.4.1.2 delegate . . . 79
29.4.1.3 has_msicfg_mmode . . . 79
29.4.1.4 has_msicfg_smode . . . 80
29.4.1.5 irqchip . . . 80
29.4.1.6 msicfg_mmode . . . 80
29.4.1.7 msicfg_smode . . . 80
29.4.1.8 num_idc . . . 80
29.4.1.9 num_source . . . 80

29.4.1.10 size . . . 80
29.4.1.11 targets_mmode . . . 80
29.5 aplic_delegate_data Struct Reference . . . 80
29.5.1 Field Documentation . . . 80
29.5.1.1 child_index . . . 80
29.5.1.2 first_irq . . . 80
29.5.1.3 last_irq . . . 80
29.6 aplic_msicfg_data Struct Reference . . . 81
29.6.1 Field Documentation . . . 81
29.6.1.1 base_addr . . . 81
29.6.1.2 hhxs . . . 81
29.6.1.3 hhxw . . . 81
29.6.1.4 lhxs . . . 81
29.6.1.5 lhxw . . . 81
29.7 atomic_t Struct Reference . . . 81
29.7.1 Field Documentation . . . 81
29.7.1.1 counter . . . 81
29.8 dw_i2c_adapter Struct Reference . . . 81
29.8.1 Field Documentation . . . 82
29.8.1.1 adapter . . . 82
29.8.1.2 addr . . . 82
29.9 fdt_driver Struct Reference . . . 82
29.9.1 Field Documentation . . . 82
29.9.1.1 experimental . . . 82
29.9.1.2 init . . . 82
29.9.1.3 match_table . . . 82
29.10 fdt_general_fixup Struct Reference . . . 82
29.10.1 Detailed Description . . . 82
29.10.2 Field Documentation . . . 83
29.10.2.1 do_fixup . . . 83
29.10.2.2 head . . . 83
29.10.2.3 name . . . 83
29.11 fdt_gpio Struct Reference . . . 83
29.11.1 Detailed Description . . . 83
29.11.2 Field Documentation . . . 83
29.11.2.1 driver . . . 83
29.11.2.2 xlate . . . 83
29.12 fdt_header Struct Reference . . . 83
29.12.1 Field Documentation . . . 84
29.12.1.1 boot_cpuid_phys . . . 84
29.12.1.2 last_comp_version . . . 84
29.12.1.3 magic . . . 84

29.12.1.4 off_dt_strings . . . 84
29.12.1.5 off_dt_struct . . . 84
29.12.1.6 off_mem_rsvmap . . . 84
29.12.1.7 size_dt_strings . . . 84
29.12.1.8 size_dt_struct . . . 84
29.12.1.9 totalsize . . . 84
29.12.1.10 version . . . 84
29.13 fdt_mailbox Struct Reference . . . 84
29.13.1 Detailed Description . . . 84
29.13.2 Field Documentation . . . 85
29.13.2.1 driver . . . 85
29.13.2.2 xlate . . . 85
29.14 fdt_match Struct Reference . . . 85
29.14.1 Field Documentation . . . 85
29.14.1.1 compatible . . . 85
29.14.1.2 data . . . 85
29.15 fdt_node_header Struct Reference . . . 85
29.15.1 Field Documentation . . . 85
29.15.1.1 name . . . 85
29.15.1.2 tag . . . 85
29.16 fdt_phandle_args Struct Reference . . . 86
29.16.1 Field Documentation . . . 86
29.16.1.1 args . . . 86
29.16.1.2 args_count . . . 86
29.16.1.3 node_offset . . . 86
29.17 fdt_pmu_hw_event_counter_map Struct Reference . . . 86
29.17.1 Field Documentation . . . 86
29.17.1.1 ctr_map . . . 86
29.17.1.2 eidx_end . . . 86
29.17.1.3 eidx_start . . . 86
29.18 fdt_pmu_hw_event_select_map Struct Reference . . . 86
29.18.1 Field Documentation . . . 87
29.18.1.1 eidx . . . 87
29.18.1.2 select . . . 87
29.19 fdt_pmu_raw_event_counter_map Struct Reference . . . 87
29.19.1 Field Documentation . . . 87
29.19.1.1 ctr_map . . . 87
29.19.1.2 select . . . 87
29.19.1.3 select_mask . . . 87
29.20 fdt_property Struct Reference . . . 87
29.20.1 Field Documentation . . . 87
29.20.1.1 data . . . 87

29.20.1.2 len . . . 88
29.20.1.3 nameoff . . . 88
29.20.1.4 tag . . . 88
29.21 fdt_reserve_entry Struct Reference . . . 88
29.21.1 Field Documentation . . . 88
29.21.1.1 address . . . 88
29.21.1.2 size . . . 88
29.22 fw_dynamic_info Struct Reference . . . 88
29.22.1 Detailed Description . . . 88
29.22.2 Field Documentation . . . 89
29.22.2.1 boot_hart . . . 89
29.22.2.2 magic . . . 89
29.22.2.3 next_addr . . . 89
29.22.2.4 next_mode . . . 89
29.22.2.5 options . . . 89
29.22.2.6 version . . . 89
29.23 gpio_chip Struct Reference . . . 89
29.23.1 Detailed Description . . . 89
29.23.2 Field Documentation . . . 90
29.23.2.1 direction_input . . . 90
29.23.2.2 direction_output . . . 90
29.23.2.3 driver . . . 90
29.23.2.4 get . . . 90
29.23.2.5 get_direction . . . 90
29.23.2.6 id . . . 90
29.23.2.7 ngpio . . . 90
29.23.2.8 node . . . 90
29.23.2.9 set . . . 91
29.24 gpio_pin Struct Reference . . . 91
29.24.1 Detailed Description . . . 91
29.24.2 Field Documentation . . . 91
29.24.2.1 chip . . . 91
29.24.2.2 flags . . . 91
29.24.2.3 offset . . . 91
29.25 i2c_adapter Struct Reference . . . 91
29.25.1 Detailed Description . . . 91
29.25.2 Field Documentation . . . 92
29.25.2.1 id . . . 92
29.25.2.2 node . . . 92
29.25.2.3 read . . . 92
29.25.2.4 write . . . 92
29.26 imsic_data Struct Reference . . . 92

29.26.1 Field Documentation . . . 92
29.26.1.1 group_index_bits . . . 92
29.26.1.2 group_index_shift . . . 92
29.26.1.3 guest_index_bits . . . 93
29.26.1.4 hart_index_bits . . . 93
29.26.1.5 num_ids . . . 93
29.26.1.6 regs . . . 93
29.26.1.7 targets_mmode . . . 93
29.27 imsic_regs Struct Reference . . . 93
29.27.1 Field Documentation . . . 93
29.27.1.1 addr . . . 93
29.27.1.2 size . . . 93
29.28 mbox_chan Struct Reference . . . 93
29.28.1 Detailed Description . . . 93
29.28.2 Field Documentation . . . 94
29.28.2.1 chan_args . . . 94
29.28.2.2 mbox . . . 94
29.28.2.3 node . . . 94
29.29 mbox_controller Struct Reference . . . 94
29.29.1 Detailed Description . . . 94
29.29.2 Field Documentation . . . 94
29.29.2.1 chan_list . . . 94
29.29.2.2 driver . . . 94
29.29.2.3 free_chan . . . 94
29.29.2.4 get_attribute . . . 95
29.29.2.5 id . . . 95
29.29.2.6 max_xfer_len . . . 95
29.29.2.7 node . . . 95
29.29.2.8 request_chan . . . 95
29.29.2.9 set_attribute . . . 95
29.29.2.10 xfer . . . 95
29.29.2.11 xfer_next_seq . . . 95
29.30 mbox_xfer Struct Reference . . . 95
29.30.1 Detailed Description . . . 96
29.30.2 Field Documentation . . . 96
29.30.2.1 args . . . 96
29.30.2.2 flags . . . 96
29.30.2.3 rx . . . 96
29.30.2.4 rx_len . . . 96
29.30.2.5 rx_timeout . . . 96
29.30.2.6 seq . . . 96
29.30.2.7 tx . . . 96

29.30.2.8 tx_len . . . 96
29.30.2.9 tx_timeout . . . 96
29.31 platform_uart_data Struct Reference . . . 97
29.31.1 Field Documentation . . . 97
29.31.1.1 addr . . . 97
29.31.1.2 baud . . . 97
29.31.1.3 freq . . . 97
29.31.1.4 reg_io_width . . . 97
29.31.1.5 reg_offset . . . 97
29.31.1.6 reg_shift . . . 97
29.32 plic_data Struct Reference . . . 97
29.32.1 Field Documentation . . . 98
29.32.1.1 addr . . . 98
29.32.1.2 context_map . . . 98
29.32.1.3 flags . . . 98
29.32.1.4 irqchip . . . 98
29.32.1.5 num_src . . . 98
29.32.1.6 pm_data . . . 98
29.32.1.7 size . . . 98
29.33 plicsw_data Struct Reference . . . 98
29.33.1 Field Documentation . . . 98
29.33.1.1 addr . . . 98
29.33.1.2 hart_count . . . 98
29.33.1.3 size . . . 98
29.34 plmt_data Struct Reference . . . 99
29.34.1 Field Documentation . . . 99
29.34.1.1 hart_count . . . 99
29.34.1.2 size . . . 99
29.34.1.3 time_cmp . . . 99
29.34.1.4 time_val . . . 99
29.34.1.5 timer_freq . . . 99
29.35 regmap Struct Reference . . . 99
29.35.1 Detailed Description . . . 99
29.35.2 Field Documentation . . . 100
29.35.2.1 id . . . 100
29.35.2.2 node . . . 100
29.35.2.3 reg_base . . . 100
29.35.2.4 reg_max . . . 100
29.35.2.5 reg_read . . . 100
29.35.2.6 reg_shift . . . 100
29.35.2.7 reg_stride . . . 100
29.35.2.8 reg_update_bits . . . 100

29.35.2.9 reg_write . . . 100
29.36 rpmi_base_get_attributes_resp Struct Reference . . . 100
29.36.1 Field Documentation . . . 101
29.36.1.1 f0 . . . 101
29.36.1.2 f1 . . . 101
29.36.1.3 f2 . . . 101
29.36.1.4 f3 . . . 101
29.36.1.5 status_code . . . 101
29.37 rpmi_base_get_platform_info_resp Struct Reference . . . 101
29.37.1 Field Documentation . . . 101
29.37.1.1 plat_info . . . 101
29.37.1.2 plat_info_len . . . 101
29.37.1.3 status . . . 101
29.38 rpmi_clock_get_attributes_req Struct Reference . . . 101
29.38.1 Field Documentation . . . 102
29.38.1.1 clock_id . . . 102
29.39 rpmi_clock_get_attributes_resp Struct Reference . . . 102
29.39.1 Field Documentation . . . 102
29.39.1.1 flags . . . 102
29.39.1.2 name . . . 102
29.39.1.3 num_rates . . . 102
29.39.1.4 status . . . 102
29.39.1.5 transition_latency . . . 102
29.40 rpmi_clock_get_config_req Struct Reference . . . 102
29.40.1 Field Documentation . . . 102
29.40.1.1 clock_id . . . 102
29.41 rpmi_clock_get_config_resp Struct Reference . . . 103
29.41.1 Field Documentation . . . 103
29.41.1.1 config . . . 103
29.41.1.2 status . . . 103
29.42 rpmi_clock_get_num_clocks_resp Struct Reference . . . 103
29.42.1 Field Documentation . . . 103
29.42.1.1 num_clocks . . . 103
29.42.1.2 status . . . 103
29.43 rpmi_clock_get_rate_req Struct Reference . . . 103
29.43.1 Field Documentation . . . 103
29.43.1.1 clock_id . . . 103
29.44 rpmi_clock_get_rate_resp Struct Reference . . . 104
29.44.1 Field Documentation . . . 104
29.44.1.1 clock_rate_high . . . 104
29.44.1.2 clock_rate_low . . . 104
29.44.1.3 status . . . 104

29.45 rpmi_clock_get_supported_rates_req Struct Reference . . . 104
29.45.1 Field Documentation . . . 104
29.45.1.1 clock_id . . . 104
29.45.1.2 clock_rate_index . . . 104
29.46 rpmi_clock_get_supported_rates_resp Struct Reference . . . 104
29.46.1 Field Documentation . . . 105
29.46.1.1 clock_rate . . . 105
29.46.1.2 flags . . . 105
29.46.1.3 remaining . . . 105
29.46.1.4 returned . . . 105
29.46.1.5 status . . . 105
29.47 rpmi_clock_set_config_req Struct Reference . . . 105
29.47.1 Field Documentation . . . 105
29.47.1.1 clock_id . . . 105
29.47.1.2 config . . . 105
29.48 rpmi_clock_set_config_resp Struct Reference . . . 105
29.48.1 Field Documentation . . . 106
29.48.1.1 status . . . 106
29.49 rpmi_clock_set_rate_req Struct Reference . . . 106
29.49.1 Field Documentation . . . 106
29.49.1.1 clock_id . . . 106
29.49.1.2 clock_rate_high . . . 106
29.49.1.3 clock_rate_low . . . 106
29.49.1.4 flags . . . 106
29.50 rpmi_clock_set_rate_resp Struct Reference . . . 106
29.50.1 Field Documentation . . . 106
29.50.1.1 status . . . 106
29.51 rpmi_cppc_get_fastchan_offset_req Struct Reference . . . 107
29.51.1 Field Documentation . . . 107
29.51.1.1 hart_id . . . 107
29.52 rpmi_cppc_get_fastchan_offset_resp Struct Reference . . . 107
29.52.1 Field Documentation . . . 107
29.52.1.1 fc_perf_feedback_offset_hi . . . 107
29.52.1.2 fc_perf_feedback_offset_lo . . . 107
29.52.1.3 fc_perf_request_offset_hi . . . 107
29.52.1.4 fc_perf_request_offset_lo . . . 107
29.52.1.5 status . . . 107
29.53 rpmi_cppc_get_fastchan_region_resp Struct Reference . . . 107
29.53.1 Field Documentation . . . 108
29.53.1.1 db_addr_hi . . . 108
29.53.1.2 db_addr_lo . . . 108
29.53.1.3 db_preservemask_hi . . . 108

29.53.1.4 db_preservemask_lo . . . 108
29.53.1.5 db_setmask_hi . . . 108
29.53.1.6 db_setmask_lo . . . 108
29.53.1.7 flags . . . 108
29.53.1.8 region_addr_hi . . . 108
29.53.1.9 region_addr_lo . . . 108
29.53.1.10 region_size_hi . . . 108
29.53.1.11 region_size_lo . . . 108
29.53.1.12 status . . . 109
29.54 rpmi_cppc_hart_list_req Struct Reference . . . 109
29.54.1 Field Documentation . . . 109
29.54.1.1 start_index . . . 109
29.55 rpmi_cppc_hart_list_resp Struct Reference . . . 109
29.55.1 Field Documentation . . . 109
29.55.1.1 hartid . . . 109
29.55.1.2 remaining . . . 109
29.55.1.3 returned . . . 109
29.55.1.4 status . . . 109
29.56 rpmi_cppc_probe_req Struct Reference . . . 109
29.56.1 Field Documentation . . . 110
29.56.1.1 hart_id . . . 110
29.56.1.2 reg_id . . . 110
29.57 rpmi_cppc_probe_resp Struct Reference . . . 110
29.57.1 Field Documentation . . . 110
29.57.1.1 reg_len . . . 110
29.57.1.2 status . . . 110
29.58 rpmi_cppc_read_reg_req Struct Reference . . . 110
29.58.1 Field Documentation . . . 110
29.58.1.1 hart_id . . . 110
29.58.1.2 reg_id . . . 110
29.59 rpmi_cppc_read_reg_resp Struct Reference . . . 111
29.59.1 Field Documentation . . . 111
29.59.1.1 data_hi . . . 111
29.59.1.2 data_lo . . . 111
29.59.1.3 status . . . 111
29.60 rpmi_cppc_write_reg_req Struct Reference . . . 111
29.60.1 Field Documentation . . . 111
29.60.1.1 data_hi . . . 111
29.60.1.2 data_lo . . . 111
29.60.1.3 hart_id . . . 111
29.60.1.4 reg_id . . . 111
29.61 rpmi_cppc_write_reg_resp Struct Reference . . . 112

29.61.1 Field Documentation . . . 112
29.61.1.1 status . . . 112
29.62 rpmi_enable_notification_req Struct Reference . . . 112
29.62.1 Detailed Description . . . 112
29.62.2 Field Documentation . . . 112
29.62.2.1 eventid . . . 112
29.63 rpmi_enable_notification_resp Struct Reference . . . 112
29.63.1 Detailed Description . . . 112
29.63.2 Field Documentation . . . 112
29.63.2.1 status . . . 112
29.64 rpmi_hsm_get_hart_list_req Struct Reference . . . 113
29.64.1 Field Documentation . . . 113
29.64.1.1 start_index . . . 113
29.65 rpmi_hsm_get_hart_list_resp Struct Reference . . . 113
29.65.1 Field Documentation . . . 113
29.65.1.1 hartid . . . 113
29.65.1.2 remaining . . . 113
29.65.1.3 returned . . . 113
29.65.1.4 status . . . 113
29.66 rpmi_hsm_get_hart_status_req Struct Reference . . . 113
29.66.1 Field Documentation . . . 114
29.66.1.1 hartid . . . 114
29.67 rpmi_hsm_get_hart_status_resp Struct Reference . . . 114
29.67.1 Field Documentation . . . 114
29.67.1.1 hart_status . . . 114
29.67.1.2 status . . . 114
29.68 rpmi_hsm_get_susp_info_req Struct Reference . . . 114
29.68.1 Field Documentation . . . 114
29.68.1.1 suspend_type . . . 114
29.69 rpmi_hsm_get_susp_info_resp Struct Reference . . . 114
29.69.1 Field Documentation . . . 115
29.69.1.1 entry_latency_us . . . 115
29.69.1.2 exit_latency_us . . . 115
29.69.1.3 flags . . . 115
29.69.1.4 min_residency_us . . . 115
29.69.1.5 status . . . 115
29.69.1.6 wakeup_latency_us . . . 115
29.70 rpmi_hsm_get_susp_types_req Struct Reference . . . 115
29.70.1 Field Documentation . . . 115
29.70.1.1 start_index . . . 115
29.71 rpmi_hsm_get_susp_types_resp Struct Reference . . . 115
29.71.1 Field Documentation . . . 116

29.71.1.1 remaining . . . 116
29.71.1.2 returned . . . 116
29.71.1.3 status . . . 116
29.71.1.4 types . . . 116
29.72 rpmi_hsm_hart_start_req Struct Reference . . . 116
29.72.1 Field Documentation . . . 116
29.72.1.1 hartid . . . 116
29.72.1.2 start_addr_hi . . . 116
29.72.1.3 start_addr_lo . . . 116
29.73 rpmi_hsm_hart_start_resp Struct Reference . . . 116
29.73.1 Field Documentation . . . 117
29.73.1.1 status . . . 117
29.74 rpmi_hsm_hart_stop_req Struct Reference . . . 117
29.74.1 Field Documentation . . . 117
29.74.1.1 hartid . . . 117
29.75 rpmi_hsm_hart_stop_resp Struct Reference . . . 117
29.75.1 Field Documentation . . . 117
29.75.1.1 status . . . 117
29.76 rpmi_hsm_hart_susp_req Struct Reference . . . 117
29.76.1 Field Documentation . . . 117
29.76.1.1 hartid . . . 117
29.76.1.2 resume_addr_hi . . . 118
29.76.1.3 resume_addr_lo . . . 118
29.76.1.4 suspend_type . . . 118
29.77 rpmi_hsm_hart_susp_resp Struct Reference . . . 118
29.77.1 Field Documentation . . . 118
29.77.1.1 status . . . 118
29.78 rpmi_message Struct Reference . . . 118
29.78.1 Detailed Description . . . 118
29.78.2 Field Documentation . . . 118
29.78.2.1 data . . . 118
29.78.2.2 header . . . 118
29.79 rpmi_message_args Struct Reference . . . 119
29.79.1 Detailed Description . . . 119
29.79.2 Field Documentation . . . 119
29.79.2.1 flags . . . 119
29.79.2.2 rx_data_len . . . 119
29.79.2.3 rx_endian_words . . . 119
29.79.2.4 rx_token . . . 119
29.79.2.5 service_id . . . 119
29.79.2.6 tx_endian_words . . . 119
29.79.2.7 type . . . 119

29.80 rpmi_message_header Struct Reference . . . 119
29.80.1 Detailed Description . . . 120
29.80.2 Field Documentation . . . 120
29.80.2.1 datalen . . . 120
29.80.2.2 flags . . . 120
29.80.2.3 service_id . . . 120
29.80.2.4 servicegroup_id . . . 120
29.80.2.5 token . . . 120
29.81 rpmi_sysrst_get_reset_attributes_resp Struct Reference . . . 120
29.81.1 Detailed Description . . . 120
29.81.2 Field Documentation . . . 120
29.81.2.1 flags . . . 120
29.81.2.2 status . . . 121
29.82 rpmi_syssusp_get_attr_req Struct Reference . . . 121
29.82.1 Detailed Description . . . 121
29.82.2 Field Documentation . . . 121
29.82.2.1 susp_type . . . 121
29.83 rpmi_syssusp_get_attr_resp Struct Reference . . . 121
29.83.1 Detailed Description . . . 121
29.83.2 Field Documentation . . . 121
29.83.2.1 flags . . . 121
29.83.2.2 status . . . 121
29.84 rpmi_syssusp_suspend_req Struct Reference . . . 121
29.84.1 Field Documentation . . . 122
29.84.1.1 hartid . . . 122
29.84.1.2 resume_addr_hi . . . 122
29.84.1.3 resume_addr_lo . . . 122
29.84.1.4 suspend_type . . . 122
29.85 rpmi_syssusp_suspend_resp Struct Reference . . . 122
29.85.1 Field Documentation . . . 122
29.85.1.1 status . . . 122
29.86 sbi_console_device Struct Reference . . . 122
29.86.1 Field Documentation . . . 123
29.86.1.1 console_getc . . . 123
29.86.1.2 console_putc . . . 123
29.86.1.3 console_puts . . . 123
29.86.1.4 name . . . 123
29.87 sbi_cppc_device Struct Reference . . . 123
29.87.1 Detailed Description . . . 123
29.87.2 Field Documentation . . . 123
29.87.2.1 cppc_probe . . . 123
29.87.2.2 cppc_read . . . 123

29.87.2.3 cppc_write . . . 123
29.87.2.4 name . . . 124
29.88 sbi_cpu_idle_state Struct Reference . . . 124
29.88.1 Field Documentation . . . 124
29.88.1.1 entry_latency_us . . . 124
29.88.1.2 exit_latency_us . . . 124
29.88.1.3 local_timer_stop . . . 124
29.88.1.4 min_residency_us . . . 124
29.88.1.5 name . . . 124
29.88.1.6 suspend_param . . . 124
29.88.1.7 wakeup_latency_us . . . 124
29.89 sbi_dbtr_data_msg Struct Reference . . . 124
29.89.1 Field Documentation . . . 125
29.89.1.1 tdata1 . . . 125
29.89.1.2 tdata2 . . . 125
29.89.1.3 tdata3 . . . 125
29.89.1.4 tstate . . . 125
29.90 sbi_dbtr_hart_triggers_state Struct Reference . . . 125
29.90.1 Field Documentation . . . 125
29.90.1.1 available_trigs . . . 125
29.90.1.2 hartid . . . 125
29.90.1.3 probed . . . 125
29.90.1.4 shmem . . . 125
29.90.1.5 total_trigs . . . 126
29.90.1.6 triggers . . . 126
29.91 sbi_dbtr_id_msg Struct Reference . . . 126
29.91.1 Field Documentation . . . 126
29.91.1.1 idx . . . 126
29.92 sbi_dbtr_shmem Struct Reference . . . 126
29.92.1 Field Documentation . . . 126
29.92.1.1 phys_hi . . . 126
29.92.1.2 phys_lo . . . 126
29.93 sbi_dbtr_shmem_entry Struct Reference . . . 126
29.93.1 Detailed Description . . . 127
29.93.2 Field Documentation . . . 127
29.93.2.1 data . . . 127
29.93.2.2 id . . . 127
29.94 sbi_dbtr_trigger Struct Reference . . . 127
29.94.1 Field Documentation . . . 127
29.94.1.1 index . . . 127
29.94.1.2 state . . . 127
29.94.1.3 tdata1 . . . 127

29.94.1.4 tdata2 . . . 127
29.94.1.5 tdata3 . . . 127
29.94.1.6 type_mask . . . 127
29.95 sbi_dlist Struct Reference . . . 128
29.95.1 Field Documentation . . . 128
29.95.1.1 next . . . 128
29.95.1.2 prev . . . 128
29.96 sbi_domain Struct Reference . . . 128
29.96.1 Detailed Description . . . 128
29.96.2 Field Documentation . . . 128
29.96.2.1 assigned_harts . . . 128
29.96.2.2 assigned_harts_lock . . . 128
29.96.2.3 boot_hartid . . . 129
29.96.2.4 data_priv . . . 129
29.96.2.5 fw_region_inited . . . 129
29.96.2.6 index . . . 129
29.96.2.7 name . . . 129
29.96.2.8 next_addr . . . 129
29.96.2.9 next_arg1 . . . 129
29.96.2.10 next_mode . . . 129
29.96.2.11 node . . . 129
29.96.2.12 possible_harts . . . 129
29.96.2.13 regions . . . 129
29.96.2.14 system_reset_allowed . . . 129
29.96.2.15 system_suspend_allowed . . . 130
29.97 sbi_domain_data Struct Reference . . . 130
29.97.1 Detailed Description . . . 130
29.97.2 Field Documentation . . . 130
29.97.2.1 data_cleanup . . . 130
29.97.2.2 data_idx . . . 130
29.97.2.3 data_setup . . . 130
29.97.2.4 data_size . . . 130
29.97.2.5 head . . . 130
29.98 sbi_domain_data_priv Struct Reference . . . 131
29.98.1 Detailed Description . . . 131
29.98.2 Field Documentation . . . 131
29.98.2.1 idx_to_data_ptr . . . 131
29.99 sbi_domain_memregion Struct Reference . . . 131
29.99.1 Detailed Description . . . 131
29.99.2 Field Documentation . . . 131
29.99.2.1 base . . . 131
29.99.2.2 flags . . . 131

29.99.2.3 order . . . 131
29.100 sbi_ecall_extension Struct Reference . . . 131
29.100.1 Field Documentation . . . 132
29.100.1.1 experimental . . . 132
29.100.1.2 extid_end . . . 132
29.100.1.3 extid_start . . . 132
29.100.1.4 handle . . . 132
29.100.1.5 head . . . 132
29.100.1.6 name . . . 132
29.100.1.7 probe . . . 132
29.100.1.8 register_extensions . . . 132
29.101 sbi_ecall_return Struct Reference . . . 132
29.101.1 Field Documentation . . . 133
29.101.1.1 skip_regs_update . . . 133
29.101.1.2 value . . . 133
29.102 sbi_fifo Struct Reference . . . 133
29.102.1 Field Documentation . . . 133
29.102.1.1 avail . . . 133
29.102.1.2 entry_size . . . 133
29.102.1.3 num_entries . . . 133
29.102.1.4 qlock . . . 133
29.102.1.5 queue . . . 133
29.102.1.6 tail . . . 133
29.103 sbi_hart_ext_data Struct Reference . . . 133
29.103.1 Field Documentation . . . 134
29.103.1.1 id . . . 134
29.103.1.2 name . . . 134
29.104 sbi_hart_features Struct Reference . . . 134
29.104.1 Field Documentation . . . 134
29.104.1.1 detected . . . 134
29.104.1.2 extensions . . . 134
29.104.1.3 mhpm_bits . . . 134
29.104.1.4 mhpm_mask . . . 134
29.104.1.5 pmp_addr_bits . . . 134
29.104.1.6 pmp_count . . . 134
29.104.1.7 pmp_log2gran . . . 135
29.104.1.8 priv_version . . . 135
29.105 sbi_hartmask Struct Reference . . . 135
29.105.1 Detailed Description . . . 135
29.105.2 Member Function Documentation . . . 135
29.105.2.1 DECLARE_BITMAP() . . . 135
29.106 sbi_hsm_device Struct Reference . . . 135

29.106.1 Detailed Description . . . 135
29.106.2 Field Documentation . . . 135
29.106.2.1 hart_resume . . . 135
29.106.2.2 hart_start . . . 136
29.106.2.3 hart_stop . . . 136
29.106.2.4 hart_suspend . . . 136
29.106.2.5 name . . . 136
29.107 sbi_ipi_device Struct Reference . . . 136
29.107.1 Detailed Description . . . 136
29.107.2 Field Documentation . . . 136
29.107.2.1 ipi_clear . . . 136
29.107.2.2 ipi_send . . . 136
29.107.2.3 name . . . 137
29.108 sbi_ipi_event_ops Struct Reference . . . 137
29.108.1 Detailed Description . . . 137
29.108.2 Field Documentation . . . 137
29.108.2.1 name . . . 137
29.108.2.2 process . . . 137
29.108.2.3 sync . . . 137
29.108.2.4 update . . . 137
29.109 sbi_irqchip_device Struct Reference . . . 138
29.109.1 Detailed Description . . . 138
29.109.2 Field Documentation . . . 138
29.109.2.1 irq_handle . . . 138
29.109.2.2 node . . . 138
29.109.2.3 warm_init . . . 138
29.110 sbi_ldst_data Union Reference . . . 138
29.110.1 Field Documentation . . . 138
29.110.1.1 data_bytes . . . 138
29.110.1.2 data_u32 . . . 138
29.110.1.3 data_u64 . . . 138
29.110.1.4 data_ulong . . . 139
29.111 sbi_mpxy_channel Struct Reference . . . 139
29.111.1 Detailed Description . . . 139
29.111.2 Field Documentation . . . 139
29.111.2.1 attrs . . . 139
29.111.2.2 channel_id . . . 139
29.111.2.3 get_notification_events . . . 139
29.111.2.4 head . . . 139
29.111.2.5 read_attributes . . . 139
29.111.2.6 send_message_with_response . . . 140
29.111.2.7 send_message_without_response . . . 140

29.111.2.8 switch_eventsstate . . . 140
29.111.2.9 write_attributes . . . 140
29.112 sbi_mpxy_channel_attrs Struct Reference . . . 140
29.112.1 Detailed Description . . . 140
29.112.2 Field Documentation . . . 140
29.112.2.1 capability . . . 140
29.112.2.2 eventsstate_ctrl . . . 140
29.112.2.3 msg_completion_timeout . . . 141
29.112.2.4 msg_data_maxlen . . . 141
29.112.2.5 msg_proto_id . . . 141
29.112.2.6 msg_proto_version . . . 141
29.112.2.7 msg_send_timeout . . . 141
29.112.2.8 msi_control . . . 141
29.112.2.9 msi_info . . . 141
29.112.2.10 sse_event_id . . . 141
29.113 sbi_mpxy_msi_info Struct Reference . . . 141
29.113.1 Field Documentation . . . 141
29.113.1.1 msi_addr_hi . . . 141
29.113.1.2 msi_addr_lo . . . 141
29.113.1.3 msi_data . . . 141
29.114 sbi_platform Struct Reference . . . 142
29.114.1 Detailed Description . . . 142
29.114.2 Field Documentation . . . 142
29.114.2.1 features . . . 142
29.114.2.2 firmware_context . . . 142
29.114.2.3 hart_count . . . 142
29.114.2.4 hart_index2id . . . 142
29.114.2.5 hart_stack_size . . . 142
29.114.2.6 heap_size . . . 143
29.114.2.7 name . . . 143
29.114.2.8 opensbi_version . . . 143
29.114.2.9 platform_ops_addr . . . 143
29.114.2.10 platform_version . . . 143
29.114.2.11 reserved . . . 143
29.115 sbi_platform_operations Struct Reference . . . 143
29.115.1 Detailed Description . . . 144
29.115.2 Field Documentation . . . 144
29.115.2.1 cold_boot_allowed . . . 144
29.115.2.2 domains_init . . . 144
29.115.2.3 early_exit . . . 144
29.115.2.4 early_init . . . 144
29.115.2.5 emulate_load . . . 144

29.115.2.6 emulate_store . . . 144
29.115.2.7 extensions_init . . . 144
29.115.2.8 final_exit . . . 144
29.115.2.9 final_init . . . 144
29.115.2.10 get_tlb_num_entries . . . 145
29.115.2.11 get_tlbr_flush_limit . . . 145
29.115.2.12 ipi_init . . . 145
29.115.2.13 irqchip_init . . . 145
29.115.2.14 misa_check_extension . . . 145
29.115.2.15 misa_get_xlen . . . 145
29.115.2.16 mpxy_init . . . 145
29.115.2.17 nascent_init . . . 145
29.115.2.18 pmu_init . . . 145
29.115.2.19 pmu_xlate_to_mhpmevent . . . 145
29.115.2.20 timer_init . . . 145
29.115.2.21 vendor_ext_check . . . 145
29.115.2.22 vendor_ext_provider . . . 146
29.116 sbi_pmu_device Struct Reference . . . 146
29.116.1 Field Documentation . . . 146
29.116.1.1 fw_counter_match_encoding . . . 146
29.116.1.2 fw_counter_read_value . . . 146
29.116.1.3 fw_counter_start . . . 146
29.116.1.4 fw_counter_stop . . . 146
29.116.1.5 fw_counter_width . . . 146
29.116.1.6 fw_counter_write_value . . . 147
29.116.1.7 fw_event_validate_encoding . . . 147
29.116.1.8 hw_counter_disable_irq . . . 147
29.116.1.9 hw_counter_enable_irq . . . 147
29.116.1.10 hw_counter_filter_mode . . . 147
29.116.1.11 hw_counter_irq_bit . . . 147
29.116.1.12 name . . . 147
29.117 sbi_pmu_event_info Struct Reference . . . 147
29.117.1 Field Documentation . . . 147
29.117.1.1 event_data . . . 147
29.117.1.2 event_idx . . . 147
29.117.1.3 output . . . 148
29.118 sbi_scratch Struct Reference . . . 148
29.118.1 Detailed Description . . . 148
29.118.2 Field Documentation . . . 148
29.118.2.1 fw_heap_offset . . . 148
29.118.2.2 fw_heap_size . . . 148
29.118.2.3 fw_rw_offset . . . 148

29.118.2.4 fw_size . . . 148
29.118.2.5 fw_start . . . 148
29.118.2.6 hartid_to_scratch . . . 149
29.118.2.7 hartindex . . . 149
29.118.2.8 next_addr . . . 149
29.118.2.9 next_arg1 . . . 149
29.118.2.10 next_mode . . . 149
29.118.2.11 options . . . 149
29.118.2.12 platform_addr . . . 149
29.118.2.13 tmp0 . . . 149
29.118.2.14 trap_context . . . 149
29.118.2.15 warmboot_addr . . . 149
29.119 sbi_sse_cb_ops Struct Reference . . . 149
29.119.1 Field Documentation . . . 150
29.119.1.1 complete_cb . . . 150
29.119.1.2 disable_cb . . . 150
29.119.1.3 enable_cb . . . 150
29.119.1.4 register_cb . . . 150
29.119.1.5 set_hartid_cb . . . 150
29.119.1.6 unregister_cb . . . 150
29.120 sbi_system_reset_device Struct Reference . . . 150
29.120.1 Detailed Description . . . 150
29.120.2 Field Documentation . . . 151
29.120.2.1 name . . . 151
29.120.2.2 node . . . 151
29.120.2.3 system_reset . . . 151
29.120.2.4 system_reset_check . . . 151
29.121 sbi_system_suspend_device Struct Reference . . . 151
29.121.1 Detailed Description . . . 151
29.121.2 Field Documentation . . . 151
29.121.2.1 name . . . 151
29.121.2.2 system_suspend . . . 151
29.121.2.3 system_suspend_check . . . 151
29.122 sbi_timer_device Struct Reference . . . 152
29.122.1 Detailed Description . . . 152
29.122.2 Field Documentation . . . 152
29.122.2.1 name . . . 152
29.122.2.2 timer_event_start . . . 152
29.122.2.3 timer_event_stop . . . 152
29.122.2.4 timer_freq . . . 152
29.122.2.5 timer_value . . . 152
29.122.2.6 warm_init . . . 152

29.123 sbi_tlb_info Struct Reference . . . 153
29.123.1 Field Documentation . . . 153
29.123.1.1 asid . . . 153
29.123.1.2 size . . . 153
29.123.1.3 smask . . . 153
29.123.1.4 start . . . 153
29.123.1.5 type . . . 153
29.123.1.6 vmid . . . 153
29.124 sbi_trap_context Struct Reference . . . 153
29.124.1 Detailed Description . . . 153
29.124.2 Field Documentation . . . 154
29.124.2.1 prev_context . . . 154
29.124.2.2 regs . . . 154
29.124.2.3 trap . . . 154
29.125 sbi_trap_info Struct Reference . . . 154
29.125.1 Detailed Description . . . 154
29.125.2 Field Documentation . . . 154
29.125.2.1 cause . . . 154
29.125.2.2 gva . . . 154
29.125.2.3 tinst . . . 154
29.125.2.4 tval . . . 154
29.125.2.5 tval2 . . . 155
29.126 sbi_trap_regs Struct Reference . . . 155
29.126.1 Detailed Description . . . 155
29.126.2 Field Documentation . . . 156
29.126.2.1 a0 . . . 156
29.126.2.2 a1 . . . 156
29.126.2.3 a2 . . . 156
29.126.2.4 a3 . . . 156
29.126.2.5 a4 . . . 156
29.126.2.6 a5 . . . 156
29.126.2.7 a6 . . . 156
29.126.2.8 a7 . . . 156
29.126.2.9 gp . . . 156
29.126.2.10 mepc . . . 156
29.126.2.11 mstatus . . . 156
29.126.2.12 mstatusH . . . 156
29.126.2.13 ra . . . 157
29.126.2.14 s0 . . . 157
29.126.2.15 s1 . . . 157
29.126.2.16 s10 . . . 157
29.126.2.17 s11 . . . 157

29.126.2.18 s2 . . . 157
29.126.2.19 s3 . . . 157
29.126.2.20 s4 . . . 157
29.126.2.21 s5 . . . 157
29.126.2.22 s6 . . . 157
29.126.2.23 s7 . . . 157
29.126.2.24 s8 . . . 157
29.126.2.25 s9 . . . 158
29.126.2.26 sp . . . 158
29.126.2.27 t0 . . . 158
29.126.2.28 t1 . . . 158
29.126.2.29 t2 . . . 158
29.126.2.30 t3 . . . 158
29.126.2.31 t4 . . . 158
29.126.2.32 t5 . . . 158
29.126.2.33 t6 . . . 158
29.126.2.34 tp . . . 158
29.126.2.35 zero . . . 158
29.127 smu_data Struct Reference . . . 158
29.127.1 Field Documentation . . . 159
29.127.1.1 addr . . . 159
29.128 uu Union Reference . . . 159
29.128.1 Field Documentation . . . 159
29.128.1.1 q . . . 159
29.128.1.2 sl . . . 159
29.128.1.3 ul . . . 159
29.128.1.4 uq . . . 159

30 File Documentation **161**
30.1 docs/contributing.md File Reference . . . 162
30.2 docs/domain_support.md File Reference . . . 162
30.3 docs/firmware/fw.md File Reference . . . 162
30.4 docs/firmware/fw_dynamic.md File Reference . . . 162
30.5 docs/firmware/fw_jump.md File Reference . . . 162
30.6 docs/firmware/fw_payload.md File Reference . . . 162
30.7 docs/firmware/payload_linux.md File Reference . . . 162
30.8 docs/firmware/payload_uboot.md File Reference . . . 162
30.9 docs/library_usage.md File Reference . . . 162
30.10 docs/opensbi_config.md File Reference . . . 162
30.11 docs/platform/andes-ae350.md File Reference . . . 162
30.12 docs/platform/fpga-ariane.md File Reference . . . 162
30.13 docs/platform/fpga-openpiton.md File Reference . . . 162

30.14 docs/platform/generic.md File Reference . . . 162
30.15 docs/platform/nuclei_ux600.md File Reference . . . 162
30.16 docs/platform/platform.md File Reference . . . 162
30.17 docs/platform/qemu_virt.md File Reference . . . 162
30.18 docs/platform/renesas-rzfive.md File Reference . . . 162
30.19 docs/platform/shakti_cclass.md File Reference . . . 162
30.20 docs/platform/sifive_fu540.md File Reference . . . 162
30.21 docs/platform/spike.md File Reference . . . 162
30.22 docs/platform/thead-c9xx.md File Reference . . . 162
30.23 docs/platform_guide.md File Reference . . . 162
30.24 docs/platform_requirements.md File Reference . . . 162
30.25 docs/writing_tests.md File Reference . . . 162
30.26 include/sbi/fw_dynamic.h File Reference . . . 162
30.26.1 Macro Definition Documentation . . . 163
30.26.1.1 FW_DYNAMIC_INFO_BOOT_HART_OFFSET . . . 163
30.26.1.2 FW_DYNAMIC_INFO_MAGIC_OFFSET . . . 164
30.26.1.3 FW_DYNAMIC_INFO_MAGIC_VALUE . . . 164
30.26.1.4 FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET . . . 164
30.26.1.5 FW_DYNAMIC_INFO_NEXT_MODE_M . . . 164
30.26.1.6 FW_DYNAMIC_INFO_NEXT_MODE_OFFSET . . . 164
30.26.1.7 FW_DYNAMIC_INFO_NEXT_MODE_S . . . 164
30.26.1.8 FW_DYNAMIC_INFO_NEXT_MODE_U . . . 164
30.26.1.9 FW_DYNAMIC_INFO_OPTIONS_OFFSET . . . 164
30.26.1.10 FW_DYNAMIC_INFO_VERSION_2 . . . 164
30.26.1.11 FW_DYNAMIC_INFO_VERSION_MAX . . . 164
30.26.1.12 FW_DYNAMIC_INFO_VERSION_OFFSET . . . 164
30.26.2 Variable Documentation . . . 164
30.26.2.1 __packed . . . 164
30.27 fw_dynamic.h . . . 165
30.28 include/sbi/riscv_asm.h File Reference . . . 166
30.28.1 Macro Definition Documentation . . . 167
30.28.1.1 __ASM_STR . . . 167
30.28.1.2 csr_clear . . . 167
30.28.1.3 csr_read . . . 167
30.28.1.4 csr_read_clear . . . 167
30.28.1.5 csr_read_relaxed . . . 167
30.28.1.6 csr_read_set . . . 168
30.28.1.7 csr_set . . . 168
30.28.1.8 csr_swap . . . 168
30.28.1.9 csr_write . . . 168
30.28.1.10 current_hartid . . . 168
30.28.1.11 ebreak . . . 168

30.28.1.12 LGREG . . . 169
30.28.1.13 misa_extension . . . 169
30.28.1.14 PAGE_MASK . . . 169
30.28.1.15 PAGE_SHIFT . . . 169
30.28.1.16 PAGE_SIZE . . . 169
30.28.1.17 REG_L . . . 169
30.28.1.18 REG_S . . . 169
30.28.1.19 SZREG . . . 169
30.28.1.20 wfi . . . 169
30.28.2 Function Documentation . . . 169
30.28.2.1 csr_read_num() . . . 169
30.28.2.2 csr_write_num() . . . 170
30.28.2.3 is_pmp_entry_mapped() . . . 171
30.28.2.4 misa_extension_imp() . . . 172
30.28.2.5 misa_string() . . . 172
30.28.2.6 misa_xlen() . . . 172
30.28.2.7 pmp_disable() . . . 173
30.28.2.8 pmp_get() . . . 174
30.28.2.9 pmp_set() . . . 175
30.29 riscv_asm.h . . . 176
30.30 include/sbi/riscv_atomic.h File Reference . . . 178
30.30.1 Macro Definition Documentation . . . 179
30.30.1.1 ATOMIC_INIT . . . 179
30.30.1.2 ATOMIC_INITIALIZER . . . 179
30.30.2 Function Documentation . . . 179
30.30.2.1 atomic_add_return() . . . 179
30.30.2.2 atomic_clear_bit() . . . 179
30.30.2.3 atomic_cmpxchg() . . . 180
30.30.2.4 atomic_raw_clear_bit() . . . 180
30.30.2.5 atomic_raw_set_bit() . . . 181
30.30.2.6 atomic_raw_xchg_uint() . . . 181
30.30.2.7 atomic_raw_xchg_ulong() . . . 181
30.30.2.8 atomic_read() . . . 181
30.30.2.9 atomic_set_bit() . . . 182
30.30.2.10 atomic_sub_return() . . . 182
30.30.2.11 atomic_write() . . . 183
30.30.2.12 atomic_xchg() . . . 183
30.31 riscv_atomic.h . . . 184
30.32 include/sbi/riscv_barrier.h File Reference . . . 185
30.32.1 Macro Definition Documentation . . . 185
30.32.1.1 __smp_load_acquire . . . 185
30.32.1.2 __smp_store_release . . . 185

30.32.1.3 cpu_relax . . . 185
30.32.1.4 mb . . . 185
30.32.1.5 RISCV_ACQUIRE_BARRIER . . . 185
30.32.1.6 RISCV_FENCE . . . 186
30.32.1.7 RISCV_FENCE_I . . . 186
30.32.1.8 RISCV_RELEASE_BARRIER . . . 186
30.32.1.9 rmb . . . 186
30.32.1.10 smp_mb . . . 186
30.32.1.11 smp_rmb . . . 186
30.32.1.12 smp_wmb . . . 186
30.32.1.13 wmb . . . 186
30.33 riscv_barrier.h . . . 186
30.34 include/sbi/riscv_dbtr.h File Reference . . . 187
30.34.1 Macro Definition Documentation . . . 189
30.34.1.1 CLEAR_DBTR_BIT . . . 189
30.34.1.2 RV_DBTR_BIT . . . 189
30.34.1.3 RV_DBTR_BIT_MASK . . . 189
30.34.1.4 RV_DBTR_DECLARE_BIT . . . 189
30.34.1.5 RV_DBTR_DECLARE_BIT_MASK . . . 189
30.34.1.6 RV_DBTR_SET_MC6_EXEC . . . 189
30.34.1.7 RV_DBTR_SET_MC6_LOAD . . . 189
30.34.1.8 RV_DBTR_SET_MC6_SIZE . . . 190
30.34.1.9 RV_DBTR_SET_MC6_STORE . . . 190
30.34.1.10 RV_DBTR_SET_MC6_TYPE . . . 190
30.34.1.11 RV_DBTR_SET_MC_EXEC . . . 190
30.34.1.12 RV_DBTR_SET_MC_LOAD . . . 190
30.34.1.13 RV_DBTR_SET_MC_SIZEHI . . . 190
30.34.1.14 RV_DBTR_SET_MC_SIZELO . . . 190
30.34.1.15 RV_DBTR_SET_MC_STORE . . . 191
30.34.1.16 RV_DBTR_SET_MC_TYPE . . . 191
30.34.1.17 RV_DBTR_SET_TDATA1_TYPE . . . 191
30.34.1.18 RV_MAX_TRIGGERS . . . 191
30.34.1.19 SET_DBTR_BIT . . . 191
30.34.2 Typedef Documentation . . . 191
30.34.2.1 riscv_dbtr_tdata1_mcontrol6_t . . . 191
30.34.2.2 riscv_dbtr_tdata1_mcontrol_t . . . 191
30.34.2.3 riscv_dbtr_tdata1_t . . . 191
30.34.3 Enumeration Type Documentation . . . 191
30.34.3.1 anonymous enum . . . 191
30.34.3.2 anonymous enum . . . 192
30.34.3.3 anonymous enum . . . 192
30.34.3.4 anonymous enum . . . 192

30.34.3.5 anonymous enum . . . 192
30.34.3.6 anonymous enum . . . 193
30.34.3.7 anonymous enum . . . 193
30.35 riscv_dbtr.h . . . 194
30.36 include/sbi/riscv_elf.h File Reference . . . 197
30.36.1 Macro Definition Documentation . . . 197
30.36.1.1 R_RISCV_RELATIVE . . . 197
30.37 riscv_elf.h . . . 197
30.38 include/sbi/riscv_encoding.h File Reference . . . 197
30.38.1 Macro Definition Documentation . . . 213
30.38.1.1 CAUSE_BREAKPOINT . . . 213
30.38.1.2 CAUSE_DOUBLE_TRAP . . . 214
30.38.1.3 CAUSE_FETCH_ACCESS . . . 214
30.38.1.4 CAUSE_FETCH_GUEST_PAGE_FAULT . . . 214
30.38.1.5 CAUSE_FETCH_PAGE_FAULT . . . 214
30.38.1.6 CAUSE_ILLEGAL_INSTRUCTION . . . 214
30.38.1.7 CAUSE_LOAD_ACCESS . . . 214
30.38.1.8 CAUSE_LOAD_GUEST_PAGE_FAULT . . . 214
30.38.1.9 CAUSE_LOAD_PAGE_FAULT . . . 214
30.38.1.10 CAUSE_MACHINE_ECALL . . . 214
30.38.1.11 CAUSE_MISALIGNED_FETCH . . . 214
30.38.1.12 CAUSE_MISALIGNED_LOAD . . . 214
30.38.1.13 CAUSE_MISALIGNED_STORE . . . 214
30.38.1.14 CAUSE_STORE_ACCESS . . . 214
30.38.1.15 CAUSE_STORE_GUEST_PAGE_FAULT . . . 214
30.38.1.16 CAUSE_STORE_PAGE_FAULT . . . 214
30.38.1.17 CAUSE_SUPERVISOR_ECALL . . . 215
30.38.1.18 CAUSE_SW_CHECK_EXCP . . . 215
30.38.1.19 CAUSE_USER_ECALL . . . 215
30.38.1.20 CAUSE_VIRTUAL_INST_FAULT . . . 215
30.38.1.21 CAUSE_VIRTUAL_SUPERVISOR_ECALL . . . 215
30.38.1.22 CSR_CYCLE . . . 215
30.38.1.23 CSR_CYCLEH . . . 215
30.38.1.24 CSR_DCSR . . . 215
30.38.1.25 CSR_DPC . . . 215
30.38.1.26 CSR_DSCRATCH0 . . . 215
30.38.1.27 CSR_DSCRATCH1 . . . 215
30.38.1.28 CSR_FCSR . . . 215
30.38.1.29 CSR_FFLAGS . . . 215
30.38.1.30 CSR_FRM . . . 215
30.38.1.31 CSR_HCOUNTEREN . . . 215
30.38.1.32 CSR_HEDELEG . . . 216

30.38.1.33 CSR_HENVCFG . . . 216
30.38.1.34 CSR_HENVCFGH . . . 216
30.38.1.35 CSR_HGATP . . . 216
30.38.1.36 CSR_HGEIE . . . 216
30.38.1.37 CSR_HGEIP . . . 216
30.38.1.38 CSR_HIDELEG . . . 216
30.38.1.39 CSR_HIDELEGH . . . 216
30.38.1.40 CSR_HIE . . . 216
30.38.1.41 CSR_HIP . . . 216
30.38.1.42 CSR_HPMCOUNTER10 . . . 216
30.38.1.43 CSR_HPMCOUNTER10H . . . 216
30.38.1.44 CSR_HPMCOUNTER11 . . . 216
30.38.1.45 CSR_HPMCOUNTER11H . . . 216
30.38.1.46 CSR_HPMCOUNTER12 . . . 216
30.38.1.47 CSR_HPMCOUNTER12H . . . 217
30.38.1.48 CSR_HPMCOUNTER13 . . . 217
30.38.1.49 CSR_HPMCOUNTER13H . . . 217
30.38.1.50 CSR_HPMCOUNTER14 . . . 217
30.38.1.51 CSR_HPMCOUNTER14H . . . 217
30.38.1.52 CSR_HPMCOUNTER15 . . . 217
30.38.1.53 CSR_HPMCOUNTER15H . . . 217
30.38.1.54 CSR_HPMCOUNTER16 . . . 217
30.38.1.55 CSR_HPMCOUNTER16H . . . 217
30.38.1.56 CSR_HPMCOUNTER17 . . . 217
30.38.1.57 CSR_HPMCOUNTER17H . . . 217
30.38.1.58 CSR_HPMCOUNTER18 . . . 217
30.38.1.59 CSR_HPMCOUNTER18H . . . 217
30.38.1.60 CSR_HPMCOUNTER19 . . . 217
30.38.1.61 CSR_HPMCOUNTER19H . . . 217
30.38.1.62 CSR_HPMCOUNTER20 . . . 218
30.38.1.63 CSR_HPMCOUNTER20H . . . 218
30.38.1.64 CSR_HPMCOUNTER21 . . . 218
30.38.1.65 CSR_HPMCOUNTER21H . . . 218
30.38.1.66 CSR_HPMCOUNTER22 . . . 218
30.38.1.67 CSR_HPMCOUNTER22H . . . 218
30.38.1.68 CSR_HPMCOUNTER23 . . . 218
30.38.1.69 CSR_HPMCOUNTER23H . . . 218
30.38.1.70 CSR_HPMCOUNTER24 . . . 218
30.38.1.71 CSR_HPMCOUNTER24H . . . 218
30.38.1.72 CSR_HPMCOUNTER25 . . . 218
30.38.1.73 CSR_HPMCOUNTER25H . . . 218
30.38.1.74 CSR_HPMCOUNTER26 . . . 218

30.38.1.75 CSR_HPMCOUNTER26H . . . 218
30.38.1.76 CSR_HPMCOUNTER27 . . . 218
30.38.1.77 CSR_HPMCOUNTER27H . . . 219
30.38.1.78 CSR_HPMCOUNTER28 . . . 219
30.38.1.79 CSR_HPMCOUNTER28H . . . 219
30.38.1.80 CSR_HPMCOUNTER29 . . . 219
30.38.1.81 CSR_HPMCOUNTER29H . . . 219
30.38.1.82 CSR_HPMCOUNTER3 . . . 219
30.38.1.83 CSR_HPMCOUNTER30 . . . 219
30.38.1.84 CSR_HPMCOUNTER30H . . . 219
30.38.1.85 CSR_HPMCOUNTER31 . . . 219
30.38.1.86 CSR_HPMCOUNTER31H . . . 219
30.38.1.87 CSR_HPMCOUNTER3H . . . 219
30.38.1.88 CSR_HPMCOUNTER4 . . . 219
30.38.1.89 CSR_HPMCOUNTER4H . . . 219
30.38.1.90 CSR_HPMCOUNTER5 . . . 219
30.38.1.91 CSR_HPMCOUNTER5H . . . 219
30.38.1.92 CSR_HPMCOUNTER6 . . . 220
30.38.1.93 CSR_HPMCOUNTER6H . . . 220
30.38.1.94 CSR_HPMCOUNTER7 . . . 220
30.38.1.95 CSR_HPMCOUNTER7H . . . 220
30.38.1.96 CSR_HPMCOUNTER8 . . . 220
30.38.1.97 CSR_HPMCOUNTER8H . . . 220
30.38.1.98 CSR_HPMCOUNTER9 . . . 220
30.38.1.99 CSR_HPMCOUNTER9H . . . 220
30.38.1.100 CSR_HSTATEEN0 . . . 220
30.38.1.101 CSR_HSTATEEN0H . . . 220
30.38.1.102 CSR_HSTATEEN1 . . . 220
30.38.1.103 CSR_HSTATEEN1H . . . 220
30.38.1.104 CSR_HSTATEEN2 . . . 220
30.38.1.105 CSR_HSTATEEN2H . . . 220
30.38.1.106 CSR_HSTATEEN3 . . . 220
30.38.1.107 CSR_HSTATEEN3H . . . 221
30.38.1.108 CSR_HSTATUS . . . 221
30.38.1.109 CSR_HTIMEDELTA . . . 221
30.38.1.110 CSR_HTIMEDELTAH . . . 221
30.38.1.111 CSR_HTINST . . . 221
30.38.1.112 CSR_HTVAL . . . 221
30.38.1.113 CSR_HVICTL . . . 221
30.38.1.114 CSR_HVIEN . . . 221
30.38.1.115 CSR_HVIENH . . . 221
30.38.1.116 CSR_HVIP . . . 221

30.38.1.117 CSR_HVIPH . . . 221
30.38.1.118 CSR_HVIPRIO1 . . . 221
30.38.1.119 CSR_HVIPRIO1H . . . 221
30.38.1.120 CSR_HVIPRIO2 . . . 221
30.38.1.121 CSR_HVIPRIO2H . . . 221
30.38.1.122 CSR_INSTRET . . . 222
30.38.1.123 CSR_INSTRETH . . . 222
30.38.1.124 CSR_MARCHID . . . 222
30.38.1.125 CSR_MCAUSE . . . 222
30.38.1.126 CSR_MCONFIGPTR . . . 222
30.38.1.127 CSR_MCOUNTEREN . . . 222
30.38.1.128 CSR_MCOUNTINHIBIT . . . 222
30.38.1.129 CSR_MCYCLE . . . 222
30.38.1.130 CSR_MCYCLECFG . . . 222
30.38.1.131 CSR_MCYCLECFGH . . . 222
30.38.1.132 CSR_MCYCLEH . . . 222
30.38.1.133 CSR_MEDELEG . . . 222
30.38.1.134 CSR_MENVCFG . . . 222
30.38.1.135 CSR_MENVCFGH . . . 222
30.38.1.136 CSR_MEPC . . . 222
30.38.1.137 CSR_MHARTID . . . 223
30.38.1.138 CSR_MHPMCOUNTER10 . . . 223
30.38.1.139 CSR_MHPMCOUNTER10H . . . 223
30.38.1.140 CSR_MHPMCOUNTER11 . . . 223
30.38.1.141 CSR_MHPMCOUNTER11H . . . 223
30.38.1.142 CSR_MHPMCOUNTER12 . . . 223
30.38.1.143 CSR_MHPMCOUNTER12H . . . 223
30.38.1.144 CSR_MHPMCOUNTER13 . . . 223
30.38.1.145 CSR_MHPMCOUNTER13H . . . 223
30.38.1.146 CSR_MHPMCOUNTER14 . . . 223
30.38.1.147 CSR_MHPMCOUNTER14H . . . 223
30.38.1.148 CSR_MHPMCOUNTER15 . . . 223
30.38.1.149 CSR_MHPMCOUNTER15H . . . 223
30.38.1.150 CSR_MHPMCOUNTER16 . . . 223
30.38.1.151 CSR_MHPMCOUNTER16H . . . 223
30.38.1.152 CSR_MHPMCOUNTER17 . . . 224
30.38.1.153 CSR_MHPMCOUNTER17H . . . 224
30.38.1.154 CSR_MHPMCOUNTER18 . . . 224
30.38.1.155 CSR_MHPMCOUNTER18H . . . 224
30.38.1.156 CSR_MHPMCOUNTER19 . . . 224
30.38.1.157 CSR_MHPMCOUNTER19H . . . 224
30.38.1.158 CSR_MHPMCOUNTER20 . . . 224

30.38.1.159 CSR_MHPMCOUNTER20H 224
30.38.1.160 CSR_MHPMCOUNTER21 224
30.38.1.161 CSR_MHPMCOUNTER21H 224
30.38.1.162 CSR_MHPMCOUNTER22 224
30.38.1.163 CSR_MHPMCOUNTER22H 224
30.38.1.164 CSR_MHPMCOUNTER23 224
30.38.1.165 CSR_MHPMCOUNTER23H 224
30.38.1.166 CSR_MHPMCOUNTER24 224
30.38.1.167 CSR_MHPMCOUNTER24H 225
30.38.1.168 CSR_MHPMCOUNTER25 225
30.38.1.169 CSR_MHPMCOUNTER25H 225
30.38.1.170 CSR_MHPMCOUNTER26 225
30.38.1.171 CSR_MHPMCOUNTER26H 225
30.38.1.172 CSR_MHPMCOUNTER27 225
30.38.1.173 CSR_MHPMCOUNTER27H 225
30.38.1.174 CSR_MHPMCOUNTER28 225
30.38.1.175 CSR_MHPMCOUNTER28H 225
30.38.1.176 CSR_MHPMCOUNTER29 225
30.38.1.177 CSR_MHPMCOUNTER29H 225
30.38.1.178 CSR_MHPMCOUNTER3 225
30.38.1.179 CSR_MHPMCOUNTER30 225
30.38.1.180 CSR_MHPMCOUNTER30H 225
30.38.1.181 CSR_MHPMCOUNTER31 225
30.38.1.182 CSR_MHPMCOUNTER31H 226
30.38.1.183 CSR_MHPMCOUNTER3H 226
30.38.1.184 CSR_MHPMCOUNTER4 226
30.38.1.185 CSR_MHPMCOUNTER4H 226
30.38.1.186 CSR_MHPMCOUNTER5 226
30.38.1.187 CSR_MHPMCOUNTER5H 226
30.38.1.188 CSR_MHPMCOUNTER6 226
30.38.1.189 CSR_MHPMCOUNTER6H 226
30.38.1.190 CSR_MHPMCOUNTER7 226
30.38.1.191 CSR_MHPMCOUNTER7H 226
30.38.1.192 CSR_MHPMCOUNTER8 226
30.38.1.193 CSR_MHPMCOUNTER8H 226
30.38.1.194 CSR_MHPMCOUNTER9 226
30.38.1.195 CSR_MHPMCOUNTER9H 226
30.38.1.196 CSR_MHPMEVENT10 226
30.38.1.197 CSR_MHPMEVENT10H 227
30.38.1.198 CSR_MHPMEVENT11 227
30.38.1.199 CSR_MHPMEVENT11H 227
30.38.1.200 CSR_MHPMEVENT12 227

30.38.1.201 CSR_MHPMEVENT12H . . . 227
30.38.1.202 CSR_MHPMEVENT13 . . . 227
30.38.1.203 CSR_MHPMEVENT13H . . . 227
30.38.1.204 CSR_MHPMEVENT14 . . . 227
30.38.1.205 CSR_MHPMEVENT14H . . . 227
30.38.1.206 CSR_MHPMEVENT15 . . . 227
30.38.1.207 CSR_MHPMEVENT15H . . . 227
30.38.1.208 CSR_MHPMEVENT16 . . . 227
30.38.1.209 CSR_MHPMEVENT16H . . . 227
30.38.1.210 CSR_MHPMEVENT17 . . . 227
30.38.1.211 CSR_MHPMEVENT17H . . . 227
30.38.1.212 CSR_MHPMEVENT18 . . . 228
30.38.1.213 CSR_MHPMEVENT18H . . . 228
30.38.1.214 CSR_MHPMEVENT19 . . . 228
30.38.1.215 CSR_MHPMEVENT19H . . . 228
30.38.1.216 CSR_MHPMEVENT20 . . . 228
30.38.1.217 CSR_MHPMEVENT20H . . . 228
30.38.1.218 CSR_MHPMEVENT21 . . . 228
30.38.1.219 CSR_MHPMEVENT21H . . . 228
30.38.1.220 CSR_MHPMEVENT22 . . . 228
30.38.1.221 CSR_MHPMEVENT22H . . . 228
30.38.1.222 CSR_MHPMEVENT23 . . . 228
30.38.1.223 CSR_MHPMEVENT23H . . . 228
30.38.1.224 CSR_MHPMEVENT24 . . . 228
30.38.1.225 CSR_MHPMEVENT24H . . . 228
30.38.1.226 CSR_MHPMEVENT25 . . . 228
30.38.1.227 CSR_MHPMEVENT25H . . . 229
30.38.1.228 CSR_MHPMEVENT26 . . . 229
30.38.1.229 CSR_MHPMEVENT26H . . . 229
30.38.1.230 CSR_MHPMEVENT27 . . . 229
30.38.1.231 CSR_MHPMEVENT27H . . . 229
30.38.1.232 CSR_MHPMEVENT28 . . . 229
30.38.1.233 CSR_MHPMEVENT28H . . . 229
30.38.1.234 CSR_MHPMEVENT29 . . . 229
30.38.1.235 CSR_MHPMEVENT29H . . . 229
30.38.1.236 CSR_MHPMEVENT3 . . . 229
30.38.1.237 CSR_MHPMEVENT30 . . . 229
30.38.1.238 CSR_MHPMEVENT30H . . . 229
30.38.1.239 CSR_MHPMEVENT31 . . . 229
30.38.1.240 CSR_MHPMEVENT31H . . . 229
30.38.1.241 CSR_MHPMEVENT3H . . . 229
30.38.1.242 CSR_MHPMEVENT4 . . . 230

30.38.1.243 CSR_MHPMEVENT4H . . . 230
30.38.1.244 CSR_MHPMEVENT5 . . . 230
30.38.1.245 CSR_MHPMEVENT5H . . . 230
30.38.1.246 CSR_MHPMEVENT6 . . . 230
30.38.1.247 CSR_MHPMEVENT6H . . . 230
30.38.1.248 CSR_MHPMEVENT7 . . . 230
30.38.1.249 CSR_MHPMEVENT7H . . . 230
30.38.1.250 CSR_MHPMEVENT8 . . . 230
30.38.1.251 CSR_MHPMEVENT8H . . . 230
30.38.1.252 CSR_MHPMEVENT9 . . . 230
30.38.1.253 CSR_MHPMEVENT9H . . . 230
30.38.1.254 CSR_MIDELEG . . . 230
30.38.1.255 CSR_MIDELEGH . . . 230
30.38.1.256 CSR_MIE . . . 230
30.38.1.257 CSR_MIEH . . . 231
30.38.1.258 CSR_MIMPID . . . 231
30.38.1.259 CSR_MINSTRET . . . 231
30.38.1.260 CSR_MINSTRETCFG . . . 231
30.38.1.261 CSR_MINSTRETCFGH . . . 231
30.38.1.262 CSR_MINSTRETH . . . 231
30.38.1.263 CSR_MIP . . . 231
30.38.1.264 CSR_MIPH . . . 231
30.38.1.265 CSR_MIREG . . . 231
30.38.1.266 CSR_MIREG2 . . . 231
30.38.1.267 CSR_MIREG3 . . . 231
30.38.1.268 CSR_MIREG4 . . . 231
30.38.1.269 CSR_MIREG5 . . . 231
30.38.1.270 CSR_MIREG6 . . . 231
30.38.1.271 CSR_MISA . . . 231
30.38.1.272 CSR_MISELECT . . . 232
30.38.1.273 CSR_MSCRATCH . . . 232
30.38.1.274 CSR_MSECCFG . . . 232
30.38.1.275 CSR_MSECCFGH . . . 232
30.38.1.276 CSR_MSTATEEN0 . . . 232
30.38.1.277 CSR_MSTATEEN0H . . . 232
30.38.1.278 CSR_MSTATEEN1 . . . 232
30.38.1.279 CSR_MSTATEEN1H . . . 232
30.38.1.280 CSR_MSTATEEN2 . . . 232
30.38.1.281 CSR_MSTATEEN2H . . . 232
30.38.1.282 CSR_MSTATEEN3 . . . 232
30.38.1.283 CSR_MSTATEEN3H . . . 232
30.38.1.284 CSR_MSTATUS . . . 232

30.38.1.285 CSR_MSTATUSH . . . 232
30.38.1.286 CSR_MTINST . . . 232
30.38.1.287 CSR_MTOPEI . . . 233
30.38.1.288 CSR_MTOPI . . . 233
30.38.1.289 CSR_MTVAL . . . 233
30.38.1.290 CSR_MTVAL2 . . . 233
30.38.1.291 CSR_MTVEC . . . 233
30.38.1.292 CSR_MVENDORID . . . 233
30.38.1.293 CSR_MVIEN . . . 233
30.38.1.294 CSR_MVIENH . . . 233
30.38.1.295 CSR_MVIP . . . 233
30.38.1.296 CSR_MVIPH . . . 233
30.38.1.297 CSR_PMPADDR0 . . . 233
30.38.1.298 CSR_PMPADDR1 . . . 233
30.38.1.299 CSR_PMPADDR10 . . . 233
30.38.1.300 CSR_PMPADDR11 . . . 233
30.38.1.301 CSR_PMPADDR12 . . . 233
30.38.1.302 CSR_PMPADDR13 . . . 234
30.38.1.303 CSR_PMPADDR14 . . . 234
30.38.1.304 CSR_PMPADDR15 . . . 234
30.38.1.305 CSR_PMPADDR16 . . . 234
30.38.1.306 CSR_PMPADDR17 . . . 234
30.38.1.307 CSR_PMPADDR18 . . . 234
30.38.1.308 CSR_PMPADDR19 . . . 234
30.38.1.309 CSR_PMPADDR2 . . . 234
30.38.1.310 CSR_PMPADDR20 . . . 234
30.38.1.311 CSR_PMPADDR21 . . . 234
30.38.1.312 CSR_PMPADDR22 . . . 234
30.38.1.313 CSR_PMPADDR23 . . . 234
30.38.1.314 CSR_PMPADDR24 . . . 234
30.38.1.315 CSR_PMPADDR25 . . . 234
30.38.1.316 CSR_PMPADDR26 . . . 234
30.38.1.317 CSR_PMPADDR27 . . . 235
30.38.1.318 CSR_PMPADDR28 . . . 235
30.38.1.319 CSR_PMPADDR29 . . . 235
30.38.1.320 CSR_PMPADDR3 . . . 235
30.38.1.321 CSR_PMPADDR30 . . . 235
30.38.1.322 CSR_PMPADDR31 . . . 235
30.38.1.323 CSR_PMPADDR32 . . . 235
30.38.1.324 CSR_PMPADDR33 . . . 235
30.38.1.325 CSR_PMPADDR34 . . . 235
30.38.1.326 CSR_PMPADDR35 . . . 235

30.38.1.327 CSR_PMPADDR36 235
30.38.1.328 CSR_PMPADDR37 235
30.38.1.329 CSR_PMPADDR38 235
30.38.1.330 CSR_PMPADDR39 235
30.38.1.331 CSR_PMPADDR4 235
30.38.1.332 CSR_PMPADDR40 236
30.38.1.333 CSR_PMPADDR41 236
30.38.1.334 CSR_PMPADDR42 236
30.38.1.335 CSR_PMPADDR43 236
30.38.1.336 CSR_PMPADDR44 236
30.38.1.337 CSR_PMPADDR45 236
30.38.1.338 CSR_PMPADDR46 236
30.38.1.339 CSR_PMPADDR47 236
30.38.1.340 CSR_PMPADDR48 236
30.38.1.341 CSR_PMPADDR49 236
30.38.1.342 CSR_PMPADDR5 236
30.38.1.343 CSR_PMPADDR50 236
30.38.1.344 CSR_PMPADDR51 236
30.38.1.345 CSR_PMPADDR52 236
30.38.1.346 CSR_PMPADDR53 236
30.38.1.347 CSR_PMPADDR54 237
30.38.1.348 CSR_PMPADDR55 237
30.38.1.349 CSR_PMPADDR56 237
30.38.1.350 CSR_PMPADDR57 237
30.38.1.351 CSR_PMPADDR58 237
30.38.1.352 CSR_PMPADDR59 237
30.38.1.353 CSR_PMPADDR6 237
30.38.1.354 CSR_PMPADDR60 237
30.38.1.355 CSR_PMPADDR61 237
30.38.1.356 CSR_PMPADDR62 237
30.38.1.357 CSR_PMPADDR63 237
30.38.1.358 CSR_PMPADDR7 237
30.38.1.359 CSR_PMPADDR8 237
30.38.1.360 CSR_PMPADDR9 237
30.38.1.361 CSR_PMPCFG0 237
30.38.1.362 CSR_PMPCFG1 238
30.38.1.363 CSR_PMPCFG10 238
30.38.1.364 CSR_PMPCFG11 238
30.38.1.365 CSR_PMPCFG12 238
30.38.1.366 CSR_PMPCFG13 238
30.38.1.367 CSR_PMPCFG14 238
30.38.1.368 CSR_PMPCFG15 238

30.38.1.369 CSR_PMPCFG2 238
30.38.1.370 CSR_PMPCFG3 238
30.38.1.371 CSR_PMPCFG4 238
30.38.1.372 CSR_PMPCFG5 238
30.38.1.373 CSR_PMPCFG6 238
30.38.1.374 CSR_PMPCFG7 238
30.38.1.375 CSR_PMPCFG8 238
30.38.1.376 CSR_PMPCFG9 238
30.38.1.377 CSR_SATP 239
30.38.1.378 CSR_SCAUSE 239
30.38.1.379 CSR_SCOUNTEREN 239
30.38.1.380 CSR_SCOUNTINHIBIT 239
30.38.1.381 CSR_SCOUNTOVF 239
30.38.1.382 CSR_SENVCFG 239
30.38.1.383 CSR_SEPC 239
30.38.1.384 CSR_SIE 239
30.38.1.385 CSR_SIEH 239
30.38.1.386 CSR_SIP 239
30.38.1.387 CSR_SIPH 239
30.38.1.388 CSR_SIREG 239
30.38.1.389 CSR_SIREG2 239
30.38.1.390 CSR_SIREG3 239
30.38.1.391 CSR_SIREG4 239
30.38.1.392 CSR_SIREG5 240
30.38.1.393 CSR_SIREG6 240
30.38.1.394 CSR_SISELECT 240
30.38.1.395 CSR_SSCRATCH 240
30.38.1.396 CSR_SSP 240
30.38.1.397 CSR_SSTATEEN0 240
30.38.1.398 CSR_SSTATEEN1 240
30.38.1.399 CSR_SSTATEEN2 240
30.38.1.400 CSR_SSTATEEN3 240
30.38.1.401 CSR_SSTATUS 240
30.38.1.402 CSR_STIMECMP 240
30.38.1.403 CSR_STIMECMPH 240
30.38.1.404 CSR_STOPEI 240
30.38.1.405 CSR_STOPI 240
30.38.1.406 CSR_STVAL 240
30.38.1.407 CSR_STVEC 241
30.38.1.408 CSR_TDATA1 241
30.38.1.409 CSR_TDATA2 241
30.38.1.410 CSR_TDATA3 241

30.38.1.411 CSR_TIME . . . 241
30.38.1.412 CSR_TIMEH . . . 241
30.38.1.413 CSR_TINFO . . . 241
30.38.1.414 CSR_TSELECT . . . 241
30.38.1.415 CSR_UCAUSE . . . 241
30.38.1.416 CSR_UEPC . . . 241
30.38.1.417 CSR_UIE . . . 241
30.38.1.418 CSR_UIP . . . 241
30.38.1.419 CSR_USCRATCH . . . 241
30.38.1.420 CSR_USTATUS . . . 241
30.38.1.421 CSR_UTVAL . . . 241
30.38.1.422 CSR_UTVEC . . . 242
30.38.1.423 CSR_VL . . . 242
30.38.1.424 CSR_VLENB . . . 242
30.38.1.425 CSR_VSATP . . . 242
30.38.1.426 CSR_VSCAUSE . . . 242
30.38.1.427 CSR_VSEPC . . . 242
30.38.1.428 CSR_VSIE . . . 242
30.38.1.429 CSR_VSIEH . . . 242
30.38.1.430 CSR_VSIP . . . 242
30.38.1.431 CSR_VSIPH . . . 242
30.38.1.432 CSR_VSIREG . . . 242
30.38.1.433 CSR_VSIREG2 . . . 242
30.38.1.434 CSR_VSIREG3 . . . 242
30.38.1.435 CSR_VSIREG4 . . . 242
30.38.1.436 CSR_VSIREG5 . . . 242
30.38.1.437 CSR_VSIREG6 . . . 243
30.38.1.438 CSR_VSISELECT . . . 243
30.38.1.439 CSR_VSSCRATCH . . . 243
30.38.1.440 CSR_VSSTATUS . . . 243
30.38.1.441 CSR_VSTART . . . 243
30.38.1.442 CSR_VSTOPEI . . . 243
30.38.1.443 CSR_VSTOPI . . . 243
30.38.1.444 CSR_VSTVAL . . . 243
30.38.1.445 CSR_VSTVEC . . . 243
30.38.1.446 CSR_VTYPE . . . 243
30.38.1.447 CSRRC . . . 243
30.38.1.448 CSRRCI . . . 243
30.38.1.449 CSRRS . . . 243
30.38.1.450 CSRRSI . . . 243
30.38.1.451 CSRRW . . . 243
30.38.1.452 CSRRWI . . . 244

30.38.1.453 ENVCFG_ADUE . . . 244
30.38.1.454 ENVCFG_ADUE_SHIFT . . . 244
30.38.1.455 ENVCFG_CBCFE . . . 244
30.38.1.456 ENVCFG_CBIE . . . 244
30.38.1.457 ENVCFG_CBIE_FLUSH . . . 244
30.38.1.458 ENVCFG_CBIE_ILL . . . 244
30.38.1.459 ENVCFG_CBIE_INV . . . 244
30.38.1.460 ENVCFG_CBIE_SHIFT . . . 244
30.38.1.461 ENVCFG_CBZE . . . 244
30.38.1.462 ENVCFG_CDE . . . 244
30.38.1.463 ENVCFG_DTE . . . 244
30.38.1.464 ENVCFG_DTE_SHIFT . . . 244
30.38.1.465 ENVCFG_FIOM . . . 244
30.38.1.466 ENVCFG_LPE . . . 244
30.38.1.467 ENVCFG_LPE_SHIFT . . . 245
30.38.1.468 ENVCFG_PBMTE . . . 245
30.38.1.469 ENVCFG_PMM . . . 245
30.38.1.470 ENVCFG_PMM_PMLEN_0 . . . 245
30.38.1.471 ENVCFG_PMM_PMLEN_16 . . . 245
30.38.1.472 ENVCFG_PMM_PMLEN_7 . . . 245
30.38.1.473 ENVCFG_SSE . . . 245
30.38.1.474 ENVCFG_SSE_SHIFT . . . 245
30.38.1.475 ENVCFG_STCE . . . 245
30.38.1.476 GET_CSR_NUM . . . 245
30.38.1.477 GET_EMUL . . . 245
30.38.1.478 GET_LEN . . . 245
30.38.1.479 GET_MEW . . . 245
30.38.1.480 GET_NF . . . 245
30.38.1.481 GET_RM . . . 246
30.38.1.482 GET_RS1 . . . 246
30.38.1.483 GET_RS1_NUM . . . 246
30.38.1.484 GET_RS1S . . . 246
30.38.1.485 GET_RS2 . . . 246
30.38.1.486 GET_RS2C . . . 246
30.38.1.487 GET_RS2S . . . 246
30.38.1.488 GET_SP . . . 246
30.38.1.489 GET_VD . . . 246
30.38.1.490 GET_VEMUL . . . 246
30.38.1.491 GET_VIEW . . . 246
30.38.1.492 GET_VLMUL . . . 247
30.38.1.493 GET_VS2 . . . 247
30.38.1.494 GET_VSEW . . . 247

30.38.1.495 HGATP32_MODE_SHIFT . . . 247
30.38.1.496 HGATP32_PPN . . . 247
30.38.1.497 HGATP32_VMID_MASK . . . 247
30.38.1.498 HGATP32_VMID_SHIFT . . . 247
30.38.1.499 HGATP64_MODE_SHIFT . . . 247
30.38.1.500 HGATP64_PPN . . . 247
30.38.1.501 HGATP64_VMID_MASK . . . 247
30.38.1.502 HGATP64_VMID_SHIFT . . . 247
30.38.1.503 HGATP_MODE_OFF . . . 247
30.38.1.504 HGATP_MODE_SHIFT . . . 247
30.38.1.505 HGATP_MODE_SV32X4 . . . 247
30.38.1.506 HGATP_MODE_SV39X4 . . . 248
30.38.1.507 HGATP_MODE_SV48X4 . . . 248
30.38.1.508 HGATP_PPN . . . 248
30.38.1.509 HGATP_VMID_MASK . . . 248
30.38.1.510 HGATP_VMID_SHIFT . . . 248
30.38.1.511 HSTATUS_GVA . . . 248
30.38.1.512 HSTATUS_HU . . . 248
30.38.1.513 HSTATUS_SPV . . . 248
30.38.1.514 HSTATUS_SPVP . . . 248
30.38.1.515 HSTATUS_VGEIN . . . 248
30.38.1.516 HSTATUS_VGEIN_SHIFT . . . 248
30.38.1.517 HSTATUS_VSBE . . . 248
30.38.1.518 HSTATUS_VTSR . . . 248
30.38.1.519 HSTATUS_VTVM . . . 248
30.38.1.520 HSTATUS_VTW . . . 248
30.38.1.521 IMM_I . . . 249
30.38.1.522 IMM_S . . . 249
30.38.1.523 INSN_16BIT_MASK . . . 249
30.38.1.524 INSN_32BIT_MASK . . . 249
30.38.1.525 INSN_IS_16BIT . . . 249
30.38.1.526 INSN_IS_32BIT . . . 249
30.38.1.527 INSN_LEN . . . 249
30.38.1.528 INSN_MASK_C_FLD . . . 249
30.38.1.529 INSN_MASK_C_FLDSP . . . 249
30.38.1.530 INSN_MASK_C_FLW . . . 249
30.38.1.531 INSN_MASK_C_FLWSP . . . 249
30.38.1.532 INSN_MASK_C_FSD . . . 249
30.38.1.533 INSN_MASK_C_FSDSP . . . 249
30.38.1.534 INSN_MASK_C_FSW . . . 250
30.38.1.535 INSN_MASK_C_FSWSP . . . 250
30.38.1.536 INSN_MASK_C_LD . . . 250

30.38.1.537 INSN_MASK_C_LDSP . . . 250
30.38.1.538 INSN_MASK_C_LH . . . 250
30.38.1.539 INSN_MASK_C_LHU . . . 250
30.38.1.540 INSN_MASK_C_LW . . . 250
30.38.1.541 INSN_MASK_C_LWSP . . . 250
30.38.1.542 INSN_MASK_C_SD . . . 250
30.38.1.543 INSN_MASK_C_SDSP . . . 250
30.38.1.544 INSN_MASK_C_SH . . . 250
30.38.1.545 INSN_MASK_C_SW . . . 250
30.38.1.546 INSN_MASK_C_SWSP . . . 250
30.38.1.547 INSN_MASK_FENCE_TSO . . . 250
30.38.1.548 INSN_MASK_FLD . . . 250
30.38.1.549 INSN_MASK_FLQ . . . 251
30.38.1.550 INSN_MASK_FLW . . . 251
30.38.1.551 INSN_MASK_FSD . . . 251
30.38.1.552 INSN_MASK_FSQ . . . 251
30.38.1.553 INSN_MASK_FSW . . . 251
30.38.1.554 INSN_MASK_LB . . . 251
30.38.1.555 INSN_MASK_LBU . . . 251
30.38.1.556 INSN_MASK_LD . . . 251
30.38.1.557 INSN_MASK_LH . . . 251
30.38.1.558 INSN_MASK_LHU . . . 251
30.38.1.559 INSN_MASK_LW . . . 251
30.38.1.560 INSN_MASK_LWU . . . 251
30.38.1.561 INSN_MASK_SB . . . 251
30.38.1.562 INSN_MASK_SD . . . 251
30.38.1.563 INSN_MASK_SH . . . 251
30.38.1.564 INSN_MASK_SW . . . 252
30.38.1.565 INSN_MASK_VECTOR_FAULT_ONLY_FIRST . . . 252
30.38.1.566 INSN_MASK_VECTOR_INDEXED . . . 252
30.38.1.567 INSN_MASK_VECTOR_LOAD_STORE . . . 252
30.38.1.568 INSN_MASK_VECTOR_STRIDE . . . 252
30.38.1.569 INSN_MASK_VECTOR_UNIT_STRIDE . . . 252
30.38.1.570 INSN_MASK_VECTOR_WHOLE_REG . . . 252
30.38.1.571 INSN_MASK_WFI . . . 252
30.38.1.572 INSN_MATCH_C_FLD . . . 252
30.38.1.573 INSN_MATCH_C_FLDSP . . . 252
30.38.1.574 INSN_MATCH_C_FLW . . . 252
30.38.1.575 INSN_MATCH_C_FLWSP . . . 252
30.38.1.576 INSN_MATCH_C_FSD . . . 252
30.38.1.577 INSN_MATCH_C_FSDSP . . . 252
30.38.1.578 INSN_MATCH_C_FSW . . . 252

30.38.1.579 INSN_MATCH_C_FSWSP 253
30.38.1.580 INSN_MATCH_C_LD 253
30.38.1.581 INSN_MATCH_C_LDSP 253
30.38.1.582 INSN_MATCH_C_LH 253
30.38.1.583 INSN_MATCH_C_LHU 253
30.38.1.584 INSN_MATCH_C_LW 253
30.38.1.585 INSN_MATCH_C_LWSP 253
30.38.1.586 INSN_MATCH_C_SD 253
30.38.1.587 INSN_MATCH_C_SDSP 253
30.38.1.588 INSN_MATCH_C_SH 253
30.38.1.589 INSN_MATCH_C_SW 253
30.38.1.590 INSN_MATCH_C_SWSP 253
30.38.1.591 INSN_MATCH_FENCE_TSO 253
30.38.1.592 INSN_MATCH_FLD 253
30.38.1.593 INSN_MATCH_FLQ 253
30.38.1.594 INSN_MATCH_FLW 254
30.38.1.595 INSN_MATCH_FSD 254
30.38.1.596 INSN_MATCH_FSQ 254
30.38.1.597 INSN_MATCH_FSW 254
30.38.1.598 INSN_MATCH_LB 254
30.38.1.599 INSN_MATCH_LBU 254
30.38.1.600 INSN_MATCH_LD 254
30.38.1.601 INSN_MATCH_LH 254
30.38.1.602 INSN_MATCH_LHU 254
30.38.1.603 INSN_MATCH_LW 254
30.38.1.604 INSN_MATCH_LWU 254
30.38.1.605 INSN_MATCH_SB 254
30.38.1.606 INSN_MATCH_SD 254
30.38.1.607 INSN_MATCH_SH 254
30.38.1.608 INSN_MATCH_SW 254
30.38.1.609 INSN_MATCH_VECTOR_LOAD 255
30.38.1.610 INSN_MATCH_VECTOR_STORE 255
30.38.1.611 INSN_MATCH_VL1RE16V 255
30.38.1.612 INSN_MATCH_VL1RE32V 255
30.38.1.613 INSN_MATCH_VL1RE64V 255
30.38.1.614 INSN_MATCH_VL1RE8V 255
30.38.1.615 INSN_MATCH_VL2RE16V 255
30.38.1.616 INSN_MATCH_VL2RE32V 255
30.38.1.617 INSN_MATCH_VL2RE64V 255
30.38.1.618 INSN_MATCH_VL2RE8V 255
30.38.1.619 INSN_MATCH_VL4RE16V 255
30.38.1.620 INSN_MATCH_VL4RE32V 255

30.38.1.621 INSN_MATCH_VL4RE64V 255
30.38.1.622 INSN_MATCH_VL4RE8V 255
30.38.1.623 INSN_MATCH_VL8RE16V 255
30.38.1.624 INSN_MATCH_VL8RE32V 256
30.38.1.625 INSN_MATCH_VL8RE64V 256
30.38.1.626 INSN_MATCH_VL8RE8V 256
30.38.1.627 INSN_MATCH_VLE16FFV 256
30.38.1.628 INSN_MATCH_VLE16V 256
30.38.1.629 INSN_MATCH_VLE32FFV 256
30.38.1.630 INSN_MATCH_VLE32V 256
30.38.1.631 INSN_MATCH_VLE64FFV 256
30.38.1.632 INSN_MATCH_VLE64V 256
30.38.1.633 INSN_MATCH_VLOXEI16V 256
30.38.1.634 INSN_MATCH_VLOXEI32V 256
30.38.1.635 INSN_MATCH_VLOXEI64V 256
30.38.1.636 INSN_MATCH_VLOXSEG 256
30.38.1.637 INSN_MATCH_VLSE16V 256
30.38.1.638 INSN_MATCH_VLSE32V 257
30.38.1.639 INSN_MATCH_VLSE64V 257
30.38.1.640 INSN_MATCH_VLSEG 257
30.38.1.641 INSN_MATCH_VLSEGFF 257
30.38.1.642 INSN_MATCH_VLSSEG 257
30.38.1.643 INSN_MATCH_VLUXEI16V 257
30.38.1.644 INSN_MATCH_VLUXEI32V 257
30.38.1.645 INSN_MATCH_VLUXEI64V 257
30.38.1.646 INSN_MATCH_VLUXSEG 257
30.38.1.647 INSN_MATCH_VS1RV 257
30.38.1.648 INSN_MATCH_VS2RV 257
30.38.1.649 INSN_MATCH_VS4RV 258
30.38.1.650 INSN_MATCH_VS8RV 258
30.38.1.651 INSN_MATCH_VSE16V 258
30.38.1.652 INSN_MATCH_VSE32V 258
30.38.1.653 INSN_MATCH_VSE64V 258
30.38.1.654 INSN_MATCH_VSOXEI16V 258
30.38.1.655 INSN_MATCH_VSOXEI32V 258
30.38.1.656 INSN_MATCH_VSOXEI64V 258
30.38.1.657 INSN_MATCH_VSOXSEG 258
30.38.1.658 INSN_MATCH_VSSE16V 258
30.38.1.659 INSN_MATCH_VSSE32V 258
30.38.1.660 INSN_MATCH_VSSE64V 258
30.38.1.661 INSN_MATCH_VSSEG 258
30.38.1.662 INSN_MATCH_VSSSEG 259

30.38.1.663 INSN_MATCH_VSUXEI16V . . . 259
30.38.1.664 INSN_MATCH_VSUXEI32V . . . 259
30.38.1.665 INSN_MATCH_VSUXEI64V . . . 259
30.38.1.666 INSN_MATCH_VSUXSEG . . . 259
30.38.1.667 INSN_MATCH_WFI . . . 259
30.38.1.668 IRQ_M_EXT . . . 259
30.38.1.669 IRQ_M_SOFT . . . 259
30.38.1.670 IRQ_M_TIMER . . . 259
30.38.1.671 IRQ_PMU_OVF . . . 259
30.38.1.672 IRQ_S_EXT . . . 259
30.38.1.673 IRQ_S_GEXT . . . 259
30.38.1.674 IRQ_S_SOFT . . . 259
30.38.1.675 IRQ_S_TIMER . . . 260
30.38.1.676 IRQ_VS_EXT . . . 260
30.38.1.677 IRQ_VS_SOFT . . . 260
30.38.1.678 IRQ_VS_TIMER . . . 260
30.38.1.679 IS_FAULT_ONLY_FIRST_LOAD . . . 260
30.38.1.680 IS_FAULT_ONLY_FIRST_MATCH . . . 260
30.38.1.681 IS_INDEXED_LOAD . . . 260
30.38.1.682 IS_INDEXED_MATCH . . . 260
30.38.1.683 IS_INDEXED_STORE . . . 260
30.38.1.684 IS_MASKED . . . 261
30.38.1.685 IS_STRIDE_LOAD . . . 261
30.38.1.686 IS_STRIDE_MATCH . . . 261
30.38.1.687 IS_STRIDE_STORE . . . 261
30.38.1.688 IS_UNIT_STRIDE_LOAD . . . 261
30.38.1.689 IS_UNIT_STRIDE_MATCH . . . 262
30.38.1.690 IS_UNIT_STRIDE_STORE . . . 262
30.38.1.691 IS_VECTOR_INSN_MATCH . . . 262
30.38.1.692 IS_VECTOR_LOAD_STORE . . . 262
30.38.1.693 IS_WHOLE_REG_LOAD . . . 263
30.38.1.694 IS_WHOLE_REG_MATCH . . . 263
30.38.1.695 IS_WHOLE_REG_STORE . . . 263
30.38.1.696 LOG_REGBYTES . . . 263
30.38.1.697 MASK_CSR . . . 263
30.38.1.698 MASK_FUNCT3 . . . 263
30.38.1.699 MASK_RS1 . . . 263
30.38.1.700 MCAUSE_IRQ_MASK . . . 263
30.38.1.701 MHPMEVENT_MINH . . . 263
30.38.1.702 MHPMEVENT_OF . . . 264
30.38.1.703 MHPMEVENT_SINH . . . 264
30.38.1.704 MHPMEVENT_SSCOF_MASK . . . 264

30.38.1.705 MHPMEVENT_UINH . . . 264
30.38.1.706 MHPMEVENT_VSINH . . . 264
30.38.1.707 MHPMEVENT_VUINH . . . 264
30.38.1.708 MHPMEVENTH_MINH . . . 264
30.38.1.709 MHPMEVENTH_OF . . . 264
30.38.1.710 MHPMEVENTH_SINH . . . 264
30.38.1.711 MHPMEVENTH_UINH . . . 264
30.38.1.712 MHPMEVENTH_VSINH . . . 264
30.38.1.713 MHPMEVENTH_VUINH . . . 264
30.38.1.714 MIP_LCOFIP . . . 264
30.38.1.715 MIP_MEIP . . . 264
30.38.1.716 MIP_MSIP . . . 264
30.38.1.717 MIP_MTIP . . . 265
30.38.1.718 MIP_SEIP . . . 265
30.38.1.719 MIP_SGEIP . . . 265
30.38.1.720 MIP_SSIP . . . 265
30.38.1.721 MIP_STIP . . . 265
30.38.1.722 MIP_VSEIP . . . 265
30.38.1.723 MIP_VSSIP . . . 265
30.38.1.724 MIP_VSTIP . . . 265
30.38.1.725 MSECCFG_MML . . . 265
30.38.1.726 MSECCFG_MML_SHIFT . . . 265
30.38.1.727 MSECCFG_MMWP . . . 265
30.38.1.728 MSECCFG_MMWP_SHIFT . . . 265
30.38.1.729 MSECCFG_RLB . . . 265
30.38.1.730 MSECCFG_RLB_SHIFT . . . 265
30.38.1.731 MSECCFG_SSEED . . . 265
30.38.1.732 MSECCFG_SSEED_SHIFT . . . 266
30.38.1.733 MSECCFG_USEED . . . 266
30.38.1.734 MSECCFG_USEED_SHIFT . . . 266
30.38.1.735 MSTATUS32_SD [1/2] . . . 266
30.38.1.736 MSTATUS32_SD [2/2] . . . 266
30.38.1.737 MSTATUS64_SD . . . 266
30.38.1.738 MSTATUS_FS . . . 266
30.38.1.739 MSTATUS_MIE . . . 266
30.38.1.740 MSTATUS_MPIE . . . 266
30.38.1.741 MSTATUS_MPP . . . 266
30.38.1.742 MSTATUS_MPP_SHIFT . . . 266
30.38.1.743 MSTATUS_MPRV . . . 266
30.38.1.744 MSTATUS_MXR . . . 266
30.38.1.745 MSTATUS_SD . . . 266
30.38.1.746 MSTATUS_SIE . . . 266

30.38.1.747 MSTATUS_SPELP . . . 267
30.38.1.748 MSTATUS_SPIE . . . 267
30.38.1.749 MSTATUS_SPIE_SHIFT . . . 267
30.38.1.750 MSTATUS_SPP . . . 267
30.38.1.751 MSTATUS_SPP_SHIFT . . . 267
30.38.1.752 MSTATUS_SUM . . . 267
30.38.1.753 MSTATUS_TSR . . . 267
30.38.1.754 MSTATUS_TVM . . . 267
30.38.1.755 MSTATUS_TW . . . 267
30.38.1.756 MSTATUS_UBE . . . 267
30.38.1.757 MSTATUS_VS . . . 267
30.38.1.758 MSTATUS_XS . . . 267
30.38.1.759 MSTATUSH_GVA . . . 267
30.38.1.760 MSTATUSH_GVA_SHIFT . . . 267
30.38.1.761 MSTATUSH_MBE . . . 267
30.38.1.762 MSTATUSH_MDT . . . 268
30.38.1.763 MSTATUSH_MPELP . . . 268
30.38.1.764 MSTATUSH_MPV . . . 268
30.38.1.765 MSTATUSH_SBE . . . 268
30.38.1.766 PMP_A . . . 268
30.38.1.767 PMP_A_NA4 . . . 268
30.38.1.768 PMP_A_NAPOT . . . 268
30.38.1.769 PMP_A_TOR . . . 268
30.38.1.770 PMP_ADDR_MASK . . . 268
30.38.1.771 PMP_COUNT . . . 268
30.38.1.772 PMP_L . . . 268
30.38.1.773 PMP_R . . . 268
30.38.1.774 PMP_SHIFT . . . 268
30.38.1.775 PMP_W . . . 268
30.38.1.776 PMP_X . . . 268
30.38.1.777 PRV_M . . . 269
30.38.1.778 PRV_S . . . 269
30.38.1.779 PRV_U . . . 269
30.38.1.780 REG_MASK . . . 269
30.38.1.781 REG_OFFSET . . . 269
30.38.1.782 REG_PTR . . . 269
30.38.1.783 REGBYTES . . . 269
30.38.1.784 RV_X . . . 269
30.38.1.785 RVC_LD_IMM . . . 269
30.38.1.786 RVC_LDSP_IMM . . . 269
30.38.1.787 RVC_LW_IMM . . . 270
30.38.1.788 RVC_LWSP_IMM . . . 270

30.38.1.789 RVC_RS1S . . . 270
30.38.1.790 RVC_RS2 . . . 270
30.38.1.791 RVC_RS2S . . . 270
30.38.1.792 RVC_SDSP_IMM . . . 270
30.38.1.793 RVC_SWSP_IMM . . . 270
30.38.1.794 SATP32_ASID . . . 270
30.38.1.795 SATP32_MODE . . . 270
30.38.1.796 SATP32_PPN . . . 270
30.38.1.797 SATP64_ASID . . . 270
30.38.1.798 SATP64_MODE . . . 271
30.38.1.799 SATP64_PPN . . . 271
30.38.1.800 SATP_MODE . . . 271
30.38.1.801 SATP_MODE_OFF . . . 271
30.38.1.802 SATP_MODE_SV32 . . . 271
30.38.1.803 SATP_MODE_SV39 . . . 271
30.38.1.804 SATP_MODE_SV48 . . . 271
30.38.1.805 SATP_MODE_SV57 . . . 271
30.38.1.806 SATP_MODE_SV64 . . . 271
30.38.1.807 SET_RD . . . 271
30.38.1.808 SH_MEW . . . 271
30.38.1.809 SH_RD . . . 271
30.38.1.810 SH_RS1 . . . 271
30.38.1.811 SH_RS2 . . . 271
30.38.1.812 SH_RS2C . . . 272
30.38.1.813 SH_VD . . . 272
30.38.1.814 SH_VIEW . . . 272
30.38.1.815 SH_VM . . . 272
30.38.1.816 SH_VS2 . . . 272
30.38.1.817 SH_VSEW . . . 272
30.38.1.818 SHIFT_CSR . . . 272
30.38.1.819 SHIFT_FUNCT3 . . . 272
30.38.1.820 SHIFT_RIGHT . . . 272
30.38.1.821 SIP_SSIP . . . 272
30.38.1.822 SIP_STIP . . . 272
30.38.1.823 SMSTATEEN0_AIA . . . 272
30.38.1.824 SMSTATEEN0_AIA_SHIFT . . . 272
30.38.1.825 SMSTATEEN0_CONTEXT . . . 272
30.38.1.826 SMSTATEEN0_CONTEXT_SHIFT . . . 272
30.38.1.827 SMSTATEEN0_CS . . . 273
30.38.1.828 SMSTATEEN0_CS_SHIFT . . . 273
30.38.1.829 SMSTATEEN0_FCSR . . . 273
30.38.1.830 SMSTATEEN0_FCSR_SHIFT . . . 273

30.38.1.831 SMSTATEEN0_HSENVCFG . . . 273
30.38.1.832 SMSTATEEN0_HSENVCFG_SHIFT . . . 273
30.38.1.833 SMSTATEEN0_IMSIC . . . 273
30.38.1.834 SMSTATEEN0_IMSIC_SHIFT . . . 273
30.38.1.835 SMSTATEEN0_SVSLCT . . . 273
30.38.1.836 SMSTATEEN0_SVSLCT_SHIFT . . . 273
30.38.1.837 SMSTATEEN_MAX_COUNT . . . 273
30.38.1.838 SMSTATEEN_STATEN . . . 273
30.38.1.839 SMSTATEEN_STATEN_SHIFT . . . 273
30.38.1.840 SSTATUS32_SD . . . 273
30.38.1.841 SSTATUS64_SD . . . 273
30.38.1.842 SSTATUS64_UXL . . . 274
30.38.1.843 SSTATUS_FS . . . 274
30.38.1.844 SSTATUS_MXR . . . 274
30.38.1.845 SSTATUS_SD . . . 274
30.38.1.846 SSTATUS_SIE . . . 274
30.38.1.847 SSTATUS_SPIE . . . 274
30.38.1.848 SSTATUS_SPIE_SHIFT . . . 274
30.38.1.849 SSTATUS_SPP . . . 274
30.38.1.850 SSTATUS_SPP_SHIFT . . . 274
30.38.1.851 SSTATUS_SUM . . . 274
30.38.1.852 SSTATUS_VS . . . 274
30.38.1.853 SSTATUS_XS . . . 274
30.38.1.854 TOPI_IID_MASK . . . 274
30.38.1.855 TOPI_IID_SHIFT . . . 274
30.38.1.856 TOPI_IPRIO_MASK . . . 274
30.38.1.857 VD_MASK . . . 275
30.38.1.858 VIEW_MASK . . . 275
30.38.1.859 VLMUL_MASK . . . 275
30.38.1.860 VM_MASK . . . 275
30.38.1.861 VS2_MASK . . . 275
30.38.1.862 VSEW_MASK . . . 275
30.39 riscv_encoding.h . . . 275
30.40 include/sbi/riscv_fp.h File Reference . . . 291
30.40.1 Macro Definition Documentation . . . 292
30.40.1.1 GET_PRECISION . . . 292
30.40.1.2 PRECISION_D . . . 292
30.40.1.3 PRECISION_S . . . 292
30.41 riscv_fp.h . . . 292
30.42 include/sbi/riscv_io.h File Reference . . . 294
30.42.1 Macro Definition Documentation . . . 295
30.42.1.1 __io_ar . . . 295

30.42.1.2 __io_aw . . . 295
30.42.1.3 __io_br . . . 295
30.42.1.4 __io_bw . . . 295
30.42.1.5 __io_rar . . . 295
30.42.1.6 __io_raw . . . 295
30.42.1.7 __io_rbr . . . 295
30.42.1.8 __io_rbw . . . 295
30.42.1.9 readb . . . 295
30.42.1.10 readb_relaxed . . . 295
30.42.1.11 readl . . . 295
30.42.1.12 readl_relaxed . . . 296
30.42.1.13 readq . . . 296
30.42.1.14 readq_relaxed . . . 296
30.42.1.15 readw . . . 296
30.42.1.16 readw_relaxed . . . 296
30.42.1.17 writeb . . . 296
30.42.1.18 writeb_relaxed . . . 296
30.42.1.19 writel . . . 296
30.42.1.20 writel_relaxed . . . 296
30.42.1.21 writeq . . . 296
30.42.1.22 writeq_relaxed . . . 296
30.42.1.23 writew . . . 297
30.42.1.24 writew_relaxed . . . 297
30.42.2 Function Documentation . . . 297
30.42.2.1 __raw_readb() . . . 297
30.42.2.2 __raw_readl() . . . 297
30.42.2.3 __raw_readq() . . . 297
30.42.2.4 __raw_readw() . . . 297
30.42.2.5 __raw_writeb() . . . 297
30.42.2.6 __raw_writel() . . . 297
30.42.2.7 __raw_writeq() . . . 297
30.42.2.8 __raw_writew() . . . 297
30.43 riscv_io.h . . . 298
30.44 include/sbi/riscv_locks.h File Reference . . . 299
30.44.1 Macro Definition Documentation . . . 300
30.44.1.1 __SPIN_LOCK_UNLOCKED . . . 300
30.44.1.2 DEFINE_SPIN_LOCK . . . 300
30.44.1.3 SPIN_LOCK_INIT . . . 300
30.44.1.4 SPIN_LOCK_INITIALIZER . . . 300
30.44.1.5 TICKET_SHIFT . . . 300
30.44.2 Function Documentation . . . 300
30.44.2.1 spin_lock() . . . 300

30.44.2.2 spin_lock_check() . . . 301
30.44.2.3 spin_trylock() . . . 302
30.44.2.4 spin_unlock() . . . 302
30.45 riscv_locks.h . . . 303
30.46 include/sbi/sbi_bitmap.h File Reference . . . 304
30.46.1 Macro Definition Documentation . . . 305
30.46.1.1 BITMAP_FIRST_WORD_MASK . . . 305
30.46.1.2 BITMAP_LAST_WORD_MASK . . . 305
30.46.1.3 DECLARE_BITMAP . . . 305
30.46.1.4 DEFINE_BITMAP . . . 305
30.46.1.5 small_const_nbits . . . 305
30.46.2 Function Documentation . . . 306
30.46.2.1 __bitmap_and() . . . 306
30.46.2.2 __bitmap_or() . . . 306
30.46.2.3 __bitmap_xor() . . . 306
30.46.2.4 bitmap_and() . . . 307
30.46.2.5 bitmap_clear() . . . 307
30.46.2.6 bitmap_copy() . . . 307
30.46.2.7 bitmap_estimate_size() . . . 308
30.46.2.8 bitmap_fill() . . . 308
30.46.2.9 bitmap_or() . . . 308
30.46.2.10 bitmap_set() . . . 309
30.46.2.11 bitmap_test() . . . 309
30.46.2.12 bitmap_xor() . . . 309
30.46.2.13 bitmap_zero() . . . 310
30.46.2.14 bitmap_zero_except() . . . 310
30.47 sbi_bitmap.h . . . 311
30.48 include/sbi/sbi_bitops.h File Reference . . . 312
30.48.1 Macro Definition Documentation . . . 313
30.48.1.1 BIT . . . 313
30.48.1.2 BIT_ALIGN . . . 313
30.48.1.3 BIT_MASK . . . 314
30.48.1.4 BIT_ULL . . . 314
30.48.1.5 BIT_WORD . . . 314
30.48.1.6 BIT_WORD_OFFSET . . . 314
30.48.1.7 BITS_PER_LONG . . . 314
30.48.1.8 BITS_PER_LONG_LONG . . . 314
30.48.1.9 BITS_TO_LONGS . . . 314
30.48.1.10 EXTRACT_FIELD . . . 314
30.48.1.11 for_each_clear_bit . . . 314
30.48.1.12 for_each_clear_bit_from . . . 314
30.48.1.13 for_each_set_bit . . . 315

30.48.1.14 for_each_set_bit_from . . . 315
30.48.1.15 GENMASK . . . 315
30.48.1.16 GENMASK_ULL . . . 315
30.48.1.17 INSERT_FIELD . . . 315
30.48.1.18 sbi_ffz . . . 315
30.48.2 Function Documentation . . . 315
30.48.2.1 __change_bit() . . . 315
30.48.2.2 __clear_bit() . . . 315
30.48.2.3 __set_bit() . . . 316
30.48.2.4 __test_and_clear_bit() . . . 316
30.48.2.5 __test_and_set_bit() . . . 316
30.48.2.6 __test_bit() . . . 316
30.48.2.7 find_first_bit() . . . 317
30.48.2.8 find_first_zero_bit() . . . 318
30.48.2.9 find_last_bit() . . . 318
30.48.2.10 find_next_bit() . . . 318
30.48.2.11 find_next_zero_bit() . . . 319
30.48.2.12 sbi_ffs() . . . 319
30.48.2.13 sbi_fls() . . . 319
30.48.2.14 sbi_popcount() . . . 320
30.49 sbi_bitops.h . . . 320
30.50 include/sbi/sbi_byteorder.h File Reference . . . 323
30.50.1 Macro Definition Documentation . . . 323
30.50.1.1 _conv_cast . . . 323
30.50.1.2 be16_to_cpu . . . 323
30.50.1.3 be32_to_cpu . . . 324
30.50.1.4 be64_to_cpu . . . 324
30.50.1.5 BSWAP16 . . . 324
30.50.1.6 BSWAP32 . . . 324
30.50.1.7 BSWAP64 . . . 324
30.50.1.8 cpu_to_be16 . . . 324
30.50.1.9 cpu_to_be32 . . . 324
30.50.1.10 cpu_to_be64 . . . 324
30.50.1.11 cpu_to_le16 . . . 324
30.50.1.12 cpu_to_le32 . . . 325
30.50.1.13 cpu_to_le64 . . . 325
30.50.1.14 le16_to_cpu . . . 325
30.50.1.15 le32_to_cpu . . . 325
30.50.1.16 le64_to_cpu . . . 325
30.51 sbi_byteorder.h . . . 325
30.52 include/sbi/sbi_console.h File Reference . . . 326
30.52.1 Macro Definition Documentation . . . 327

30.52.1.1 __printf . . . 327
30.52.1.2 SBI_ASSERT . . . 327
30.52.2 Function Documentation . . . 327
30.52.2.1 __printf() [1/3] . . . 327
30.52.2.2 __printf() [2/3] . . . 327
30.52.2.3 __printf() [3/3] . . . 327
30.52.2.4 sbi_console_get_device() . . . 328
30.52.2.5 sbi_console_set_device() . . . 328
30.52.2.6 sbi_getc() . . . 329
30.52.2.7 sbi_gets() . . . 329
30.52.2.8 sbi_isprintable() . . . 330
30.52.2.9 sbi_ngets() . . . 330
30.52.2.10 sbi_nputs() . . . 330
30.52.2.11 sbi_putc() . . . 331
30.52.2.12 sbi_puts() . . . 332
30.52.3 Variable Documentation . . . 333
30.52.3.1 format . . . 333
30.52.3.2 out_sz . . . 333
30.53 sbi_console.h . . . 333
30.54 include/sbi/sbi_const.h File Reference . . . 334
30.54.1 Macro Definition Documentation . . . 334
30.54.1.1 __AC . . . 334
30.54.1.2 __STR . . . 334
30.54.1.3 _AC . . . 334
30.54.1.4 _AT . . . 334
30.54.1.5 _BITUL . . . 334
30.54.1.6 _BITULL . . . 334
30.54.1.7 _UL . . . 335
30.54.1.8 _ULL . . . 335
30.54.1.9 STRINGIFY . . . 335
30.54.1.10 UL . . . 335
30.54.1.11 ULL . . . 335
30.55 sbi_const.h . . . 335
30.56 include/sbi/sbi_cppc.h File Reference . . . 336
30.56.1 Function Documentation . . . 336
30.56.1.1 sbi_cppc_get_device() . . . 336
30.56.1.2 sbi_cppc_probe() . . . 337
30.56.1.3 sbi_cppc_read() . . . 337
30.56.1.4 sbi_cppc_set_device() . . . 338
30.56.1.5 sbi_cppc_write() . . . 338
30.57 sbi_cppc.h . . . 339
30.58 include/sbi/sbi_csr_detect.h File Reference . . . 339

30.58.1 Macro Definition Documentation . . . 340
30.58.1.1 csr_read_allowed . . . 340
30.58.1.2 csr_write_allowed . . . 341
30.59 sbi_csr_detect.h . . . 341
30.60 include/sbi/sbi_dbtr.h File Reference . . . 342
30.60.1 Macro Definition Documentation . . . 343
30.60.1.1 SBI_DBTR_SHMEM_ALIGN_MASK . . . 343
30.60.1.2 SET_TRIG_HW_INDEX . . . 343
30.60.1.3 TDATA1_GET_TYPE . . . 343
30.60.2 Enumeration Type Documentation . . . 344
30.60.2.1 anonymous enum . . . 344
30.60.2.2 anonymous enum . . . 344
30.60.3 Function Documentation . . . 344
30.60.3.1 sbi_dbtr_disable_trig() . . . 344
30.60.3.2 sbi_dbtr_enable_trig() . . . 345
30.60.3.3 sbi_dbtr_get_total_triggers() . . . 345
30.60.3.4 sbi_dbtr_init() . . . 346
30.60.3.5 sbi_dbtr_install_trig() . . . 346
30.60.3.6 sbi_dbtr_num_trig() . . . 347
30.60.3.7 sbi_dbtr_read_trig() . . . 347
30.60.3.8 sbi_dbtr_setup_shmem() . . . 348
30.60.3.9 sbi_dbtr_supported() . . . 348
30.60.3.10 sbi_dbtr_uninstall_trig() . . . 348
30.60.3.11 sbi_dbtr_update_trig() . . . 349
30.61 sbi_dbtr.h . . . 350
30.62 include/sbi/sbi_domain.h File Reference . . . 351
30.62.1 Macro Definition Documentation . . . 353
30.62.1.1 sbi_domain_for_each . . . 353
30.62.1.2 sbi_domain_for_each_memregion . . . 353
30.62.1.3 SBI_DOMAIN_MEMREGION_ACCESS_MASK . . . 353
30.62.1.4 SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE . . . 353
30.62.1.5 SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS . . . 353
30.62.1.6 SBI_DOMAIN_MEMREGION_ENF_READABLE . . . 353
30.62.1.7 SBI_DOMAIN_MEMREGION_ENF_WRITABLE . . . 354
30.62.1.8 SBI_DOMAIN_MEMREGION_EXECUTABLE . . . 354
30.62.1.9 SBI_DOMAIN_MEMREGION_IS_SHARED . . . 354
30.62.1.10 SBI_DOMAIN_MEMREGION_IS_SUR_MR . . . 354
30.62.1.11 SBI_DOMAIN_MEMREGION_IS_SUR_MRW . . . 354
30.62.1.12 SBI_DOMAIN_MEMREGION_IS_SURW_MRW . . . 354
30.62.1.13 SBI_DOMAIN_MEMREGION_IS_SUX_MRX . . . 354
30.62.1.14 SBI_DOMAIN_MEMREGION_IS_SUX_MX . . . 354
30.62.1.15 SBI_DOMAIN_MEMREGION_M_ACCESS_MASK . . . 355

30.62.1.16 SBI_DOMAIN_MEMREGION_M_EXECUTABLE . . . 355
30.62.1.17 SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS . . . 355
30.62.1.18 SBI_DOMAIN_MEMREGION_M_READABLE . . . 355
30.62.1.19 SBI_DOMAIN_MEMREGION_M_RWX . . . 355
30.62.1.20 SBI_DOMAIN_MEMREGION_M_WRITABLE . . . 355
30.62.1.21 SBI_DOMAIN_MEMREGION_MMIO . . . 355
30.62.1.22 SBI_DOMAIN_MEMREGION_READABLE . . . 355
30.62.1.23 SBI_DOMAIN_MEMREGION_SHARED_RDONLY . . . 355
30.62.1.24 SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW . . . 355
30.62.1.25 SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW . . . 355
30.62.1.26 SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX . . . 356
30.62.1.27 SBI_DOMAIN_MEMREGION_SHARED_SUX_MX . . . 356
30.62.1.28 SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK . . . 356
30.62.1.29 SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT . . . 356
30.62.1.30 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE . . . 356
30.62.1.31 SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS . . . 356
30.62.1.32 SBI_DOMAIN_MEMREGION_SU_READABLE . . . 356
30.62.1.33 SBI_DOMAIN_MEMREGION_SU_RWX . . . 356
30.62.1.34 SBI_DOMAIN_MEMREGION_SU_WRITABLE . . . 356
30.62.1.35 SBI_DOMAIN_MEMREGION_WRITEABLE . . . 356
30.62.1.36 sbi_domain_thishart_ptr . . . 356
30.62.2 Enumeration Type Documentation . . . 357
30.62.2.1 sbi_domain_access . . . 357
30.62.3 Function Documentation . . . 357
30.62.3.1 sbi_domain_check_addr() . . . 357
30.62.3.2 sbi_domain_check_addr_range() . . . 358
30.62.3.3 sbi_domain_dump() . . . 359
30.62.3.4 sbi_domain_dump_all() . . . 360
30.62.3.5 sbi_domain_finalize() . . . 360
30.62.3.6 sbi_domain_get_assigned_hartmask() . . . 361
30.62.3.7 sbi_domain_init() . . . 362
30.62.3.8 sbi_domain_is_assigned_hart() . . . 362
30.62.3.9 sbi_domain_memregion_init() . . . 363
30.62.3.10 sbi_domain_register() . . . 364
30.62.3.11 sbi_domain_root_add_memrange() . . . 365
30.62.3.12 sbi_hartindex_to_domain() . . . 366
30.62.3.13 sbi_update_hartindex_to_domain() . . . 367
30.62.4 Variable Documentation . . . 367
30.62.4.1 domain_list . . . 367
30.62.4.2 root . . . 367
30.63 sbi_domain.h . . . 367
30.64 include/sbi/sbi_domain_context.h File Reference . . . 370

30.64.1 Function Documentation . . . 370
30.64.1.1 sbi_domain_context_deinit() . . . 370
30.64.1.2 sbi_domain_context_enter() . . . 371
30.64.1.3 sbi_domain_context_exit() . . . 372
30.64.1.4 sbi_domain_context_init() . . . 372
30.65 sbi_domain_context.h . . . 373
30.66 include/sbi/sbi_domain_data.h File Reference . . . 373
30.66.1 Macro Definition Documentation . . . 374
30.66.1.1 SBI_DOMAIN_MAX_DATA_PTRS . . . 374
30.66.2 Function Documentation . . . 374
30.66.2.1 sbi_domain_cleanup_data() . . . 374
30.66.2.2 sbi_domain_data_ptr() . . . 374
30.66.2.3 sbi_domain_register_data() . . . 375
30.66.2.4 sbi_domain_setup_data() . . . 376
30.66.2.5 sbi_domain_unregister_data() . . . 376
30.67 sbi_domain_data.h . . . 377
30.68 include/sbi/sbi_ecall.h File Reference . . . 377
30.68.1 Macro Definition Documentation . . . 378
30.68.1.1 SBI_ECALL_VERSION_MAJOR . . . 378
30.68.1.2 SBI_ECALL_VERSION_MINOR . . . 379
30.68.1.3 SBI_OPENSBI_IMPID . . . 379
30.68.2 Function Documentation . . . 379
30.68.2.1 sbi_ecall_find_extension() . . . 379
30.68.2.2 sbi_ecall_get_extensions_str() . . . 379
30.68.2.3 sbi_ecall_get_impid() . . . 379
30.68.2.4 sbi_ecall_handler() . . . 380
30.68.2.5 sbi_ecall_init() . . . 380
30.68.2.6 sbi_ecall_register_extension() . . . 381
30.68.2.7 sbi_ecall_set_impid() . . . 382
30.68.2.8 sbi_ecall_unregister_extension() . . . 382
30.68.2.9 sbi_ecall_version_major() . . . 383
30.68.2.10 sbi_ecall_version_minor() . . . 383
30.69 sbi_ecall.h . . . 383
30.70 include/sbi/sbi_ecall_interface.h File Reference . . . 385
30.70.1 Macro Definition Documentation . . . 390
30.70.1.1 SBI_ERR_ALREADY_AVAILABLE . . . 390
30.70.1.2 SBI_ERR_ALREADY_STARTED . . . 390
30.70.1.3 SBI_ERR_ALREADY_STOPPED . . . 390
30.70.1.4 SBI_ERR_BAD_RANGE . . . 390
30.70.1.5 SBI_ERR_DENIED . . . 390
30.70.1.6 SBI_ERR_FAILED . . . 390
30.70.1.7 SBI_ERR_INVALID_ADDRESS . . . 390

30.70.1.8 SBI_ERR_INVALID_PARAM . . . 390
30.70.1.9 SBI_ERR_INVALID_STATE . . . 390
30.70.1.10 SBI_ERR_IO . . . 391
30.70.1.11 SBI_ERR_NO_SHMEM . . . 391
30.70.1.12 SBI_ERR_NOT_SUPPORTED . . . 391
30.70.1.13 SBI_ERR_TIMEOUT . . . 391
30.70.1.14 SBI_EXT_0_1_CLEAR_IPI . . . 391
30.70.1.15 SBI_EXT_0_1_CONSOLE_GETCHAR . . . 391
30.70.1.16 SBI_EXT_0_1_CONSOLE_PUTCHAR . . . 391
30.70.1.17 SBI_EXT_0_1_REMOTE_FENCE_I . . . 391
30.70.1.18 SBI_EXT_0_1_REMOTE_SFENCE_VMA . . . 391
30.70.1.19 SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID . . . 391
30.70.1.20 SBI_EXT_0_1_SEND_IPI . . . 391
30.70.1.21 SBI_EXT_0_1_SET_TIMER . . . 391
30.70.1.22 SBI_EXT_0_1_SHUTDOWN . . . 391
30.70.1.23 SBI_EXT_BASE . . . 391
30.70.1.24 SBI_EXT_BASE_GET_IMP_ID . . . 391
30.70.1.25 SBI_EXT_BASE_GET_IMP_VERSION . . . 392
30.70.1.26 SBI_EXT_BASE_GET_MARCHID . . . 392
30.70.1.27 SBI_EXT_BASE_GET_MIMPID . . . 392
30.70.1.28 SBI_EXT_BASE_GET_MVENDORID . . . 392
30.70.1.29 SBI_EXT_BASE_GET_SPEC_VERSION . . . 392
30.70.1.30 SBI_EXT_BASE_PROBE_EXT . . . 392
30.70.1.31 SBI_EXT_CPPC . . . 392
30.70.1.32 SBI_EXT_CPPC_PROBE . . . 392
30.70.1.33 SBI_EXT_CPPC_READ . . . 392
30.70.1.34 SBI_EXT_CPPC_READ_HI . . . 392
30.70.1.35 SBI_EXT_CPPC_WRITE . . . 392
30.70.1.36 SBI_EXT_DBCN . . . 392
30.70.1.37 SBI_EXT_DBCN_CONSOLE_READ . . . 392
30.70.1.38 SBI_EXT_DBCN_CONSOLE_WRITE . . . 392
30.70.1.39 SBI_EXT_DBCN_CONSOLE_WRITE_BYTE . . . 392
30.70.1.40 SBI_EXT_DBTR . . . 393
30.70.1.41 SBI_EXT_DBTR_NUM_TRIGGERS . . . 393
30.70.1.42 SBI_EXT_DBTR_SETUP_SHMEM . . . 393
30.70.1.43 SBI_EXT_DBTR_TRIGGER_DISABLE . . . 393
30.70.1.44 SBI_EXT_DBTR_TRIGGER_ENABLE . . . 393
30.70.1.45 SBI_EXT_DBTR_TRIGGER_INSTALL . . . 393
30.70.1.46 SBI_EXT_DBTR_TRIGGER_READ . . . 393
30.70.1.47 SBI_EXT_DBTR_TRIGGER_UNINSTALL . . . 393
30.70.1.48 SBI_EXT_DBTR_TRIGGER_UPDATE . . . 393
30.70.1.49 SBI_EXT_FIRMWARE_END . . . 393

30.70.1.50 SBI_EXT_FIRMWARE_START . . . 393
30.70.1.51 SBI_EXT_FWFT . . . 393
30.70.1.52 SBI_EXT_FWFT_GET . . . 393
30.70.1.53 SBI_EXT_FWFT_SET . . . 393
30.70.1.54 SBI_EXT_HSM . . . 393
30.70.1.55 SBI_EXT_HSM_HART_GET_STATUS . . . 394
30.70.1.56 SBI_EXT_HSM_HART_START . . . 394
30.70.1.57 SBI_EXT_HSM_HART_STOP . . . 394
30.70.1.58 SBI_EXT_HSM_HART_SUSPEND . . . 394
30.70.1.59 SBI_EXT_IPI . . . 394
30.70.1.60 SBI_EXT_IPI_SEND_IPI . . . 394
30.70.1.61 SBI_EXT_MPXY . . . 394
30.70.1.62 SBI_EXT_MPXY_GET_CHANNEL_IDS . . . 394
30.70.1.63 SBI_EXT_MPXY_GET_NOTIFICATION_EVENTS . . . 394
30.70.1.64 SBI_EXT_MPXY_READ_ATTRS . . . 394
30.70.1.65 SBI_EXT_MPXY_SEND_MSG_NO_RESP . . . 394
30.70.1.66 SBI_EXT_MPXY_SEND_MSG_WITH_RESP . . . 394
30.70.1.67 SBI_EXT_MPXY_SET_SHMEM . . . 394
30.70.1.68 SBI_EXT_MPXY_WRITE_ATTRS . . . 394
30.70.1.69 SBI_EXT_PMU . . . 394
30.70.1.70 SBI_EXT_PMU_COUNTER_CFG_MATCH . . . 395
30.70.1.71 SBI_EXT_PMU_COUNTER_FW_READ . . . 395
30.70.1.72 SBI_EXT_PMU_COUNTER_FW_READ_HI . . . 395
30.70.1.73 SBI_EXT_PMU_COUNTER_GET_INFO . . . 395
30.70.1.74 SBI_EXT_PMU_COUNTER_START . . . 395
30.70.1.75 SBI_EXT_PMU_COUNTER_STOP . . . 395
30.70.1.76 SBI_EXT_PMU_EVENT_GET_INFO . . . 395
30.70.1.77 SBI_EXT_PMU_NUM_COUNTERS . . . 395
30.70.1.78 SBI_EXT_PMU_SNAPSHOT_SET_SHMEM . . . 395
30.70.1.79 SBI_EXT_RFENCE . . . 395
30.70.1.80 SBI_EXT_RFENCE_REMOTE_FENCE_I . . . 395
30.70.1.81 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA . . . 395
30.70.1.82 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID . . . 395
30.70.1.83 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA . . . 395
30.70.1.84 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID . . . 395
30.70.1.85 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA . . . 396
30.70.1.86 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID . . . 396
30.70.1.87 SBI_EXT_SRST . . . 396
30.70.1.88 SBI_EXT_SRST_RESET . . . 396
30.70.1.89 SBI_EXT_SSE . . . 396
30.70.1.90 SBI_EXT_SSE_COMPLETE . . . 396
30.70.1.91 SBI_EXT_SSE_DISABLE . . . 396

30.70.1.92 SBI_EXT_SSE_ENABLE . . . 396
30.70.1.93 SBI_EXT_SSE_HART_MASK . . . 396
30.70.1.94 SBI_EXT_SSE_HART_UNMASK . . . 396
30.70.1.95 SBI_EXT_SSE_INJECT . . . 396
30.70.1.96 SBI_EXT_SSE_READ_ATTR . . . 396
30.70.1.97 SBI_EXT_SSE_REGISTER . . . 396
30.70.1.98 SBI_EXT_SSE_UNREGISTER . . . 396
30.70.1.99 SBI_EXT_SSE_WRITE_ATTR . . . 396
30.70.1.100 SBI_EXT_SUSP . . . 397
30.70.1.101 SBI_EXT_SUSP_SUSPEND . . . 397
30.70.1.102 SBI_EXT_TIME . . . 397
30.70.1.103 SBI_EXT_TIME_SET_TIMER . . . 397
30.70.1.104 SBI_EXT_VENDOR_END . . . 397
30.70.1.105 SBI_EXT_VENDOR_START . . . 397
30.70.1.106 SBI_FWFT_GLOBAL_FEATURE_BIT . . . 397
30.70.1.107 SBI_FWFT_PLATFORM_FEATURE_BIT . . . 397
30.70.1.108 SBI_FWFT_SET_FLAG_LOCK . . . 397
30.70.1.109 SBI_HSM_STATE_RESUME_PENDING . . . 397
30.70.1.110 SBI_HSM_STATE_START_PENDING . . . 397
30.70.1.111 SBI_HSM_STATE_STARTED . . . 397
30.70.1.112 SBI_HSM_STATE_STOP_PENDING . . . 397
30.70.1.113 SBI_HSM_STATE_STOPPED . . . 397
30.70.1.114 SBI_HSM_STATE_SUSPEND_PENDING . . . 397
30.70.1.115 SBI_HSM_STATE_SUSPENDED . . . 398
30.70.1.116 SBI_HSM_SUSP_BASE_MASK . . . 398
30.70.1.117 SBI_HSM_SUSP_NON_RET_BIT . . . 398
30.70.1.118 SBI_HSM_SUSP_PLAT_BASE . . . 398
30.70.1.119 SBI_HSM_SUSPEND_NON_RET_DEFAULT . . . 398
30.70.1.120 SBI_HSM_SUSPEND_NON_RET_LAST . . . 398
30.70.1.121 SBI_HSM_SUSPEND_NON_RET_PLATFORM . . . 398
30.70.1.122 SBI_HSM_SUSPEND_RET_DEFAULT . . . 398
30.70.1.123 SBI_HSM_SUSPEND_RET_LAST . . . 398
30.70.1.124 SBI_HSM_SUSPEND_RET_PLATFORM . . . 398
30.70.1.125 SBI_LAST_ERR . . . 398
30.70.1.126 SBI_PMU_CFG_FLAG_AUTO_START . . . 398
30.70.1.127 SBI_PMU_CFG_FLAG_CLEAR_VALUE . . . 398
30.70.1.128 SBI_PMU_CFG_FLAG_SET_MINH . . . 398
30.70.1.129 SBI_PMU_CFG_FLAG_SET_SINH . . . 399
30.70.1.130 SBI_PMU_CFG_FLAG_SET_UINH . . . 399
30.70.1.131 SBI_PMU_CFG_FLAG_SET_VSINH . . . 399
30.70.1.132 SBI_PMU_CFG_FLAG_SET_VUINH . . . 399
30.70.1.133 SBI_PMU_CFG_FLAG_SKIP_MATCH . . . 399

30.70.1.134 SBI_PMU_EVENT_HW_CACHE_ID_MASK 399
30.70.1.135 SBI_PMU_EVENT_HW_CACHE_ID_OFFSET 399
30.70.1.136 SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK 399
30.70.1.137 SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET 399
30.70.1.138 SBI_PMU_EVENT_HW_CACHE_OPS_RESULT 399
30.70.1.139 SBI_PMU_EVENT_IDX_CODE_MASK 399
30.70.1.140 SBI_PMU_EVENT_IDX_INVALID 399
30.70.1.141 SBI_PMU_EVENT_IDX_MASK 399
30.70.1.142 SBI_PMU_EVENT_IDX_TYPE_MASK 399
30.70.1.143 SBI_PMU_EVENT_IDX_TYPE_OFFSET 399
30.70.1.144 SBI_PMU_EVENT_RAW_IDX 400
30.70.1.145 SBI_PMU_EVENT_RAW_V2_IDX 400
30.70.1.146 SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT 400
30.70.1.147 SBI_PMU_START_FLAG_SET_INIT_VALUE 400
30.70.1.148 SBI_PMU_STOP_FLAG_RESET 400
30.70.1.149 SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT 400
30.70.1.150 SBI_SPEC_VERSION_MAJOR_MASK 400
30.70.1.151 SBI_SPEC_VERSION_MAJOR_OFFSET 400
30.70.1.152 SBI_SPEC_VERSION_MINOR_MASK 400
30.70.1.153 SBI_SRST_RESET_REASON_NONE 400
30.70.1.154 SBI_SRST_RESET_REASON_SYSFAIL 400
30.70.1.155 SBI_SRST_RESET_TYPE_COLD_REBOOT 400
30.70.1.156 SBI_SRST_RESET_TYPE_LAST 400
30.70.1.157 SBI_SRST_RESET_TYPE_SHUTDOWN 400
30.70.1.158 SBI_SRST_RESET_TYPE_WARM_REBOOT 400
30.70.1.159 SBI_SSE_ATTR_CONFIG_ONESHOT 401
30.70.1.160 SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV 401
30.70.1.161 SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP 401
30.70.1.162 SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE 401
30.70.1.163 SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP 401
30.70.1.164 SBI_SSE_ATTR_STATUS_INJECT_OFFSET 401
30.70.1.165 SBI_SSE_ATTR_STATUS_PENDING_OFFSET 401
30.70.1.166 SBI_SSE_ATTR_STATUS_STATE_MASK 401
30.70.1.167 SBI_SSE_ATTR_STATUS_STATE_OFFSET 401
30.70.1.168 SBI_SSE_EVENT_GLOBAL_BIT 401
30.70.1.169 SBI_SSE_EVENT_GLOBAL_PLAT_0_END 401
30.70.1.170 SBI_SSE_EVENT_GLOBAL_PLAT_0_START 401
30.70.1.171 SBI_SSE_EVENT_GLOBAL_PLAT_1_END 401
30.70.1.172 SBI_SSE_EVENT_GLOBAL_PLAT_1_START 401
30.70.1.173 SBI_SSE_EVENT_GLOBAL_PLAT_2_END 401
30.70.1.174 SBI_SSE_EVENT_GLOBAL_PLAT_2_START 402
30.70.1.175 SBI_SSE_EVENT_GLOBAL_PLAT_3_END 402

30.70.1.176 SBI_SSE_EVENT_GLOBAL_PLAT_3_START . . . 402
30.70.1.177 SBI_SSE_EVENT_GLOBAL_RAS . . . 402
30.70.1.178 SBI_SSE_EVENT_GLOBAL_SOFTWARE . . . 402
30.70.1.179 SBI_SSE_EVENT_LOCAL_DOUBLE_TRAP . . . 402
30.70.1.180 SBI_SSE_EVENT_LOCAL_PLAT_0_END . . . 402
30.70.1.181 SBI_SSE_EVENT_LOCAL_PLAT_0_START . . . 402
30.70.1.182 SBI_SSE_EVENT_LOCAL_PLAT_1_END . . . 402
30.70.1.183 SBI_SSE_EVENT_LOCAL_PLAT_1_START . . . 402
30.70.1.184 SBI_SSE_EVENT_LOCAL_PLAT_2_END . . . 402
30.70.1.185 SBI_SSE_EVENT_LOCAL_PLAT_2_START . . . 402
30.70.1.186 SBI_SSE_EVENT_LOCAL_PLAT_3_END . . . 402
30.70.1.187 SBI_SSE_EVENT_LOCAL_PLAT_3_START . . . 402
30.70.1.188 SBI_SSE_EVENT_LOCAL_PMU . . . 402
30.70.1.189 SBI_SSE_EVENT_LOCAL_RAS . . . 403
30.70.1.190 SBI_SSE_EVENT_LOCAL_SOFTWARE . . . 403
30.70.1.191 SBI_SSE_EVENT_PLATFORM_BIT . . . 403
30.70.1.192 SBI_SUCCESS . . . 403
30.70.1.193 SBI_SUSP_PLATFORM_SLEEP_START . . . 403
30.70.1.194 SBI_SUSP_SLEEP_TYPE_LAST . . . 403
30.70.1.195 SBI_SUSP_SLEEP_TYPE_SUSPEND . . . 403
30.70.2 Enumeration Type Documentation . . . 403
30.70.2.1 sbi_cppc_reg_id . . . 403
30.70.2.2 sbi_fwft_feature_t . . . 404
30.70.2.3 sbi_pmu_ctr_type . . . 404
30.70.2.4 sbi_pmu_event_type_id . . . 404
30.70.2.5 sbi_pmu_fw_event_code_id . . . 405
30.70.2.6 sbi_pmu_hw_cache_id . . . 405
30.70.2.7 sbi_pmu_hw_cache_op_id . . . 406
30.70.2.8 sbi_pmu_hw_cache_op_result_id . . . 406
30.70.2.9 sbi_pmu_hw_generic_events_t . . . 406
30.70.2.10 sbi_sse_attr_id . . . 406
30.70.2.11 sbi_sse_state . . . 407
30.71 sbi_ecall_interface.h . . . 407
30.72 include/sbi/sbi_emulate_csr.h File Reference . . . 412
30.72.1 Function Documentation . . . 413
30.72.1.1 sbi_emulate_csr_read() . . . 413
30.72.1.2 sbi_emulate_csr_write() . . . 414
30.73 sbi_emulate_csr.h . . . 414
30.74 include/sbi/sbi_error.h File Reference . . . 415
30.74.1 Macro Definition Documentation . . . 416
30.74.1.1 SBI_EALREADY . . . 416
30.74.1.2 SBI_EALREADY_STARTED . . . 416

30.74.1.3 SBI_EALREADY_STOPPED . . . 416
30.74.1.4 SBI_EBAD_RANGE . . . 416
30.74.1.5 SBI_EDENIED . . . 416
30.74.1.6 SBI_EFAIL . . . 416
30.74.1.7 SBI_EILL . . . 416
30.74.1.8 SBI_EINVAL . . . 416
30.74.1.9 SBI_EINVALID_ADDR . . . 416
30.74.1.10 SBI_EINVALID_STATE . . . 416
30.74.1.11 SBI_EIO . . . 416
30.74.1.12 SBI_ENO_SHMEM . . . 416
30.74.1.13 SBI_ENODEV . . . 416
30.74.1.14 SBI_ENOENT . . . 416
30.74.1.15 SBI_ENOMEM . . . 417
30.74.1.16 SBI_ENOSPC . . . 417
30.74.1.17 SBI_ENOSYS . . . 417
30.74.1.18 SBI_ENOTSUPP . . . 417
30.74.1.19 SBI_ETIMEDOUT . . . 417
30.74.1.20 SBI_ETIMEOUT . . . 417
30.74.1.21 SBI_EUNKNOWN . . . 417
30.74.1.22 SBI_OK . . . 417
30.75 sbi_error.h . . . 417
30.76 include/sbi/sbi_fifo.h File Reference . . . 418
30.76.1 Macro Definition Documentation . . . 419
30.76.1.1 SBI_FIFO_DEFINE . . . 419
30.76.1.2 SBI_FIFO_INITIALIZER . . . 419
30.76.2 Enumeration Type Documentation . . . 419
30.76.2.1 sbi_fifo_inplace_update_types . . . 419
30.76.3 Function Documentation . . . 419
30.76.3.1 sbi_fifo_avail() . . . 419
30.76.3.2 sbi_fifo_dequeue() . . . 420
30.76.3.3 sbi_fifo_enqueue() . . . 421
30.76.3.4 sbi_fifo_init() . . . 422
30.76.3.5 sbi_fifo_inplace_update() . . . 422
30.76.3.6 sbi_fifo_is_empty() . . . 423
30.76.3.7 sbi_fifo_is_full() . . . 423
30.77 sbi_fifo.h . . . 424
30.78 include/sbi/sbi_fwft.h File Reference . . . 425
30.78.1 Function Documentation . . . 425
30.78.1.1 sbi_fwft_get() . . . 425
30.78.1.2 sbi_fwft_init() . . . 426
30.78.1.3 sbi_fwft_set() . . . 426
30.79 sbi_fwft.h . . . 427

30.80 include/sbi/sbi_hart.h File Reference . . . 427
30.80.1 Macro Definition Documentation . . . 429
30.80.1.1 SBI_SMEPMP_RESV_ENTRY . . . 429
30.80.2 Enumeration Type Documentation . . . 429
30.80.2.1 sbi_hart_extensions . . . 429
30.80.2.2 sbi_hart_priv_versions . . . 430
30.80.3 Function Documentation . . . 430
30.80.3.1 __attribute__() . . . 430
30.80.3.2 sbi_hart_delegation_dump() . . . 430
30.80.3.3 sbi_hart_expected_trap_addr() . . . 431
30.80.3.4 sbi_hart_get_extensions_str() . . . 431
30.80.3.5 sbi_hart_get_priv_version_str() . . . 432
30.80.3.6 sbi_hart_has_extension() . . . 432
30.80.3.7 sbi_hart_init() . . . 433
30.80.3.8 sbi_hart_map_saddr() . . . 434
30.80.3.9 sbi_hart_mhpm_bits() . . . 435
30.80.3.10 sbi_hart_mhpm_mask() . . . 435
30.80.3.11 sbi_hart_pmp_addrbits() . . . 436
30.80.3.12 sbi_hart_pmp_configure() . . . 436
30.80.3.13 sbi_hart_pmp_count() . . . 436
30.80.3.14 sbi_hart_pmp_log2gran() . . . 437
30.80.3.15 sbi_hart_priv_version() . . . 437
30.80.3.16 sbi_hart_reinit() . . . 438
30.80.3.17 sbi_hart_unmap_saddr() . . . 438
30.80.3.18 sbi_hart_update_extension() . . . 439
30.80.4 Variable Documentation . . . 439
30.80.4.1 arg1 . . . 439
30.80.4.2 next_addr . . . 440
30.80.4.3 next_mode . . . 440
30.80.4.4 next_virt . . . 440
30.80.4.5 sbi_hart_expected_trap . . . 440
30.80.4.6 sbi_hart_ext . . . 440
30.81 sbi_hart.h . . . 440
30.82 include/sbi/sbi_hartmask.h File Reference . . . 441
30.82.1 Macro Definition Documentation . . . 443
30.82.1.1 sbi_hartmask_bits . . . 443
30.82.1.2 sbi_hartmask_for_each_hartindex . . . 443
30.82.1.3 SBI_HARTMASK_INIT . . . 443
30.82.1.4 SBI_HARTMASK_INIT_EXCEPT . . . 443
30.82.1.5 SBI_HARTMASK_MAX_BITS . . . 443
30.82.2 Function Documentation . . . 444
30.82.2.1 sbi_hartmask_and() . . . 444

30.82.2.2 sbi_hartmask_clear_all() . . . 444
30.82.2.3 sbi_hartmask_clear_hartid() . . . 445
30.82.2.4 sbi_hartmask_clear_hartindex() . . . 445
30.82.2.5 sbi_hartmask_copy() . . . 446
30.82.2.6 sbi_hartmask_or() . . . 447
30.82.2.7 sbi_hartmask_set_all() . . . 447
30.82.2.8 sbi_hartmask_set_hartid() . . . 448
30.82.2.9 sbi_hartmask_set_hartindex() . . . 448
30.82.2.10 sbi_hartmask_test_hartid() . . . 449
30.82.2.11 sbi_hartmask_test_hartindex() . . . 449
30.82.2.12 sbi_hartmask_xor() . . . 450
30.83 sbi_hartmask.h . . . 451
30.84 include/sbi/sbi_heap.h File Reference . . . 452
30.84.1 Macro Definition Documentation . . . 453
30.84.1.1 HEAP_BASE_ALIGN . . . 453
30.84.2 Function Documentation . . . 453
30.84.2.1 sbi_aligned_alloc() . . . 453
30.84.2.2 sbi_aligned_alloc_from() . . . 454
30.84.2.3 sbi_calloc() . . . 454
30.84.2.4 sbi_calloc_from() . . . 455
30.84.2.5 sbi_free() . . . 455
30.84.2.6 sbi_free_from() . . . 456
30.84.2.7 sbi_heap_alloc_new() . . . 457
30.84.2.8 sbi_heap_free_space() . . . 457
30.84.2.9 sbi_heap_free_space_from() . . . 458
30.84.2.10 sbi_heap_init() . . . 458
30.84.2.11 sbi_heap_init_new() . . . 459
30.84.2.12 sbi_heap_reserved_space() . . . 459
30.84.2.13 sbi_heap_reserved_space_from() . . . 460
30.84.2.14 sbi_heap_used_space() . . . 460
30.84.2.15 sbi_heap_used_space_from() . . . 460
30.84.2.16 sbi_malloc() . . . 461
30.84.2.17 sbi_malloc_from() . . . 461
30.84.2.18 sbi_zalloc() . . . 462
30.84.2.19 sbi_zalloc_from() . . . 463
30.84.3 Variable Documentation . . . 464
30.84.3.1 global_hpctrl . . . 464
30.85 sbi_heap.h . . . 464
30.86 include/sbi/sbi_hfence.h File Reference . . . 466
30.86.1 Function Documentation . . . 466
30.86.1.1 __sbi_hfence_gvma_all() . . . 466
30.86.1.2 __sbi_hfence_gvma_gpa() . . . 466

30.86.1.3 __sbi_hfence_gvma_vmid() . . . 467
30.86.1.4 __sbi_hfence_gvma_vmid_gpa() . . . 467
30.86.1.5 __sbi_hfence_vvma_all() . . . 467
30.86.1.6 __sbi_hfence_vvma_asid() . . . 467
30.86.1.7 __sbi_hfence_vvma_asid_va() . . . 467
30.86.1.8 __sbi_hfence_vvma_va() . . . 468
30.87 sbi_hfence.h . . . 468
30.88 include/sbi/sbi_hsm.h File Reference . . . 468
30.88.1 Function Documentation . . . 470
30.88.1.1 __sbi_hsm_hart_get_state() . . . 470
30.88.1.2 __sbi_hsm_suspend_non_ret_save() . . . 470
30.88.1.3 sbi_hsm_exit() . . . 471
30.88.1.4 sbi_hsm_get_device() . . . 471
30.88.1.5 sbi_hsm_hart_change_state() . . . 472
30.88.1.6 sbi_hsm_hart_get_state() . . . 472
30.88.1.7 sbi_hsm_hart_interruptible_mask() . . . 472
30.88.1.8 sbi_hsm_hart_resume_finish() . . . 473
30.88.1.9 sbi_hsm_hart_resume_start() . . . 474
30.88.1.10 sbi_hsm_hart_start() . . . 474
30.88.1.11 sbi_hsm_hart_start_finish() . . . 475
30.88.1.12 sbi_hsm_hart_stop() . . . 476
30.88.1.13 sbi_hsm_hart_suspend() . . . 476
30.88.1.14 sbi_hsm_init() . . . 477
30.88.1.15 sbi_hsm_set_device() . . . 478
30.89 sbi_hsm.h . . . 478
30.90 include/sbi/sbi_illegal_insn.h File Reference . . . 479
30.90.1 Function Documentation . . . 480
30.90.1.1 sbi_illegal_insn_handler() . . . 480
30.91 sbi_illegal_insn.h . . . 480
30.92 include/sbi/sbi_init.h File Reference . . . 481
30.92.1 Function Documentation . . . 481
30.92.1.1 sbi_entry_count() . . . 481
30.92.1.2 sbi_exit() . . . 482
30.92.1.3 sbi_init() . . . 482
30.92.1.4 sbi_init_count() . . . 484
30.93 sbi_init.h . . . 485
30.94 include/sbi/sbi_ipi.h File Reference . . . 485
30.94.1 Macro Definition Documentation . . . 486
30.94.1.1 SBI_IPI_EVENT_MAX . . . 486
30.94.2 Enumeration Type Documentation . . . 486
30.94.2.1 sbi_ipi_update_type . . . 486
30.94.3 Function Documentation . . . 486

30.94.3.1 sbi_ipi_clear_smode() . . . 486
30.94.3.2 sbi_ipi_event_create() . . . 487
30.94.3.3 sbi_ipi_event_destroy() . . . 487
30.94.3.4 sbi_ipi_exit() . . . 487
30.94.3.5 sbi_ipi_get_device() . . . 487
30.94.3.6 sbi_ipi_init() . . . 488
30.94.3.7 sbi_ipi_process() . . . 488
30.94.3.8 sbi_ipi_raw_clear() . . . 489
30.94.3.9 sbi_ipi_raw_send() . . . 489
30.94.3.10 sbi_ipi_send_halt() . . . 490
30.94.3.11 sbi_ipi_send_many() . . . 490
30.94.3.12 sbi_ipi_send_smode() . . . 491
30.94.3.13 sbi_ipi_set_device() . . . 492
30.95 sbi_ipi.h . . . 492
30.96 include/sbi/sbi_irqchip.h File Reference . . . 493
30.96.1 Function Documentation . . . 494
30.96.1.1 sbi_irqchip_add_device() . . . 494
30.96.1.2 sbi_irqchip_exit() . . . 494
30.96.1.3 sbi_irqchip_init() . . . 495
30.96.1.4 sbi_irqchip_process() . . . 496
30.97 sbi_irqchip.h . . . 496
30.98 include/sbi/sbi_list.h File Reference . . . 497
30.98.1 Macro Definition Documentation . . . 498
30.98.1.1 SBI_INIT_LIST_HEAD . . . 498
30.98.1.2 sbi_list_entry . . . 498
30.98.1.3 sbi_list_first_entry . . . 498
30.98.1.4 sbi_list_for_each . . . 498
30.98.1.5 sbi_list_for_each_entry . . . 498
30.98.1.6 SBI_LIST_HEAD . . . 499
30.98.1.7 SBI_LIST_HEAD_INIT . . . 499
30.98.1.8 sbi_list_last_entry . . . 499
30.98.1.9 SBI_LIST_POISON_NEXT . . . 499
30.98.1.10 SBI_LIST_POISON_PREV . . . 499
30.98.2 Function Documentation . . . 499
30.98.2.1 __sbi_list_add() . . . 499
30.98.2.2 __sbi_list_del() . . . 500
30.98.2.3 __sbi_list_del_entry() . . . 501
30.98.2.4 sbi_list_add() . . . 501
30.98.2.5 sbi_list_add_tail() . . . 504
30.98.2.6 sbi_list_del() . . . 505
30.98.2.7 sbi_list_del_init() . . . 506
30.98.2.8 sbi_list_empty() . . . 507

30.99 sbi_list.h . . . 507
30.100 include/sbi/sbi_math.h File Reference . . . 509
30.100.1 Function Documentation . . . 509
30.100.1.1 log2roundup() . . . 509
30.101 sbi_math.h . . . 510
30.102 include/sbi/sbi_mpxy.h File Reference . . . 511
30.102.1 Macro Definition Documentation . . . 512
30.102.1.1 SBI_MPXY_MSGPROTO_VERSION . . . 512
30.102.2 Enumeration Type Documentation . . . 512
30.102.2.1 SBI_EXT_MPXY_SHMEM_FLAGS . . . 512
30.102.2.2 sbi_mpxy_attr_id . . . 512
30.102.2.3 sbi_mpxy_msgproto_id . . . 513
30.102.3 Function Documentation . . . 513
30.102.3.1 sbi_mpxy_channel_available() . . . 513
30.102.3.2 sbi_mpxy_get_channel_ids() . . . 514
30.102.3.3 sbi_mpxy_get_notification_events() . . . 514
30.102.3.4 sbi_mpxy_init() . . . 514
30.102.3.5 sbi_mpxy_read_attrs() . . . 515
30.102.3.6 sbi_mpxy_register_channel() . . . 516
30.102.3.7 sbi_mpxy_send_message() . . . 516
30.102.3.8 sbi_mpxy_set_shmem() . . . 517
30.102.3.9 sbi_mpxy_write_attrs() . . . 517
30.103 sbi_mpxy.h . . . 518
30.104 include/sbi/sbi_platform.h File Reference . . . 520
30.104.1 Macro Definition Documentation . . . 522
30.104.1.1 SBI_PLATFORM_DEFAULT_FEATURES . . . 522
30.104.1.2 SBI_PLATFORM_DEFAULT_HART_STACK_SIZE . . . 522
30.104.1.3 SBI_PLATFORM_DEFAULT_HEAP_SIZE . . . 522
30.104.1.4 SBI_PLATFORM_FEATURES_OFFSET . . . 522
30.104.1.5 SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET . . . 522
30.104.1.6 SBI_PLATFORM_HART_COUNT_OFFSET . . . 522
30.104.1.7 SBI_PLATFORM_HART_INDEX2ID_OFFSET . . . 522
30.104.1.8 SBI_PLATFORM_HART_STACK_SIZE_OFFSET . . . 522
30.104.1.9 sbi_platform_has_mfaults_delegation . . . 522
30.104.1.10 SBI_PLATFORM_HEAP_SIZE_OFFSET . . . 522
30.104.1.11 SBI_PLATFORM_NAME_OFFSET . . . 523
30.104.1.12 SBI_PLATFORM_OPENSBI_VERSION_OFFSET . . . 523
30.104.1.13 sbi_platform_ops . . . 523
30.104.1.14 SBI_PLATFORM_OPS_OFFSET . . . 523
30.104.1.15 sbi_platform_ptr . . . 523
30.104.1.16 SBI_PLATFORM_RESERVED_OFFSET . . . 523
30.104.1.17 sbi_platform_thishart_ptr . . . 523

30.104.1.18 SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT . . . 523
30.104.1.19 SBI_PLATFORM_VERSION . . . 523
30.104.1.20 SBI_PLATFORM_VERSION_OFFSET . . . 523
30.104.2 Enumeration Type Documentation . . . 524
30.104.2.1 sbi_platform_features . . . 524
30.104.3 Function Documentation . . . 524
30.104.3.1 sbi_platform_cold_boot_allowed() . . . 524
30.104.3.2 sbi_platform_domains_init() . . . 524
30.104.3.3 sbi_platform_early_exit() . . . 525
30.104.3.4 sbi_platform_early_init() . . . 525
30.104.3.5 sbi_platform_emulate_load() . . . 526
30.104.3.6 sbi_platform_emulate_store() . . . 526
30.104.3.7 sbi_platform_extensions_init() . . . 527
30.104.3.8 sbi_platform_final_exit() . . . 527
30.104.3.9 sbi_platform_final_init() . . . 528
30.104.3.10 sbi_platform_get_features() . . . 528
30.104.3.11 sbi_platform_get_features_str() . . . 529
30.104.3.12 sbi_platform_hart_count() . . . 529
30.104.3.13 sbi_platform_hart_stack_size() . . . 530
30.104.3.14 sbi_platform_ipi_init() . . . 530
30.104.3.15 sbi_platform_irqchip_init() . . . 530
30.104.3.16 sbi_platform_misa_extension() . . . 532
30.104.3.17 sbi_platform_misa_xlen() . . . 532
30.104.3.18 sbi_platform_mpxy_init() . . . 533
30.104.3.19 sbi_platform_name() . . . 533
30.104.3.20 sbi_platform_nascent_init() . . . 534
30.104.3.21 sbi_platform_pmu_init() . . . 534
30.104.3.22 sbi_platform_pmu_xlate_to_mhpmevent() . . . 534
30.104.3.23 sbi_platform_timer_init() . . . 536
30.104.3.24 sbi_platform_tlb_fifo_num_entries() . . . 536
30.104.3.25 sbi_platform_tlbr_flush_limit() . . . 537
30.104.3.26 sbi_platform_vendor_ext_check() . . . 537
30.104.3.27 sbi_platform_vendor_ext_provider() . . . 538
30.105 sbi_platform.h . . . 538
30.106 include/sbi/sbi_pmu.h File Reference . . . 543
30.106.1 Macro Definition Documentation . . . 545
30.106.1.1 SBI_PMU_CTR_MAX . . . 545
30.106.1.2 SBI_PMU_CY_IR_MASK . . . 545
30.106.1.3 SBI_PMU_FIXED_CTR_MASK . . . 545
30.106.1.4 SBI_PMU_FW_CTR_MAX . . . 545
30.106.1.5 SBI_PMU_HW_CTR_MAX . . . 545
30.106.1.6 SBI_PMU_HW_EVENT_MAX . . . 545

30.106.2 Function Documentation . . . 545
30.106.2.1 sbi_pmu_add_hw_event_counter_map() . . . 545
30.106.2.2 sbi_pmu_add_raw_event_counter_map() . . . 546
30.106.2.3 sbi_pmu_ctr_cfg_match() . . . 547
30.106.2.4 sbi_pmu_ctr_fw_read() . . . 548
30.106.2.5 sbi_pmu_ctr_get_info() . . . 548
30.106.2.6 sbi_pmu_ctr_incr_fw() . . . 549
30.106.2.7 sbi_pmu_ctr_start() . . . 549
30.106.2.8 sbi_pmu_ctr_stop() . . . 550
30.106.2.9 sbi_pmu_event_get_info() . . . 550
30.106.2.10 sbi_pmu_exit() . . . 551
30.106.2.11 sbi_pmu_get_device() . . . 551
30.106.2.12 sbi_pmu_init() . . . 552
30.106.2.13 sbi_pmu_irq_bit() . . . 552
30.106.2.14 sbi_pmu_num_ctr() . . . 552
30.106.2.15 sbi_pmu_ovf_irq() . . . 553
30.106.2.16 sbi_pmu_set_device() . . . 553
30.107 sbi_pmu.h . . . 553
30.108 include/sbi/sbi_scratch.h File Reference . . . 555
30.108.1 Macro Definition Documentation . . . 556
30.108.1.1 current_hartindex . . . 556
30.108.1.2 sbi_hartid_to_scratch . . . 556
30.108.1.3 sbi_hartid_valid . . . 556
30.108.1.4 sbi_hartindex_to_hartid . . . 557
30.108.1.5 sbi_hartindex_to_scratch . . . 557
30.108.1.6 sbi_hartindex_valid . . . 557
30.108.1.7 sbi_scratch_alloc_type_offset . . . 557
30.108.1.8 SBI_SCRATCH_EXTRA_SPACE_OFFSET . . . 557
30.108.1.9 SBI_SCRATCH_FW_HEAP_OFFSET . . . 557
30.108.1.10 SBI_SCRATCH_FW_HEAP_SIZE_OFFSET . . . 557
30.108.1.11 SBI_SCRATCH_FW_RW_OFFSET . . . 557
30.108.1.12 SBI_SCRATCH_FW_SIZE_OFFSET . . . 557
30.108.1.13 SBI_SCRATCH_FW_START_OFFSET . . . 557
30.108.1.14 SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET . . . 558
30.108.1.15 SBI_SCRATCH_HARTINDEX_OFFSET . . . 558
30.108.1.16 sbi_scratch_last_hartindex . . . 558
30.108.1.17 SBI_SCRATCH_NEXT_ADDR_OFFSET . . . 558
30.108.1.18 SBI_SCRATCH_NEXT_ARG1_OFFSET . . . 558
30.108.1.19 SBI_SCRATCH_NEXT_MODE_OFFSET . . . 558
30.108.1.20 sbi_scratch_offset_ptr . . . 558
30.108.1.21 SBI_SCRATCH_OPTIONS_OFFSET . . . 558
30.108.1.22 SBI_SCRATCH_PLATFORM_ADDR_OFFSET . . . 558

30.108.1.23 sbi_scratch_read_type . . . 558
30.108.1.24 SBI_SCRATCH_SIZE . . . 559
30.108.1.25 sbi_scratch_thishart_arg1_ptr . . . 559
30.108.1.26 sbi_scratch_thishart_offset_ptr . . . 559
30.108.1.27 sbi_scratch_thishart_ptr . . . 559
30.108.1.28 SBI_SCRATCH_TMP0_OFFSET . . . 559
30.108.1.29 SBI_SCRATCH_TRAP_CONTEXT_OFFSET . . . 559
30.108.1.30 SBI_SCRATCH_WARMBOOT_ADDR_OFFSET . . . 559
30.108.1.31 sbi_scratch_write_type . . . 559
30.108.2 Enumeration Type Documentation . . . 559
30.108.2.1 sbi_scratch_options . . . 559
30.108.3 Function Documentation . . . 560
30.108.3.1 sbi_hartid_to_hartindex() . . . 560
30.108.3.2 sbi_scratch_alloc_offset() . . . 560
30.108.3.3 sbi_scratch_free_offset() . . . 561
30.108.3.4 sbi_scratch_init() . . . 561
30.108.3.5 sbi_scratch_used_space() . . . 561
30.108.4 Variable Documentation . . . 562
30.108.4.1 hartindex_to_hartid_table . . . 562
30.108.4.2 hartindex_to_scratch_table . . . 562
30.108.4.3 last_hartindex_having_scratch . . . 562
30.109 sbi_scratch.h . . . 562
30.110 include/sbi/sbi_sse.h File Reference . . . 565
30.110.1 Macro Definition Documentation . . . 566
30.110.1.1 EXC_MODE_PP . . . 566
30.110.1.2 EXC_MODE_PP_SHIFT . . . 566
30.110.1.3 EXC_MODE_PV . . . 566
30.110.1.4 EXC_MODE_PV_SHIFT . . . 566
30.110.1.5 EXC_MODE_SSTATUS_SPIE . . . 566
30.110.1.6 EXC_MODE_SSTATUS_SPIE_SHIFT . . . 566
30.110.2 Function Documentation . . . 566
30.110.2.1 sbi_sse_complete() . . . 566
30.110.2.2 sbi_sse_disable() . . . 567
30.110.2.3 sbi_sse_enable() . . . 567
30.110.2.4 sbi_sse_exit() . . . 568
30.110.2.5 sbi_sse_hart_mask() . . . 569
30.110.2.6 sbi_sse_hart_unmask() . . . 569
30.110.2.7 sbi_sse_init() . . . 569
30.110.2.8 sbi_sse_inject_event() . . . 570
30.110.2.9 sbi_sse_inject_from_ecall() . . . 571
30.110.2.10 sbi_sse_process_pending_events() . . . 571
30.110.2.11 sbi_sse_read_attrs() . . . 572

30.110.2.12 sbi_sse_register() . . . 572
30.110.2.13 sbi_sse_set_cb_ops() . . . 573
30.110.2.14 sbi_sse_unregister() . . . 574
30.110.2.15 sbi_sse_write_attrs() . . . 574
30.111 sbi_sse.h . . . 575
30.112 include/sbi/sbi_string.h File Reference . . . 576
30.112.1 Function Documentation . . . 577
30.112.1.1 sbi_memchr() . . . 577
30.112.1.2 sbi_memcmp() . . . 577
30.112.1.3 sbi_memcpy() . . . 577
30.112.1.4 sbi_memmove() . . . 578
30.112.1.5 sbi_memset() . . . 578
30.112.1.6 sbi_strchr() . . . 579
30.112.1.7 sbi_strcmp() . . . 579
30.112.1.8 sbi_strcpy() . . . 579
30.112.1.9 sbi_strlen() . . . 579
30.112.1.10 sbi_strncmp() . . . 580
30.112.1.11 sbi_strncpy() . . . 580
30.112.1.12 sbi_strnlen() . . . 581
30.112.1.13 sbi_strrchr() . . . 581
30.113 sbi_string.h . . . 581
30.114 include/sbi/sbi_system.h File Reference . . . 582
30.114.1 Function Documentation . . . 583
30.114.1.1 sbi_system_reset() . . . 583
30.114.1.2 sbi_system_reset_add_device() . . . 583
30.114.1.3 sbi_system_reset_get_device() . . . 584
30.114.1.4 sbi_system_reset_supported() . . . 585
30.114.1.5 sbi_system_suspend() . . . 585
30.114.1.6 sbi_system_suspend_get_device() . . . 586
30.114.1.7 sbi_system_suspend_set_device() . . . 586
30.114.1.8 sbi_system_suspend_supported() . . . 587
30.114.1.9 sbi_system_suspend_test_enable() . . . 587
30.114.1.10 to_system_reset_device() . . . 587
30.115 sbi_system.h . . . 588
30.116 include/sbi/sbi_timer.h File Reference . . . 589
30.116.1 Function Documentation . . . 589
30.116.1.1 sbi_timer_delay_loop() . . . 589
30.116.1.2 sbi_timer_event_start() . . . 590
30.116.1.3 sbi_timer_exit() . . . 591
30.116.1.4 sbi_timer_get_delta() . . . 591
30.116.1.5 sbi_timer_get_device() . . . 591
30.116.1.6 sbi_timer_init() . . . 591

30.116.1.7 sbi_timer_mdelay() . . . 592
30.116.1.8 sbi_timer_process() . . . 593
30.116.1.9 sbi_timer_set_delta() . . . 593
30.116.1.10 sbi_timer_set_device() . . . 593
30.116.1.11 sbi_timer_udelay() . . . 594
30.116.1.12 sbi_timer_value() . . . 594
30.116.1.13 sbi_timer_virt_value() . . . 594
30.116.1.14 sbi_timer_waitms_until() . . . 595
30.117 sbi_timer.h . . . 596
30.118 include/sbi/sbi_tlb.h File Reference . . . 596
30.118.1 Macro Definition Documentation . . . 598
30.118.1.1 SBI_TLB_FLUSH_ALL . . . 598
30.118.1.2 SBI_TLB_INFO_INIT . . . 598
30.118.1.3 SBI_TLB_INFO_SIZE . . . 598
30.118.2 Enumeration Type Documentation . . . 598
30.118.2.1 sbi_tlb_type . . . 598
30.118.3 Function Documentation . . . 598
30.118.3.1 sbi_tlb_init() . . . 598
30.118.3.2 sbi_tlb_request() . . . 599
30.119 sbi_tlb.h . . . 600
30.120 include/sbi/sbi_trap.h File Reference . . . 601
30.120.1 Macro Definition Documentation . . . 603
30.120.1.1 SBI_TRAP_CONTEXT_SIZE . . . 603
30.120.1.2 SBI_TRAP_INFO_cause . . . 603
30.120.1.3 SBI_TRAP_INFO_gva . . . 603
30.120.1.4 SBI_TRAP_INFO_last . . . 603
30.120.1.5 SBI_TRAP_INFO_OFFSET . . . 603
30.120.1.6 SBI_TRAP_INFO_SIZE . . . 603
30.120.1.7 SBI_TRAP_INFO_tinst . . . 603
30.120.1.8 SBI_TRAP_INFO_tval . . . 603
30.120.1.9 SBI_TRAP_INFO_tval2 . . . 603
30.120.1.10 SBI_TRAP_REGS_a0 . . . 603
30.120.1.11 SBI_TRAP_REGS_a1 . . . 603
30.120.1.12 SBI_TRAP_REGS_a2 . . . 604
30.120.1.13 SBI_TRAP_REGS_a3 . . . 604
30.120.1.14 SBI_TRAP_REGS_a4 . . . 604
30.120.1.15 SBI_TRAP_REGS_a5 . . . 604
30.120.1.16 SBI_TRAP_REGS_a6 . . . 604
30.120.1.17 SBI_TRAP_REGS_a7 . . . 604
30.120.1.18 SBI_TRAP_REGS_gp . . . 604
30.120.1.19 SBI_TRAP_REGS_last . . . 604
30.120.1.20 SBI_TRAP_REGS_mepc . . . 604

30.120.1.21 SBI_TRAP_REGS_mstatus . . . 604
30.120.1.22 SBI_TRAP_REGS_mstatusH . . . 604
30.120.1.23 SBI_TRAP_REGS_OFFSET . . . 604
30.120.1.24 SBI_TRAP_REGS_ra . . . 605
30.120.1.25 SBI_TRAP_REGS_s0 . . . 605
30.120.1.26 SBI_TRAP_REGS_s1 . . . 605
30.120.1.27 SBI_TRAP_REGS_s10 . . . 605
30.120.1.28 SBI_TRAP_REGS_s11 . . . 605
30.120.1.29 SBI_TRAP_REGS_s2 . . . 605
30.120.1.30 SBI_TRAP_REGS_s3 . . . 605
30.120.1.31 SBI_TRAP_REGS_s4 . . . 605
30.120.1.32 SBI_TRAP_REGS_s5 . . . 605
30.120.1.33 SBI_TRAP_REGS_s6 . . . 605
30.120.1.34 SBI_TRAP_REGS_s7 . . . 605
30.120.1.35 SBI_TRAP_REGS_s8 . . . 605
30.120.1.36 SBI_TRAP_REGS_s9 . . . 606
30.120.1.37 SBI_TRAP_REGS_SIZE . . . 606
30.120.1.38 SBI_TRAP_REGS_sp . . . 606
30.120.1.39 SBI_TRAP_REGS_t0 . . . 606
30.120.1.40 SBI_TRAP_REGS_t1 . . . 606
30.120.1.41 SBI_TRAP_REGS_t2 . . . 606
30.120.1.42 SBI_TRAP_REGS_t3 . . . 606
30.120.1.43 SBI_TRAP_REGS_t4 . . . 606
30.120.1.44 SBI_TRAP_REGS_t5 . . . 606
30.120.1.45 SBI_TRAP_REGS_t6 . . . 606
30.120.1.46 SBI_TRAP_REGS_tp . . . 606
30.120.1.47 SBI_TRAP_REGS_zero . . . 606
30.120.2 Function Documentation . . . 607
30.120.2.1 sbi_mstatus_prev_mode() . . . 607
30.120.2.2 sbi_regs_from_virt() . . . 607
30.120.2.3 sbi_regs_gva() . . . 608
30.120.2.4 sbi_trap_get_context() . . . 608
30.120.2.5 sbi_trap_handler() . . . 608
30.120.2.6 sbi_trap_redirect() . . . 609
30.120.2.7 sbi_trap_set_context() . . . 610
30.121 sbi_trap.h . . . 610
30.122 include/sbi/sbi_trap_ldst.h File Reference . . . 612
30.122.1 Function Documentation . . . 614
30.122.1.1 sbi_double_trap_handler() . . . 614
30.122.1.2 sbi_load_access_handler() . . . 614
30.122.1.3 sbi_misaligned_load_handler() . . . 615
30.122.1.4 sbi_misaligned_store_handler() . . . 615

30.122.1.5 sbi_misaligned_tinst_fixup() . . . 616
30.122.1.6 sbi_misaligned_v_ld_emulator() . . . 616
30.122.1.7 sbi_misaligned_v_st_emulator() . . . 616
30.122.1.8 sbi_store_access_handler() . . . 617
30.123 sbi_trap_ldst.h . . . 617
30.124 include/sbi/sbi_types.h File Reference . . . 618
30.124.1 Macro Definition Documentation . . . 619
30.124.1.1 __aligned . . . 619
30.124.1.2 __always_inline . . . 619
30.124.1.3 __has_builtin . . . 619
30.124.1.4 __noreturn . . . 619
30.124.1.5 __packed . . . 619
30.124.1.6 array_size . . . 619
30.124.1.7 CLAMP . . . 619
30.124.1.8 container_of . . . 619
30.124.1.9 false . . . 620
30.124.1.10 likely . . . 620
30.124.1.11 MAX . . . 620
30.124.1.12 MIN . . . 620
30.124.1.13 NULL . . . 620
30.124.1.14 offsetof . . . 620
30.124.1.15 ROUNDDOWN . . . 620
30.124.1.16 ROUNDUP . . . 620
30.124.1.17 STR . . . 620
30.124.1.18 true . . . 620
30.124.1.19 unlikely . . . 620
30.124.1.20 XSTR . . . 620
30.124.2 Typedef Documentation . . . 621
30.124.2.1 be16_t . . . 621
30.124.2.2 be32_t . . . 621
30.124.2.3 be64_t . . . 621
30.124.2.4 bool . . . 621
30.124.2.5 int16_t . . . 621
30.124.2.6 int32_t . . . 621
30.124.2.7 le16_t . . . 621
30.124.2.8 le32_t . . . 621
30.124.2.9 le64_t . . . 621
30.124.2.10 physical_addr_t . . . 621
30.124.2.11 physical_size_t . . . 621
30.124.2.12 s16 . . . 621
30.124.2.13 s32 . . . 621
30.124.2.14 s8 . . . 621

30.124.2.15 size_t . . . 621
30.124.2.16 ssize_t . . . 622
30.124.2.17 u16 . . . 622
30.124.2.18 u32 . . . 622
30.124.2.19 u8 . . . 622
30.124.2.20 uint16_t . . . 622
30.124.2.21 uint32_t . . . 622
30.124.2.22 uint8_t . . . 622
30.124.2.23 uintptr_t . . . 622
30.124.2.24 ulong . . . 622
30.124.2.25 virtual_addr_t . . . 622
30.124.2.26 virtual_size_t . . . 622
30.125 sbi_types.h . . . 622
30.126 include/sbi/sbi_unit_test.h File Reference . . . 624
30.126.1 Macro Definition Documentation . . . 624
30.126.1.1 run_all_tests . . . 624
30.127 sbi_unit_test.h . . . 624
30.128 include/sbi/sbi_unpriv.h File Reference . . . 625
30.128.1 Macro Definition Documentation . . . 626
30.128.1.1 DECLARE_UNPRIVILEGED_LOAD_FUNCTION . . . 626
30.128.1.2 DECLARE_UNPRIVILEGED_STORE_FUNCTION . . . 626
30.128.2 Function Documentation . . . 626
30.128.2.1 sbi_get_insn() . . . 626
30.129 sbi_unpriv.h . . . 627
30.130 include/sbi/sbi_version.h File Reference . . . 627
30.130.1 Macro Definition Documentation . . . 628
30.130.1.1 OPENSBI_VERSION . . . 628
30.130.1.2 OPENSBI_VERSION_MAJOR . . . 628
30.130.1.3 OPENSBI_VERSION_MINOR . . . 628
30.131 sbi_version.h . . . 628
30.132 include/sbi_utils/cppc/fdt_cppc.h File Reference . . . 628
30.132.1 Function Documentation . . . 629
30.132.1.1 fdt_cppc_init() . . . 629
30.133 fdt_cppc.h . . . 629
30.134 include/sbi_utils/fdt/fdt_domain.h File Reference . . . 630
30.134.1 Function Documentation . . . 631
30.134.1.1 fdt_domain_fixup() . . . 631
30.134.1.2 fdt_domains_populate() . . . 631
30.135 fdt_domain.h . . . 631
30.136 include/sbi_utils/fdt/fdt_driver.h File Reference . . . 632
30.136.1 Function Documentation . . . 632
30.136.1.1 fdt_driver_init_all() . . . 632

30.136.1.2 fdt_driver_init_by_offset() . . . 633
30.136.1.3 fdt_driver_init_one() . . . 634
30.137 fdt_driver.h . . . 635
30.138 include/sbi_utils/fdt/fdt_fixup.h File Reference . . . 636
30.138.1 Function Documentation . . . 637
30.138.1.1 fdt_add_cpu_idle_states() . . . 637
30.138.1.2 fdt_aplic_fixup() . . . 637
30.138.1.3 fdt_cpu_fixup() . . . 638
30.138.1.4 fdt_fixups() . . . 639
30.138.1.5 fdt_imsic_fixup() . . . 639
30.138.1.6 fdt_plic_fixup() . . . 640
30.138.1.7 fdt_register_general_fixup() . . . 641
30.138.1.8 fdt_reserved_memory_fixup() . . . 641
30.138.1.9 fdt_unregister_general_fixup() . . . 642
30.139 fdt_fixup.h . . . 643
30.140 include/sbi_utils/fdt/fdt_helper.h File Reference . . . 643
30.140.1 Macro Definition Documentation . . . 645
30.140.1.1 FDT_MAX_PHANDLE_ARGS . . . 645
30.140.2 Function Documentation . . . 645
30.140.2.1 fdt_check_imsic_mlevel() . . . 645
30.140.2.2 fdt_find_match() . . . 645
30.140.2.3 fdt_get_address() . . . 646
30.140.2.4 fdt_get_address_rw() . . . 646
30.140.2.5 fdt_get_node_addr_size() . . . 646
30.140.2.6 fdt_get_node_addr_size_by_name() . . . 647
30.140.2.7 fdt_match_node() . . . 648
30.140.2.8 fdt_node_is_enabled() . . . 648
30.140.2.9 fdt_parse_aclint_node() . . . 649
30.140.2.10 fdt_parse_aplic_node() . . . 650
30.140.2.11 fdt_parse_compat_addr() . . . 650
30.140.2.12 fdt_parse_gaisler_uart_node() . . . 651
30.140.2.13 fdt_parse_hart_id() . . . 652
30.140.2.14 fdt_parse_imsic_node() . . . 652
30.140.2.15 fdt_parse_isa_extensions() . . . 653
30.140.2.16 fdt_parse_max_enabled_hart_id() . . . 653
30.140.2.17 fdt_parse_phandle_with_args() . . . 654
30.140.2.18 fdt_parse_plic() . . . 654
30.140.2.19 fdt_parse_plic_node() . . . 655
30.140.2.20 fdt_parse_plicsw_node() . . . 655
30.140.2.21 fdt_parse_plmt_node() . . . 656
30.140.2.22 fdt_parse_renesas_scif_node() . . . 657
30.140.2.23 fdt_parse_shakti_uart_node() . . . 657

30.140.2.24 fdt_parse_sifive_uart_node() . . . 658
30.140.2.25 fdt_parse_timebase_frequency() . . . 659
30.140.2.26 fdt_parse_uart8250() . . . 659
30.140.2.27 fdt_parse_uart_node() . . . 659
30.140.2.28 fdt_parse_xlnx_uartlite_node() . . . 660
30.141 fdt_helper.h . . . 661
30.142 include/sbi_utils/fdt/fdt_pmu.h File Reference . . . 663
30.142.1 Function Documentation . . . 663
30.142.1.1 fdt_pmu_fixup() . . . 663
30.142.1.2 fdt_pmu_get_select_value() . . . 664
30.142.1.3 fdt_pmu_setup() . . . 664
30.143 fdt_pmu.h . . . 664
30.144 include/sbi_utils/gpio/fdt_gpio.h File Reference . . . 665
30.144.1 Function Documentation . . . 665
30.144.1.1 fdt_gpio_pin_get() . . . 665
30.144.1.2 fdt_gpio_simple_xlate() . . . 666
30.145 fdt_gpio.h . . . 666
30.146 include/sbi_utils/gpio/gpio.h File Reference . . . 667
30.146.1 Macro Definition Documentation . . . 668
30.146.1.1 GPIO_FLAG_ACTIVE_LOW . . . 668
30.146.1.2 GPIO_FLAG_OPEN_DRAIN . . . 668
30.146.1.3 GPIO_FLAG_PULL_DOWN . . . 668
30.146.1.4 GPIO_FLAG_PULL_UP . . . 668
30.146.1.5 GPIO_FLAG_SINGLE_ENDED . . . 668
30.146.1.6 GPIO_FLAG_TRANSITORY . . . 668
30.146.1.7 GPIO_LINE_DIRECTION_IN . . . 668
30.146.1.8 GPIO_LINE_DIRECTION_OUT . . . 668
30.146.2 Function Documentation . . . 669
30.146.2.1 gpio_chip_add() . . . 669
30.146.2.2 gpio_chip_find() . . . 669
30.146.2.3 gpio_chip_remove() . . . 670
30.146.2.4 gpio_direction_input() . . . 670
30.146.2.5 gpio_direction_output() . . . 670
30.146.2.6 gpio_get() . . . 671
30.146.2.7 gpio_get_direction() . . . 671
30.146.2.8 gpio_set() . . . 671
30.146.2.9 to_gpio_chip() . . . 671
30.147 gpio.h . . . 672
30.148 include/sbi_utils/hsm/fdt_hsm.h File Reference . . . 672
30.148.1 Function Documentation . . . 673
30.148.1.1 fdt_hsm_init() . . . 673
30.149 fdt_hsm.h . . . 673

30.150 include/sbi_utils/i2c/dw_i2c.h File Reference . . . 674
30.150.1 Function Documentation . . . 675
30.150.1.1 dw_i2c_init() . . . 675
30.151 dw_i2c.h . . . 676
30.152 include/sbi_utils/i2c/fdt_i2c.h File Reference . . . 676
30.152.1 Function Documentation . . . 677
30.152.1.1 fdt_i2c_adapter_get() . . . 677
30.153 fdt_i2c.h . . . 677
30.154 include/sbi_utils/i2c/i2c.h File Reference . . . 678
30.154.1 Function Documentation . . . 679
30.154.1.1 i2c_adapter_add() . . . 679
30.154.1.2 i2c_adapter_find() . . . 679
30.154.1.3 i2c_adapter_read() . . . 680
30.154.1.4 i2c_adapter_reg_read() . . . 680
30.154.1.5 i2c_adapter_reg_write() . . . 681
30.154.1.6 i2c_adapter_remove() . . . 681
30.154.1.7 i2c_adapter_write() . . . 681
30.154.1.8 to_i2c_adapter() . . . 682
30.155 i2c.h . . . 682
30.156 include/sbi_utils/ipi/aclint_mswi.h File Reference . . . 683
30.156.1 Macro Definition Documentation . . . 684
30.156.1.1 ACLINT_MSWI_ALIGN . . . 684
30.156.1.2 ACLINT_MSWI_MAX_HARTS . . . 684
30.156.1.3 ACLINT_MSWI_SIZE . . . 684
30.156.1.4 CLINT_MSWI_OFFSET . . . 684
30.156.2 Function Documentation . . . 684
30.156.2.1 aclint_mswi_cold_init() . . . 684
30.157 aclint_mswi.h . . . 685
30.158 include/sbi_utils/ipi/andes_plicsw.h File Reference . . . 685
30.158.1 Macro Definition Documentation . . . 686
30.158.1.1 PLICSW_CONTEXT_BASE . . . 686
30.158.1.2 PLICSW_CONTEXT_CLAIM . . . 686
30.158.1.3 PLICSW_CONTEXT_STRIDE . . . 686
30.158.1.4 PLICSW_ENABLE_BASE . . . 686
30.158.1.5 PLICSW_ENABLE_STRIDE . . . 686
30.158.1.6 PLICSW_PENDING_BASE . . . 686
30.158.1.7 PLICSW_PRIORITY_BASE . . . 686
30.158.1.8 PLICSW_REGION_ALIGN . . . 686
30.158.2 Function Documentation . . . 686
30.158.2.1 plicsw_cold_ipi_init() . . . 686
30.159 andes_plicsw.h . . . 687
30.160 include/sbi_utils/ipi/fdt_ipi.h File Reference . . . 687

30.160.1 Function Documentation . . . 688
30.160.1.1 fdt_ipi_init() . . . 688
30.161 fdt_ipi.h . . . 688
30.162 include/sbi_utils/irqchip/aplic.h File Reference . . . 689
30.162.1 Macro Definition Documentation . . . 690
30.162.1.1 APLIC_MAX_DELEGATE . . . 690
30.162.2 Function Documentation . . . 690
30.162.2.1 aplic_cold_irqchip_init() . . . 690
30.163 aplic.h . . . 691
30.164 include/sbi_utils/irqchip/fdt_irqchip.h File Reference . . . 692
30.164.1 Function Documentation . . . 692
30.164.1.1 fdt_irqchip_init() . . . 692
30.165 fdt_irqchip.h . . . 692
30.166 include/sbi_utils/irqchip/imsic.h File Reference . . . 693
30.166.1 Macro Definition Documentation . . . 694
30.166.1.1 IMSIC_MAX_REGS . . . 694
30.166.1.2 IMSIC_MMIO_PAGE_SHIFT . . . 694
30.166.1.3 IMSIC_MMIO_PAGE_SZ . . . 694
30.166.2 Function Documentation . . . 694
30.166.2.1 imsic_data_check() . . . 694
30.166.2.2 imsic_local_irqchip_init() . . . 694
30.167 imsic.h . . . 694
30.168 include/sbi_utils/irqchip/plic.h File Reference . . . 695
30.168.1 Macro Definition Documentation . . . 696
30.168.1.1 PLIC_DATA_SIZE . . . 696
30.168.1.2 PLIC_FLAG_ARIANE_BUG . . . 696
30.168.1.3 PLIC_FLAG_ENABLE_PM . . . 696
30.168.1.4 PLIC_FLAG_THEAD_DELEGATION . . . 696
30.168.1.5 PLIC_IE_WORDS . . . 697
30.168.1.6 PLIC_M_CONTEXT . . . 697
30.168.1.7 PLIC_S_CONTEXT . . . 697
30.168.2 Function Documentation . . . 697
30.168.2.1 plic_cold_irqchip_init() . . . 697
30.168.2.2 plic_get() . . . 697
30.168.2.3 plic_resume() . . . 697
30.168.2.4 plic_suspend() . . . 698
30.169 plic.h . . . 698
30.170 include/sbi_utils/mailbox/fdt_mailbox.h File Reference . . . 699
30.170.1 Function Documentation . . . 700
30.170.1.1 fdt_mailbox_request_chan() . . . 700
30.170.1.2 fdt_mailbox_simple_xlate() . . . 701
30.171 fdt_mailbox.h . . . 701

30.172 include/sbi_utils/mailbox/mailbox.h File Reference . . . 701
30.172.1 Macro Definition Documentation . . . 703
30.172.1.1 MBOX_CHAN_MAX_ARGS . . . 703
30.172.1.2 mbox_xfer_init_rx . . . 703
30.172.1.3 mbox_xfer_init_tx . . . 703
30.172.1.4 mbox_xfer_init_txrx . . . 703
30.172.1.5 MBOX_XFER_SEQ . . . 704
30.172.1.6 mbox_xfer_set_sequence . . . 704
30.172.1.7 to_mbox_chan . . . 704
30.172.1.8 to_mbox_controller . . . 704
30.172.2 Function Documentation . . . 704
30.172.2.1 mbox_chan_get_attribute() . . . 704
30.172.2.2 mbox_chan_set_attribute() . . . 704
30.172.2.3 mbox_chan_xfer() . . . 704
30.172.2.4 mbox_controller_add() . . . 705
30.172.2.5 mbox_controller_find() . . . 706
30.172.2.6 mbox_controller_free_chan() . . . 706
30.172.2.7 mbox_controller_remove() . . . 707
30.172.2.8 mbox_controller_request_chan() . . . 707
30.173 mailbox.h . . . 708
30.174 include/sbi_utils/mailbox/rpmi_mailbox.h File Reference . . . 710
30.174.1 Macro Definition Documentation . . . 711
30.174.1.1 rpmi_u32_count . . . 711
30.174.2 Function Documentation . . . 711
30.174.2.1 rpmi_normal_request_with_status() . . . 711
30.174.2.2 rpmi_posted_request() . . . 712
30.174.2.3 rpmi_xlate_error() . . . 713
30.175 rpmi_mailbox.h . . . 713
30.176 include/sbi_utils/mailbox/rpmi_msgprot.h File Reference . . . 714
30.176.1 Macro Definition Documentation . . . 718
30.176.1.1 RPMI_BASE_FLAGS_F0_EV_NOTIFY . . . 718
30.176.1.2 RPMI_BASE_FLAGS_F0_MSI_EN . . . 718
30.176.1.3 RPMI_BASE_FLAGS_F0_PRIVILEGE . . . 718
30.176.1.4 RPMI_CLOCK_CONFIG_ENABLE . . . 718
30.176.1.5 RPMI_CLOCK_FLAGS_FORMAT_DISCRETE . . . 718
30.176.1.6 RPMI_CLOCK_FLAGS_FORMAT_LINEAR . . . 718
30.176.1.7 RPMI_CLOCK_FLAGS_FORMAT_MASK . . . 718
30.176.1.8 RPMI_CLOCK_FLAGS_FORMAT_POS . . . 718
30.176.1.9 RPMI_CLOCK_SET_RATE_FLAGS_MASK . . . 718
30.176.1.10 RPMI_CLOCK_SET_RATE_FLAGS_ROUND_DOWN . . . 718
30.176.1.11 RPMI_CLOCK_SET_RATE_FLAGS_ROUND_PLAT . . . 718
30.176.1.12 RPMI_CLOCK_SET_RATE_FLAGS_ROUND_UP . . . 718

30.176.1.13 RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MASK 718
30.176.1.14 RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS 719
30.176.1.15 RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_SUPPORTED 719
30.176.1.16 RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_MASK 719
30.176.1.17 RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS 719
30.176.1.18 RPMI_DEF_RX_TIMEOUT 719
30.176.1.19 RPMI_DEF_TX_TIMEOUT 719
30.176.1.20 RPMI_HSM_SUSPEND_INFO_FLAGS_TIMER_STOP 719
30.176.1.21 RPMI_MSG_DATA_OFFSET 719
30.176.1.22 RPMI_MSG_DATA_SIZE 719
30.176.1.23 RPMI_MSG_DATALEN_OFFSET 719
30.176.1.24 RPMI_MSG_DATALEN_SIZE 719
30.176.1.25 RPMI_MSG_FLAGS_DOORBELL 719
30.176.1.26 RPMI_MSG_FLAGS_DOORBELL_MASK 719
30.176.1.27 RPMI_MSG_FLAGS_DOORBELL_POS 719
30.176.1.28 RPMI_MSG_FLAGS_NO_RX 720
30.176.1.29 RPMI_MSG_FLAGS_NO_RX_TOKEN 720
30.176.1.30 RPMI_MSG_FLAGS_NO_TX 720
30.176.1.31 RPMI_MSG_FLAGS_OFFSET 720
30.176.1.32 RPMI_MSG_FLAGS_SIZE 720
30.176.1.33 RPMI_MSG_FLAGS_TYPE 720
30.176.1.34 RPMI_MSG_FLAGS_TYPE_MASK 720
30.176.1.35 RPMI_MSG_FLAGS_TYPE_POS 720
30.176.1.36 RPMI_MSG_HDR_OFFSET 720
30.176.1.37 RPMI_MSG_HDR_SIZE 720
30.176.1.38 RPMI_MSG_SERVICE_ID_OFFSET 720
30.176.1.39 RPMI_MSG_SERVICE_ID_SIZE 720
30.176.1.40 RPMI_MSG_SERVICEGROUP_ID_OFFSET 720
30.176.1.41 RPMI_MSG_SERVICEGROUP_ID_SIZE 721
30.176.1.42 RPMI_MSG_TOKEN_MASK 721
30.176.1.43 RPMI_MSG_TOKEN_OFFSET 721
30.176.1.44 RPMI_MSG_TOKEN_SIZE 721
30.176.1.45 RPMI_NAME_CHARS_MAX 721
30.176.1.46 RPMI_QUEUE_HEAD_SLOT 721
30.176.1.47 RPMI_QUEUE_HEADER_SLOTS 721
30.176.1.48 RPMI_QUEUE_TAIL_SLOT 721
30.176.1.49 RPMI_SLOT_SIZE_MIN 721
30.176.1.50 RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_MASK 721
30.176.1.51 RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS 721
30.176.1.52 RPMI_SYSSUSP_ATTRS_FLAGS_RESUMEADDR 721
30.176.1.53 RPMI_SYSSUSP_ATTRS_FLAGS_SUSPENDTYPE 721
30.176.1.54 RPMI_VERSION 722

30.176.2 Enumeration Type Documentation . . . 722
30.176.2.1 rpmi_base_context_priv_level . . . 722
30.176.2.2 rpmi_base_service_id . . . 722
30.176.2.3 rpmi_clock_service_id . . . 722
30.176.2.4 rpmi_cppc_fast_channel_cppc_mode . . . 723
30.176.2.5 rpmi_cppc_fast_channel_db_width . . . 723
30.176.2.6 rpmi_cppc_service_id . . . 723
30.176.2.7 rpmi_error . . . 723
30.176.2.8 rpmi_hsm_service_id . . . 724
30.176.2.9 rpmi_message_type . . . 724
30.176.2.10 rpmi_servicegroup_id . . . 724
30.176.2.11 rpmi_sysrst_reset_type . . . 725
30.176.2.12 rpmi_system_reset_service_id . . . 725
30.176.2.13 rpmi_system_suspend_service_id . . . 725
30.176.3 Variable Documentation . . . 725
30.176.3.1 __packed . . . 725
30.177 rpmi_msgprot.h . . . 726
30.178 include/sbi_utils/mpxy/fdt_mpxy.h File Reference . . . 732
30.178.1 Function Documentation . . . 733
30.178.1.1 fdt_mpxy_init() . . . 733
30.179 fdt_mpxy.h . . . 733
30.180 include/sbi_utils/regmap/fdt_regmap.h File Reference . . . 733
30.180.1 Function Documentation . . . 734
30.180.1.1 fdt_regmap_get() . . . 734
30.180.1.2 fdt_regmap_get_by_phandle() . . . 735
30.181 fdt_regmap.h . . . 735
30.182 include/sbi_utils/regmap/regmap.h File Reference . . . 736
30.182.1 Function Documentation . . . 737
30.182.1.1 regmap_add() . . . 737
30.182.1.2 regmap_find() . . . 738
30.182.1.3 regmap_read() . . . 738
30.182.1.4 regmap_remove() . . . 739
30.182.1.5 regmap_update_bits() . . . 739
30.182.1.6 regmap_write() . . . 740
30.182.1.7 to_regmap() . . . 740
30.183 regmap.h . . . 740
30.184 include/sbi_utils/reset/fdt_reset.h File Reference . . . 741
30.184.1 Function Documentation . . . 742
30.184.1.1 fdt_reset_init() . . . 742
30.185 fdt_reset.h . . . 742
30.186 include/sbi_utils/serial/cadence-uart.h File Reference . . . 743
30.186.1 Function Documentation . . . 743

30.186.1.1 cadence_uart_init() . . . 743
30.187 cadence-uart.h . . . 744
30.188 include/sbi_utils/serial/fdt_serial.h File Reference . . . 744
30.188.1 Function Documentation . . . 745
30.188.1.1 fdt_serial_init() . . . 745
30.189 fdt_serial.h . . . 745
30.190 include/sbi_utils/serial/gaisler-uart.h File Reference . . . 746
30.190.1 Function Documentation . . . 746
30.190.1.1 gaisler_uart_init() . . . 746
30.191 gaisler-uart.h . . . 747
30.192 include/sbi_utils/serial/litex-uart.h File Reference . . . 748
30.192.1 Function Documentation . . . 748
30.192.1.1 litex_uart_init() . . . 748
30.193 litex-uart.h . . . 749
30.194 include/sbi_utils/serial/renesas-scif.h File Reference . . . 750
30.194.1 Function Documentation . . . 750
30.194.1.1 renesas_scif_init() . . . 750
30.195 renesas-scif.h . . . 751
30.196 include/sbi_utils/serial/semihosting.h File Reference . . . 751
30.196.1 Enumeration Type Documentation . . . 752
30.196.1.1 semihosting_open_mode . . . 752
30.196.2 Function Documentation . . . 752
30.196.2.1 semihosting_enabled() . . . 752
30.196.2.2 semihosting_init() . . . 752
30.197 semihosting.h . . . 752
30.198 include/sbi_utils/serial/shakti-uart.h File Reference . . . 753
30.198.1 Function Documentation . . . 753
30.198.1.1 shakti_uart_init() . . . 753
30.199 shakti-uart.h . . . 754
30.200 include/sbi_utils/serial/sifive-uart.h File Reference . . . 754
30.200.1 Function Documentation . . . 755
30.200.1.1 sifive_uart_init() . . . 755
30.201 sifive-uart.h . . . 755
30.202 include/sbi_utils/serial/uart8250.h File Reference . . . 756
30.202.1 Function Documentation . . . 757
30.202.1.1 uart8250_init() . . . 757
30.203 uart8250.h . . . 757
30.204 include/sbi_utils/serial/xlnx_uartlite.h File Reference . . . 758
30.204.1 Function Documentation . . . 758
30.204.1.1 xlnx_uartlite_init() . . . 758
30.205 xlnx_uartlite.h . . . 759
30.206 include/sbi_utils/suspend/fdt_suspend.h File Reference . . . 759

30.206.1 Function Documentation . . . 760
30.206.1.1 fdt_suspend_init() . . . 760
30.207 fdt_suspend.h . . . 760
30.208 include/sbi_utils/sys/atcsmu.h File Reference . . . 761
30.208.1 Macro Definition Documentation . . . 762
30.208.1.1 DEEP_SLEEP_CMD . . . 762
30.208.1.2 DEEPSLEEP_MODE . . . 762
30.208.1.3 FLASH_BASE . . . 762
30.208.1.4 HARTn_RESET_VEC_HI . . . 762
30.208.1.5 HARTn_RESET_VEC_LO . . . 763
30.208.1.6 LIGHT_SLEEP_CMD . . . 763
30.208.1.7 LIGHTSLEEP_MODE . . . 763
30.208.1.8 PCS0_CFG_OFFSET . . . 763
30.208.1.9 PCS0_CTL_OFFSET . . . 763
30.208.1.10 PCS0_WE_OFFSET . . . 763
30.208.1.11 PCS_CFG_DEEP_SLEEP . . . 763
30.208.1.12 PCS_CFG_DEEP_SLEEP_SHIFT . . . 763
30.208.1.13 PCS_CFG_LIGHT_SLEEP . . . 763
30.208.1.14 PCS_CFG_LIGHT_SLEEP_SHIFT . . . 763
30.208.1.15 PCS_CTL_CMD_SHIFT . . . 763
30.208.1.16 PCS_CTL_PARAM_SHIFT . . . 763
30.208.1.17 PCS_MAX_NR . . . 763
30.208.1.18 PCSm_CFG_OFFSET . . . 763
30.208.1.19 PCSm_CTL_OFFSET . . . 764
30.208.1.20 PCSm_WE_OFFSET . . . 764
30.208.1.21 RESET_VEC_8CORE_OFFSET . . . 764
30.208.1.22 RESET_VEC_HI_OFFSET . . . 764
30.208.1.23 RESET_VEC_LO_OFFSET . . . 764
30.208.1.24 SLEEP_CMD . . . 764
30.208.1.25 WAKEUP_CMD . . . 764
30.208.2 Function Documentation . . . 764
30.208.2.1 smu_set_command() . . . 764
30.208.2.2 smu_set_reset_vector() . . . 764
30.208.2.3 smu_set_wakeup_events() . . . 765
30.208.2.4 smu_support_sleep_mode() . . . 765
30.209 atcsmu.h . . . 765
30.210 include/sbi_utils/sys/htif.h File Reference . . . 766
30.210.1 Function Documentation . . . 767
30.210.1.1 htif_serial_init() . . . 767
30.210.1.2 htif_system_reset_init() . . . 767
30.211 htif.h . . . 768
30.212 include/sbi_utils/timer/aclint_mtimer.h File Reference . . . 768

30.212.1 Macro Definition Documentation . . . 769
30.212.1.1 ACLINT_DEFAULT_MTIME_OFFSET . . . 769
30.212.1.2 ACLINT_DEFAULT_MTIME_SIZE . . . 769
30.212.1.3 ACLINT_DEFAULT_MTIMECMP_OFFSET . . . 769
30.212.1.4 ACLINT_DEFAULT_MTIMECMP_SIZE . . . 769
30.212.1.5 ACLINT_MTIMER_ALIGN . . . 770
30.212.1.6 ACLINT_MTIMER_MAX_HARTS . . . 770
30.212.1.7 CLINT_MTIMER_OFFSET . . . 770
30.212.1.8 MTIMER_REGION_ALIGN . . . 770
30.212.2 Function Documentation . . . 770
30.212.2.1 aclint_mtimer_cold_init() . . . 770
30.212.2.2 aclint_mtimer_set_reference() . . . 770
30.212.2.3 aclint_mtimer_sync() . . . 771
30.213 aclint_mtimer.h . . . 771
30.214 include/sbi_utils/timer/andes_plmt.h File Reference . . . 772
30.214.1 Macro Definition Documentation . . . 772
30.214.1.1 DEFAULT_AE350_PLMT_FREQ . . . 772
30.214.1.2 PLMT_REGION_ALIGN . . . 773
30.214.2 Function Documentation . . . 773
30.214.2.1 plmt_cold_timer_init() . . . 773
30.215 andes_plmt.h . . . 773
30.216 include/sbi_utils/timer/fdt_timer.h File Reference . . . 774
30.216.1 Function Documentation . . . 774
30.216.1.1 fdt_timer_init() . . . 774
30.217 fdt_timer.h . . . 774
30.218 lib/sbi/riscv_asm.c File Reference . . . 775
30.218.1 Macro Definition Documentation . . . 776
30.218.1.1 switchcase_csr_read . . . 776
30.218.1.2 switchcase_csr_read_16 . . . 776
30.218.1.3 switchcase_csr_read_2 . . . 776
30.218.1.4 switchcase_csr_read_32 . . . 777
30.218.1.5 switchcase_csr_read_4 . . . 777
30.218.1.6 switchcase_csr_read_64 . . . 777
30.218.1.7 switchcase_csr_read_8 . . . 777
30.218.1.8 switchcase_csr_write . . . 777
30.218.1.9 switchcase_csr_write_16 . . . 777
30.218.1.10 switchcase_csr_write_2 . . . 777
30.218.1.11 switchcase_csr_write_32 . . . 778
30.218.1.12 switchcase_csr_write_4 . . . 778
30.218.1.13 switchcase_csr_write_64 . . . 778
30.218.1.14 switchcase_csr_write_8 . . . 778
30.218.2 Function Documentation . . . 778

30.218.2.1 csr_read_num() . . . 778
30.218.2.2 csr_write_num() . . . 779
30.218.2.3 ctz() . . . 780
30.218.2.4 is_pmp_entry_mapped() . . . 780
30.218.2.5 misa_extension_imp() . . . 781
30.218.2.6 misa_string() . . . 781
30.218.2.7 misa_xlen() . . . 781
30.218.2.8 pmp_disable() . . . 782
30.218.2.9 pmp_get() . . . 783
30.218.2.10 pmp_set() . . . 784
30.219 lib/sbi/riscv_atomic.c File Reference . . . 785
30.219.1 Macro Definition Documentation . . . 785
30.219.1.1 __axchg . . . 785
30.219.1.2 axchg . . . 786
30.219.2 Function Documentation . . . 786
30.219.2.1 atomic_add_return() . . . 786
30.219.2.2 atomic_clear_bit() . . . 786
30.219.2.3 atomic_cmpxchg() . . . 787
30.219.2.4 atomic_raw_clear_bit() . . . 787
30.219.2.5 atomic_raw_set_bit() . . . 788
30.219.2.6 atomic_raw_xchg_uint() . . . 788
30.219.2.7 atomic_raw_xchg_ulong() . . . 788
30.219.2.8 atomic_read() . . . 789
30.219.2.9 atomic_set_bit() . . . 789
30.219.2.10 atomic_sub_return() . . . 789
30.219.2.11 atomic_write() . . . 790
30.219.2.12 atomic_xchg() . . . 790
30.220 lib/sbi/riscv_locks.c File Reference . . . 791
30.220.1 Function Documentation . . . 791
30.220.1.1 spin_lock() . . . 791
30.220.1.2 spin_lock_check() . . . 792
30.220.1.3 spin_lock_unlocked() . . . 793
30.220.1.4 spin_trylock() . . . 793
30.220.1.5 spin_unlock() . . . 794
30.221 lib/sbi/sbi_bitmap.c File Reference . . . 795
30.221.1 Function Documentation . . . 796
30.221.1.1 __bitmap_and() . . . 796
30.221.1.2 __bitmap_or() . . . 796
30.221.1.3 __bitmap_xor() . . . 797
30.222 lib/sbi/sbi_bitops.c File Reference . . . 797
30.222.1 Macro Definition Documentation . . . 798
30.222.1.1 BITOP_WORD . . . 798

30.222.2 Function Documentation . . . 798
30.222.2.1 find_first_bit() . . . 798
30.222.2.2 find_first_zero_bit() . . . 798
30.222.2.3 find_last_bit() . . . 798
30.222.2.4 find_next_bit() . . . 799
30.222.2.5 find_next_zero_bit() . . . 799
30.223 lib/sbi/sbi_console.c File Reference . . . 799
30.223.1 Macro Definition Documentation . . . 801
30.223.1.1 CONSOLE_EARLY_BUFFER_SIZE . . . 801
30.223.1.2 CONSOLE_TBUF_MAX . . . 801
30.223.1.3 PAD_ALTERNATE . . . 801
30.223.1.4 PAD_RIGHT . . . 801
30.223.1.5 PAD_SIGN . . . 801
30.223.1.6 PAD_ZERO . . . 801
30.223.1.7 PRINT_BUF_LEN . . . 801
30.223.1.8 USE_TBUF . . . 801
30.223.1.9 va_arg . . . 801
30.223.1.10 va_end . . . 801
30.223.1.11 va_start . . . 801
30.223.2 Typedef Documentation . . . 802
30.223.2.1 va_list . . . 802
30.223.3 Function Documentation . . . 802
30.223.3.1 nputs() . . . 802
30.223.3.2 nputs_all() . . . 803
30.223.3.3 print() . . . 804
30.223.3.4 printc() . . . 805
30.223.3.5 printi() . . . 807
30.223.3.6 prints() . . . 809
30.223.3.7 sbi_console_get_device() . . . 811
30.223.3.8 sbi_console_set_device() . . . 812
30.223.3.9 sbi_dprintf() . . . 813
30.223.3.10 SBI_FIFO_DEFINE() . . . 814
30.223.3.11 sbi_getc() . . . 814
30.223.3.12 sbi_gets() . . . 814
30.223.3.13 sbi_isprintable() . . . 815
30.223.3.14 sbi_ngets() . . . 815
30.223.3.15 sbi_nputs() . . . 815
30.223.3.16 sbi_panic() . . . 816
30.223.3.17 sbi_printf() . . . 817
30.223.3.18 sbi_putc() . . . 818
30.223.3.19 sbi_puts() . . . 819
30.223.3.20 sbi_snprintf() . . . 820

30.223.3.21 sbi_sprintf() . . . 820
30.223.4 Variable Documentation . . . 821
30.223.4.1 console_dev . . . 821
30.223.4.2 console_early_buffer . . . 821
30.223.4.3 console_out_lock . . . 821
30.223.4.4 console_tbuf . . . 821
30.223.4.5 console_tbuf_len . . . 821
30.224 lib/sbi/sbi_cppc.c File Reference . . . 821
30.224.1 Function Documentation . . . 822
30.224.1.1 sbi_cppc_get_device() . . . 822
30.224.1.2 sbi_cppc_is_reserved() . . . 823
30.224.1.3 sbi_cppc_probe() . . . 823
30.224.1.4 sbi_cppc_read() . . . 823
30.224.1.5 sbi_cppc_readable() . . . 824
30.224.1.6 sbi_cppc_set_device() . . . 824
30.224.1.7 sbi_cppc_writable() . . . 825
30.224.1.8 sbi_cppc_write() . . . 825
30.224.2 Variable Documentation . . . 825
30.224.2.1 cppc_dev . . . 825
30.225 lib/sbi/sbi_dbtr.c File Reference . . . 825
30.225.1 Macro Definition Documentation . . . 827
30.225.1.1 dbtr_get_hart_state_ptr . . . 827
30.225.1.2 dbtr_set_hart_state_ptr . . . 827
30.225.1.3 DBTR_SHMEM_MAKE_PHYS . . . 827
30.225.1.4 dbtr_thishart_state_ptr . . . 827
30.225.1.5 for_each_trig_entry . . . 827
30.225.1.6 INDEX_TO_TRIGGER . . . 827
30.225.2 Function Documentation . . . 827
30.225.2.1 dbtr_trigger_clear() . . . 827
30.225.2.2 dbtr_trigger_disable() . . . 828
30.225.2.3 dbtr_trigger_enable() . . . 828
30.225.2.4 dbtr_trigger_setup() . . . 829
30.225.2.5 dbtr_trigger_supported() . . . 829
30.225.2.6 dbtr_trigger_valid() . . . 829
30.225.2.7 hart_shmem_base() . . . 830
30.225.2.8 sbi_alloc_trigger() . . . 830
30.225.2.9 sbi_dbtr_disable_shmem() . . . 831
30.225.2.10 sbi_dbtr_disable_trig() . . . 831
30.225.2.11 sbi_dbtr_enable_trig() . . . 831
30.225.2.12 sbi_dbtr_get_total_triggers() . . . 832
30.225.2.13 sbi_dbtr_init() . . . 832
30.225.2.14 sbi_dbtr_install_trig() . . . 833

30.225.2.15 sbi_dbtr_num_trig() . . . 833
30.225.2.16 sbi_dbtr_read_trig() . . . 834
30.225.2.17 sbi_dbtr_setup_shmem() . . . 834
30.225.2.18 sbi_dbtr_shmem_disabled() . . . 835
30.225.2.19 sbi_dbtr_uninstall_trig() . . . 835
30.225.2.20 sbi_dbtr_update_trig() . . . 836
30.225.2.21 sbi_free_trigger() . . . 837
30.225.2.22 sbi_trigger_init() . . . 837
30.225.2.23 update_bit() . . . 837
30.225.3 Variable Documentation . . . 838
30.225.3.1 hart_state_ptr_offset . . . 838
30.226 lib/sbi/sbi_domain.c File Reference . . . 838
30.226.1 Macro Definition Documentation . . . 839
30.226.1.1 ROOT_REGION_MAX . . . 839
30.226.2 Function Documentation . . . 839
30.226.2.1 clear_region() . . . 839
30.226.2.2 find_next_subset_region() . . . 840
30.226.2.3 find_region() . . . 840
30.226.2.4 is_region_before() . . . 841
30.226.2.5 is_region_compatible() . . . 841
30.226.2.6 is_region_subset() . . . 842
30.226.2.7 is_region_valid() . . . 843
30.226.2.8 root_add_memregion() . . . 843
30.226.2.9 sanitize_domain() . . . 844
30.226.2.10 sbi_domain_check_addr() . . . 845
30.226.2.11 sbi_domain_check_addr_range() . . . 846
30.226.2.12 sbi_domain_dump() . . . 847
30.226.2.13 sbi_domain_dump_all() . . . 848
30.226.2.14 sbi_domain_finalize() . . . 848
30.226.2.15 sbi_domain_get_assigned_hartmask() . . . 849
30.226.2.16 sbi_domain_init() . . . 850
30.226.2.17 sbi_domain_is_assigned_hart() . . . 850
30.226.2.18 sbi_domain_memregion_init() . . . 851
30.226.2.19 sbi_domain_register() . . . 852
30.226.2.20 sbi_domain_root_add_memrange() . . . 853
30.226.2.21 sbi_hartindex_to_domain() . . . 854
30.226.2.22 SBI_LIST_HEAD() . . . 855
30.226.2.23 sbi_update_hartindex_to_domain() . . . 855
30.226.2.24 swap_region() . . . 855
30.226.3 Variable Documentation . . . 856
30.226.3.1 domain_count . . . 856
30.226.3.2 domain_finalized . . . 856

30.226.3.3 domain_hart_ptr_offset . . . 856
30.226.3.4 root . . . 856
30.226.3.5 root_memregs_count . . . 856
30.227 lib/sbi/sbi_domain_context.c File Reference . . . 856
30.227.1 Macro Definition Documentation . . . 857
30.227.1.1 hart_context_thishart_get . . . 857
30.227.2 Function Documentation . . . 857
30.227.2.1 hart_context_get() . . . 857
30.227.2.2 hart_context_set() . . . 858
30.227.2.3 sbi_domain_context_deinit() . . . 859
30.227.2.4 sbi_domain_context_enter() . . . 859
30.227.2.5 sbi_domain_context_exit() . . . 860
30.227.2.6 sbi_domain_context_init() . . . 860
30.227.2.7 switch_to_next_domain_context() . . . 861
30.227.3 Variable Documentation . . . 861
30.227.3.1 dcpriv . . . 861
30.228 lib/sbi/sbi_domain_data.c File Reference . . . 862
30.228.1 Function Documentation . . . 862
30.228.1.1 DECLARE_BITMAP() . . . 862
30.228.1.2 domain_cleanup_data_one() . . . 862
30.228.1.3 domain_setup_data_one() . . . 863
30.228.1.4 sbi_domain_cleanup_data() . . . 863
30.228.1.5 sbi_domain_data_ptr() . . . 864
30.228.1.6 sbi_domain_register_data() . . . 864
30.228.1.7 sbi_domain_setup_data() . . . 865
30.228.1.8 sbi_domain_unregister_data() . . . 866
30.228.1.9 SBI_LIST_HEAD() . . . 866
30.229 lib/sbi/sbi_double_trap.c File Reference . . . 866
30.229.1 Function Documentation . . . 867
30.229.1.1 sbi_double_trap_handler() . . . 867
30.230 lib/sbi/sbi_ecall.c File Reference . . . 868
30.230.1 Function Documentation . . . 868
30.230.1.1 sbi_ecall_find_extension() . . . 868
30.230.1.2 sbi_ecall_get_extensions_str() . . . 869
30.230.1.3 sbi_ecall_get_impid() . . . 869
30.230.1.4 sbi_ecall_handler() . . . 870
30.230.1.5 sbi_ecall_init() . . . 870
30.230.1.6 sbi_ecall_register_extension() . . . 870
30.230.1.7 sbi_ecall_set_impid() . . . 871
30.230.1.8 sbi_ecall_unregister_extension() . . . 871
30.230.1.9 sbi_ecall_version_major() . . . 872
30.230.1.10 sbi_ecall_version_minor() . . . 872

30.230.1.11 SBI_LIST_HEAD() . . . 872
30.230.2 Variable Documentation . . . 872
30.230.2.1 ecall_impid . . . 872
30.230.2.2 sbi_ecall_exts . . . 872
30.231 lib/sbi/sbi_ecall_base.c File Reference . . . 872
30.231.1 Function Documentation . . . 873
30.231.1.1 sbi_ecall_base_handler() . . . 873
30.231.1.2 sbi_ecall_base_probe() . . . 873
30.231.1.3 sbi_ecall_base_register_extensions() . . . 874
30.231.2 Variable Documentation . . . 874
30.231.2.1 ecall_base . . . 874
30.232 lib/sbi/sbi_ecall_cppc.c File Reference . . . 874
30.232.1 Function Documentation . . . 875
30.232.1.1 sbi_ecall_cppc_handler() . . . 875
30.232.1.2 sbi_ecall_cppc_register_extensions() . . . 875
30.232.2 Variable Documentation . . . 876
30.232.2.1 ecall_cppc . . . 876
30.233 lib/sbi/sbi_ecall_dbcn.c File Reference . . . 876
30.233.1 Function Documentation . . . 877
30.233.1.1 sbi_ecall_dbcn_handler() . . . 877
30.233.1.2 sbi_ecall_dbcn_register_extensions() . . . 877
30.233.2 Variable Documentation . . . 877
30.233.2.1 ecall_dbcn . . . 877
30.234 lib/sbi/sbi_ecall_dbtr.c File Reference . . . 877
30.234.1 Function Documentation . . . 878
30.234.1.1 sbi_ecall_dbtr_handler() . . . 878
30.234.1.2 sbi_ecall_dbtr_register_extensions() . . . 878
30.234.2 Variable Documentation . . . 879
30.234.2.1 ecall_dbtr . . . 879
30.235 lib/sbi/sbi_ecall_fwft.c File Reference . . . 879
30.235.1 Function Documentation . . . 880
30.235.1.1 sbi_ecall_fwft_handler() . . . 880
30.235.1.2 sbi_ecall_fwft_register_extensions() . . . 880
30.235.2 Variable Documentation . . . 880
30.235.2.1 ecall_fwft . . . 880
30.236 lib/sbi/sbi_ecall_hsm.c File Reference . . . 880
30.236.1 Function Documentation . . . 881
30.236.1.1 sbi_ecall_hsm_handler() . . . 881
30.236.1.2 sbi_ecall_hsm_register_extensions() . . . 882
30.236.2 Variable Documentation . . . 882
30.236.2.1 ecall_hsm . . . 882
30.237 lib/sbi/sbi_ecall_ipi.c File Reference . . . 883

30.237.1 Function Documentation . . . 883
30.237.1.1 sbi_ecall_ipi_handler() . . . 883
30.237.1.2 sbi_ecall_ipi_register_extensions() . . . 884
30.237.2 Variable Documentation . . . 884
30.237.2.1 ecall_ipi . . . 884
30.238 lib/sbi/sbi_ecall_legacy.c File Reference . . . 884
30.238.1 Function Documentation . . . 885
30.238.1.1 sbi_ecall_legacy_handler() . . . 885
30.238.1.2 sbi_ecall_legacy_register_extensions() . . . 886
30.238.1.3 sbi_load_hart_mask_unpriv() . . . 886
30.238.2 Variable Documentation . . . 886
30.238.2.1 ecall_legacy . . . 886
30.239 lib/sbi/sbi_ecall_mpxy.c File Reference . . . 886
30.239.1 Function Documentation . . . 887
30.239.1.1 sbi_ecall_mpxy_handler() . . . 887
30.239.1.2 sbi_ecall_mpxy_register_extensions() . . . 887
30.239.2 Variable Documentation . . . 888
30.239.2.1 ecall_mpxy . . . 888
30.240 lib/sbi/sbi_ecall_pmu.c File Reference . . . 888
30.240.1 Function Documentation . . . 889
30.240.1.1 sbi_ecall_pmu_handler() . . . 889
30.240.1.2 sbi_ecall_pmu_register_extensions() . . . 889
30.240.2 Variable Documentation . . . 890
30.240.2.1 ecall_pmu . . . 890
30.241 lib/sbi/sbi_ecall_rfence.c File Reference . . . 890
30.241.1 Function Documentation . . . 890
30.241.1.1 sbi_ecall_rfence_handler() . . . 890
30.241.1.2 sbi_ecall_rfence_register_extensions() . . . 891
30.241.2 Variable Documentation . . . 891
30.241.2.1 ecall_rfence . . . 891
30.242 lib/sbi/sbi_ecall_srst.c File Reference . . . 891
30.242.1 Function Documentation . . . 892
30.242.1.1 sbi_ecall_srst_handler() . . . 892
30.242.1.2 sbi_ecall_srst_register_extensions() . . . 893
30.242.1.3 srst_available() . . . 893
30.242.2 Variable Documentation . . . 894
30.242.2.1 ecall_srst . . . 894
30.243 lib/sbi/sbi_ecall_sse.c File Reference . . . 894
30.243.1 Function Documentation . . . 895
30.243.1.1 sbi_ecall_sse_handler() . . . 895
30.243.1.2 sbi_ecall_sse_register_extensions() . . . 895
30.243.2 Variable Documentation . . . 896

30.243.2.1 ecall_sse . . . 896
30.244 lib/sbi/sbi_ecall_susp.c File Reference . . . 896
30.244.1 Function Documentation . . . 896
30.244.1.1 sbi_ecall_susp_handler() . . . 896
30.244.1.2 sbi_ecall_susp_register_extensions() . . . 897
30.244.1.3 susp_available() . . . 897
30.244.2 Variable Documentation . . . 898
30.244.2.1 ecall_susp . . . 898
30.245 lib/sbi/sbi_ecall_time.c File Reference . . . 898
30.245.1 Function Documentation . . . 899
30.245.1.1 sbi_ecall_time_handler() . . . 899
30.245.1.2 sbi_ecall_time_register_extensions() . . . 899
30.245.2 Variable Documentation . . . 899
30.245.2.1 ecall_time . . . 899
30.246 lib/sbi/sbi_ecall_vendor.c File Reference . . . 899
30.246.1 Function Documentation . . . 900
30.246.1.1 sbi_ecall_vendor_handler() . . . 900
30.246.1.2 sbi_ecall_vendor_id() . . . 901
30.246.1.3 sbi_ecall_vendor_register_extensions() . . . 901
30.246.2 Variable Documentation . . . 901
30.246.2.1 ecall_vendor . . . 901
30.247 lib/sbi/sbi_emulate_csr.c File Reference . . . 902
30.247.1 Macro Definition Documentation . . . 902
30.247.1.1 switchcase_hpm . . . 902
30.247.1.2 switchcase_hpm_16 . . . 903
30.247.1.3 switchcase_hpm_2 . . . 903
30.247.1.4 switchcase_hpm_4 . . . 903
30.247.1.5 switchcase_hpm_8 . . . 903
30.247.2 Function Documentation . . . 903
30.247.2.1 hpm_allowed() . . . 903
30.247.2.2 sbi_emulate_csr_read() . . . 904
30.247.2.3 sbi_emulate_csr_write() . . . 904
30.248 lib/sbi/sbi_fifo.c File Reference . . . 905
30.248.1 Function Documentation . . . 906
30.248.1.1 __sbi_fifo_dequeue() . . . 906
30.248.1.2 __sbi_fifo_enqueue() . . . 907
30.248.1.3 __sbi_fifo_is_empty() . . . 907
30.248.1.4 __sbi_fifo_is_full() . . . 908
30.248.1.5 __sbi_fifo_reset() . . . 908
30.248.1.6 sbi_fifo_avail() . . . 909
30.248.1.7 sbi_fifo_dequeue() . . . 909
30.248.1.8 sbi_fifo_enqueue() . . . 910

30.248.1.9 sbi_fifo_init() . . . 911
30.248.1.10 sbi_fifo_inplace_update() . . . 912
30.248.1.11 sbi_fifo_is_empty() . . . 912
30.248.1.12 sbi_fifo_is_full() . . . 913
30.248.1.13 sbi_fifo_reset() . . . 913
30.249 lib/sbi/sbi_fwft.c File Reference . . . 914
30.249.1 Macro Definition Documentation . . . 915
30.249.1.1 fwft_get_hart_state_ptr . . . 915
30.249.1.2 fwft_set_hart_state_ptr . . . 915
30.249.1.3 fwft_thishart_state_ptr . . . 915
30.249.1.4 MIS_DELEG . . . 915
30.249.2 Function Documentation . . . 915
30.249.2.1 fwft_adue_supported() . . . 915
30.249.2.2 fwft_clear_config_lock() . . . 916
30.249.2.3 fwft_double_trap_supported() . . . 916
30.249.2.4 fwft_enable_lpad() . . . 916
30.249.2.5 fwft_enable_sstack() . . . 917
30.249.2.6 fwft_get_adue() . . . 917
30.249.2.7 fwft_get_double_trap() . . . 917
30.249.2.8 fwft_get_feature() . . . 918
30.249.2.9 fwft_get_lpad() . . . 918
30.249.2.10 fwft_get_misaligned_delegation() . . . 919
30.249.2.11 fwft_get_sstack() . . . 919
30.249.2.12 fwft_is_defined_feature() . . . 919
30.249.2.13 fwft_lpad_supported() . . . 919
30.249.2.14 fwft_menvcfg_read_bit() . . . 920
30.249.2.15 fwft_menvcfg_set_bit() . . . 920
30.249.2.16 fwft_misaligned_delegation_supported() . . . 921
30.249.2.17 fwft_set_adue() . . . 921
30.249.2.18 fwft_set_double_trap() . . . 921
30.249.2.19 fwft_set_misaligned_delegation() . . . 922
30.249.2.20 fwft_sstack_supported() . . . 922
30.249.2.21 get_feature_config() . . . 922
30.249.2.22 sbi_fwft_get() . . . 922
30.249.2.23 sbi_fwft_init() . . . 923
30.249.2.24 sbi_fwft_set() . . . 923
30.249.3 Variable Documentation . . . 924
30.249.3.1 features . . . 924
30.249.3.2 fwft_defined_features . . . 924
30.249.3.3 fwft_ptr_offset . . . 924
30.250 lib/sbi/sbi_hart.c File Reference . . . 924
30.250.1 Macro Definition Documentation . . . 926

30.250.1.1 __check_csr . . . 926
30.250.1.2 __check_csr_16 . . . 927
30.250.1.3 __check_csr_2 . . . 927
30.250.1.4 __check_csr_32 . . . 927
30.250.1.5 __check_csr_4 . . . 927
30.250.1.6 __check_csr_64 . . . 927
30.250.1.7 __check_csr_8 . . . 927
30.250.1.8 __check_ext_csr . . . 928
30.250.1.9 __check_hpm_csr . . . 928
30.250.1.10 __check_hpm_csr_16 . . . 928
30.250.1.11 __check_hpm_csr_2 . . . 928
30.250.1.12 __check_hpm_csr_4 . . . 928
30.250.1.13 __check_hpm_csr_8 . . . 929
30.250.1.14 __check_priv . . . 929
30.250.1.15 __SBI_HART_EXT_DATA . . . 929
30.250.1.16 __set_menvcfg_ext . . . 929
30.250.2 Function Documentation . . . 929
30.250.2.1 __attribute__() . . . 929
30.250.2.2 __sbi_expected_trap() . . . 929
30.250.2.3 __sbi_expected_trap_hext() . . . 929
30.250.2.4 __sbi_hart_update_extension() . . . 930
30.250.2.5 delegate_traps() . . . 930
30.250.2.6 fp_init() . . . 931
30.250.2.7 hart_detect_features() . . . 931
30.250.2.8 hart_mhpm_get_allowed_bits() . . . 932
30.250.2.9 hart_pmp_get_allowed_addr() . . . 932
30.250.2.10 mstatus_init() . . . 932
30.250.2.11 sbi_hart_delegation_dump() . . . 933
30.250.2.12 sbi_hart_get_extensions_str() . . . 933
30.250.2.13 sbi_hart_get_priv_version_str() . . . 934
30.250.2.14 sbi_hart_get_smepmp_flags() . . . 934
30.250.2.15 sbi_hart_has_extension() . . . 935
30.250.2.16 sbi_hart_init() . . . 936
30.250.2.17 sbi_hart_map_saddr() . . . 937
30.250.2.18 sbi_hart_mhpm_bits() . . . 937
30.250.2.19 sbi_hart_mhpm_mask() . . . 938
30.250.2.20 sbi_hart_oldpmp_configure() . . . 938
30.250.2.21 sbi_hart_pmp_addrbits() . . . 939
30.250.2.22 sbi_hart_pmp_configure() . . . 939
30.250.2.23 sbi_hart_pmp_count() . . . 940
30.250.2.24 sbi_hart_pmp_log2gran() . . . 940
30.250.2.25 sbi_hart_priv_version() . . . 941

30.250.2.26 sbi_hart_reinit() . . . 941
30.250.2.27 sbi_hart_smepmp_configure() . . . 942
30.250.2.28 sbi_hart_smepmp_set() . . . 942
30.250.2.29 sbi_hart_unmap_saddr() . . . 943
30.250.2.30 sbi_hart_update_extension() . . . 944
30.250.3 Variable Documentation . . . 944
30.250.3.1 hart_features_offset . . . 944
30.250.3.2 sbi_hart_expected_trap . . . 945
30.250.3.3 sbi_hart_ext . . . 945
30.251 lib/sbi/sbi_heap.c File Reference . . . 945
30.251.1 Macro Definition Documentation . . . 946
30.251.1.1 HEAP_ALLOC_ALIGN . . . 946
30.251.1.2 HEAP_HOUSEKEEPING_FACTOR . . . 946
30.251.2 Function Documentation . . . 946
30.251.2.1 alloc_with_align() . . . 946
30.251.2.2 sbi_aligned_alloc_from() . . . 947
30.251.2.3 sbi_free_from() . . . 948
30.251.2.4 sbi_heap_alloc_new() . . . 949
30.251.2.5 sbi_heap_free_space_from() . . . 949
30.251.2.6 sbi_heap_init() . . . 949
30.251.2.7 sbi_heap_init_new() . . . 950
30.251.2.8 sbi_heap_reserved_space_from() . . . 950
30.251.2.9 sbi_heap_used_space_from() . . . 950
30.251.2.10 sbi_malloc_from() . . . 951
30.251.2.11 sbi_zalloc_from() . . . 952
30.251.3 Variable Documentation . . . 953
30.251.3.1 global_hpctrl . . . 953
30.252 lib/sbi/sbi_hsm.c File Reference . . . 953
30.252.1 Macro Definition Documentation . . . 955
30.252.1.1 __sbi_hsm_hart_change_state . . . 955
30.252.2 Function Documentation . . . 955
30.252.2.1 __sbi_hsm_hart_get_state() . . . 955
30.252.2.2 __sbi_hsm_suspend_default() . . . 955
30.252.2.3 __sbi_hsm_suspend_non_ret_restore() . . . 956
30.252.2.4 __sbi_hsm_suspend_non_ret_save() . . . 956
30.252.2.5 hsm_device_hart_resume() . . . 956
30.252.2.6 hsm_device_hart_start() . . . 957
30.252.2.7 hsm_device_hart_stop() . . . 957
30.252.2.8 hsm_device_hart_suspend() . . . 957
30.252.2.9 hsm_device_has_hart_hotplug() . . . 958
30.252.2.10 hsm_device_has_hart_secondary_boot() . . . 958
30.252.2.11 hsm_start_ticket_acquire() . . . 958

30.252.2.12 hsm_start_ticket_release() . . . 959
30.252.2.13 sbi_hsm_exit() . . . 959
30.252.2.14 sbi_hsm_get_device() . . . 960
30.252.2.15 sbi_hsm_hart_change_state() . . . 960
30.252.2.16 sbi_hsm_hart_get_state() . . . 960
30.252.2.17 sbi_hsm_hart_interruptible_mask() . . . 961
30.252.2.18 sbi_hsm_hart_resume_finish() . . . 961
30.252.2.19 sbi_hsm_hart_resume_start() . . . 962
30.252.2.20 sbi_hsm_hart_start() . . . 962
30.252.2.21 sbi_hsm_hart_start_finish() . . . 963
30.252.2.22 sbi_hsm_hart_stop() . . . 964
30.252.2.23 sbi_hsm_hart_suspend() . . . 964
30.252.2.24 sbi_hsm_hart_wait() . . . 965
30.252.2.25 sbi_hsm_init() . . . 966
30.252.2.26 sbi_hsm_set_device() . . . 966
30.252.3 Variable Documentation . . . 966
30.252.3.1 hart_data_offset . . . 966
30.252.3.2 hsm_dev . . . 967
30.253 lib/sbi/sbi_illegal_insn.c File Reference . . . 967
30.253.1 Typedef Documentation . . . 967
30.253.1.1 illegal_insn_func . . . 967
30.253.2 Function Documentation . . . 968
30.253.2.1 misc_mem_opcode_insn() . . . 968
30.253.2.2 sbi_illegal_insn_handler() . . . 968
30.253.2.3 system_opcode_insn() . . . 968
30.253.2.4 truly_illegal_insn() . . . 969
30.253.3 Variable Documentation . . . 969
30.253.3.1 illegal_insn_table . . . 969
30.254 lib/sbi/sbi_init.c File Reference . . . 969
30.254.1 Macro Definition Documentation . . . 971
30.254.1.1 BANNER . . . 971
30.254.2 Function Documentation . . . 971
30.254.2.1 init_coldboot() . . . 971
30.254.2.2 init_warm_resume() . . . 973
30.254.2.3 init_warm_startup() . . . 973
30.254.2.4 init_warmboot() . . . 975
30.254.2.5 sbi_boot_print_banner() . . . 976
30.254.2.6 sbi_boot_print_domains() . . . 976
30.254.2.7 sbi_boot_print_general() . . . 977
30.254.2.8 sbi_boot_print_hart() . . . 977
30.254.2.9 sbi_entry_count() . . . 978
30.254.2.10 sbi_exit() . . . 978

30.254.2.11 sbi_init() . . . 979
30.254.2.12 sbi_init_count() . . . 980
30.254.2.13 wait_for_coldboot() . . . 981
30.254.2.14 wake_coldboot_harts() . . . 981
30.254.3 Variable Documentation . . . 981
30.254.3.1 coldboot_done . . . 981
30.254.3.2 coldboot_lottery . . . 981
30.254.3.3 entry_count_offset . . . 981
30.254.3.4 init_count_offset . . . 981
30.255 lib/sbi/sbi_ipi.c File Reference . . . 981
30.255.1 Function Documentation . . . 983
30.255.1.1 sbi_ipi_clear_smode() . . . 983
30.255.1.2 sbi_ipi_event_create() . . . 983
30.255.1.3 sbi_ipi_event_destroy() . . . 983
30.255.1.4 sbi_ipi_exit() . . . 983
30.255.1.5 sbi_ipi_get_device() . . . 984
30.255.1.6 sbi_ipi_init() . . . 984
30.255.1.7 sbi_ipi_process() . . . 985
30.255.1.8 sbi_ipi_process_halt() . . . 985
30.255.1.9 sbi_ipi_process_smode() . . . 986
30.255.1.10 sbi_ipi_raw_clear() . . . 986
30.255.1.11 sbi_ipi_raw_send() . . . 986
30.255.1.12 sbi_ipi_send() . . . 986
30.255.1.13 sbi_ipi_send_halt() . . . 987
30.255.1.14 sbi_ipi_send_many() . . . 988
30.255.1.15 sbi_ipi_send_smode() . . . 988
30.255.1.16 sbi_ipi_set_device() . . . 989
30.255.1.17 sbi_ipi_sync() . . . 989
30.255.1.18 sbi_ipi_update_halt() . . . 990
30.255.2 Variable Documentation . . . 990
30.255.2.1 ipi_data_off . . . 990
30.255.2.2 ipi_dev . . . 990
30.255.2.3 ipi_halt_event . . . 990
30.255.2.4 ipi_halt_ops . . . 990
30.255.2.5 ipi_ops_array . . . 990
30.255.2.6 ipi_smode_event . . . 990
30.255.2.7 ipi_smode_ops . . . 990
30.256 lib/sbi/sbi_irqchip.c File Reference . . . 991
30.256.1 Function Documentation . . . 991
30.256.1.1 default_irqfn() . . . 991
30.256.1.2 sbi_irqchip_add_device() . . . 992
30.256.1.3 sbi_irqchip_exit() . . . 992

30.256.1.4 sbi_irqchip_init() . . . 993
30.256.1.5 SBI_LIST_HEAD() . . . 993
30.256.2 Variable Documentation . . . 993
30.256.2.1 ext_irqfn . . . 993
30.257 lib/sbi/sbi_math.c File Reference . . . 994
30.257.1 Function Documentation . . . 994
30.257.1.1 log2roundup() . . . 994
30.258 lib/sbi/sbi_mpxy.c File Reference . . . 994
30.258.1 Macro Definition Documentation . . . 996
30.258.1.1 assert_field_offset . . . 996
30.258.1.2 ATTR_SIZE . . . 996
30.258.1.3 CAP_DISABLE . . . 996
30.258.1.4 CAP_ENABLE . . . 996
30.258.1.5 CAP_EVENTSSTATE_MASK . . . 996
30.258.1.6 CAP_EVENTSSTATE_POS . . . 997
30.258.1.7 CAP_GET . . . 997
30.258.1.8 CAP_GET_NOTIFICATIONS_MASK . . . 997
30.258.1.9 CAP_GET_NOTIFICATIONS_POS . . . 997
30.258.1.10 CAP_MSI_MASK . . . 997
30.258.1.11 CAP_MSI_POS . . . 997
30.258.1.12 CAP_SEND_MSG_WITH_RESP_MASK . . . 997
30.258.1.13 CAP_SEND_MSG_WITH_RESP_POS . . . 997
30.258.1.14 CAP_SEND_MSG_WITHOUT_RESP_MASK . . . 997
30.258.1.15 CAP_SEND_MSG_WITHOUT_RESP_POS . . . 997
30.258.1.16 CAP_SSE_MASK . . . 997
30.258.1.17 CAP_SSE_POS . . . 997
30.258.1.18 INVALID_ADDR . . . 997
30.258.2 Function Documentation . . . 998
30.258.2.1 assert_field_offset() [1/12] . . . 998
30.258.2.2 assert_field_offset() [2/12] . . . 998
30.258.2.3 assert_field_offset() [3/12] . . . 998
30.258.2.4 assert_field_offset() [4/12] . . . 998
30.258.2.5 assert_field_offset() [5/12] . . . 998
30.258.2.6 assert_field_offset() [6/12] . . . 998
30.258.2.7 assert_field_offset() [7/12] . . . 998
30.258.2.8 assert_field_offset() [8/12] . . . 998
30.258.2.9 assert_field_offset() [9/12] . . . 998
30.258.2.10 assert_field_offset() [10/12] . . . 998
30.258.2.11 assert_field_offset() [11/12] . . . 999
30.258.2.12 assert_field_offset() [12/12] . . . 999
30.258.2.13 hart_shmem_base() . . . 999
30.258.2.14 mpxy_check_write_std_attr() . . . 999

30.258.2.15 mpxy_copy_std_attrs() . . . 999
30.258.2.16 mpxy_find_channel() . . . 1000
30.258.2.17 mpxy_is_std_attr() . . . 1000
30.258.2.18 mpxy_shmem_enabled() . . . 1000
30.258.2.19 mpxy_std_attrs_init() . . . 1001
30.258.2.20 mpxy_write_std_attr() . . . 1001
30.258.2.21 SBI_LIST_HEAD() . . . 1001
30.258.2.22 sbi_mpxy_channel_available() . . . 1002
30.258.2.23 sbi_mpxy_get_channel_ids() . . . 1002
30.258.2.24 sbi_mpxy_get_notification_events() . . . 1002
30.258.2.25 sbi_mpxy_init() . . . 1003
30.258.2.26 sbi_mpxy_read_attrs() . . . 1004
30.258.2.27 sbi_mpxy_register_channel() . . . 1004
30.258.2.28 sbi_mpxy_send_message() . . . 1005
30.258.2.29 sbi_mpxy_set_shmem() . . . 1005
30.258.2.30 sbi_mpxy_shmem_disable() . . . 1006
30.258.2.31 sbi_mpxy_write_attrs() . . . 1006
30.258.3 Variable Documentation . . . 1007
30.258.3.1 mpxy_state_offset . . . 1007
30.259 lib/sbi/sbi_platform.c File Reference . . . 1007
30.259.1 Function Documentation . . . 1008
30.259.1.1 sbi_platform_feature_id2string() . . . 1008
30.259.1.2 sbi_platform_get_features_str() . . . 1008
30.260 lib/sbi/sbi_pmu.c File Reference . . . 1009
30.260.1 Macro Definition Documentation . . . 1011
30.260.1.1 get_cidx_code . . . 1011
30.260.1.2 get_cidx_type . . . 1011
30.260.1.3 pmu_get_hart_state_ptr . . . 1011
30.260.1.4 pmu_set_hart_state_ptr . . . 1011
30.260.1.5 pmu_thishart_state_ptr . . . 1011
30.260.2 Function Documentation . . . 1011
30.260.2.1 pmu_add_hw_event_map() . . . 1011
30.260.2.2 pmu_ctr_enable_irq_hw() . . . 1012
30.260.2.3 pmu_ctr_find_fixed_hw() . . . 1012
30.260.2.4 pmu_ctr_find_fw() . . . 1012
30.260.2.5 pmu_ctr_find_hw() . . . 1013
30.260.2.6 pmu_ctr_start_fw() . . . 1013
30.260.2.7 pmu_ctr_start_hw() . . . 1013
30.260.2.8 pmu_ctr_stop_fw() . . . 1014
30.260.2.9 pmu_ctr_stop_hw() . . . 1014
30.260.2.10 pmu_ctr_validate() . . . 1015
30.260.2.11 pmu_ctr_write_hw() . . . 1015

30.260.2.12 pmu_event_range_overlap() . . . 1016
30.260.2.13 pmu_event_select_overlap() . . . 1016
30.260.2.14 pmu_event_validate() . . . 1017
30.260.2.15 pmu_fixed_ctr_update_inhibit_bits() . . . 1017
30.260.2.16 pmu_reset_event_map() . . . 1017
30.260.2.17 pmu_reset_hw_mhpmevent() . . . 1018
30.260.2.18 pmu_sse_complete() . . . 1018
30.260.2.19 pmu_sse_disable() . . . 1018
30.260.2.20 pmu_sse_enable() . . . 1019
30.260.2.21 pmu_update_hw_mhpmevent() . . . 1019
30.260.2.22 pmu_update_inhibit_flags() . . . 1019
30.260.2.23 sbi_pmu_add_hw_event_counter_map() . . . 1020
30.260.2.24 sbi_pmu_add_raw_event_counter_map() . . . 1020
30.260.2.25 sbi_pmu_ctr_cfg_match() . . . 1021
30.260.2.26 sbi_pmu_ctr_fw_read() . . . 1021
30.260.2.27 sbi_pmu_ctr_get_info() . . . 1022
30.260.2.28 sbi_pmu_ctr_incr_fw() . . . 1023
30.260.2.29 sbi_pmu_ctr_start() . . . 1023
30.260.2.30 sbi_pmu_ctr_stop() . . . 1023
30.260.2.31 sbi_pmu_event_get_info() . . . 1024
30.260.2.32 sbi_pmu_exit() . . . 1024
30.260.2.33 sbi_pmu_get_device() . . . 1025
30.260.2.34 sbi_pmu_init() . . . 1025
30.260.2.35 sbi_pmu_irq_bit() . . . 1026
30.260.2.36 sbi_pmu_num_ctr() . . . 1026
30.260.2.37 sbi_pmu_ovf_irq() . . . 1027
30.260.2.38 sbi_pmu_set_device() . . . 1027
30.260.3 Variable Documentation . . . 1027
30.260.3.1 hw_event_map . . . 1027
30.260.3.2 num_hw_ctrs . . . 1027
30.260.3.3 num_hw_events . . . 1028
30.260.3.4 phs_ptr_offset . . . 1028
30.260.3.5 pmu_dev . . . 1028
30.260.3.6 pmu_sse_cb_ops . . . 1028
30.260.3.7 total_ctrs . . . 1028
30.261 lib/sbi/sbi_scratch.c File Reference . . . 1028
30.261.1 Typedef Documentation . . . 1029
30.261.1.1 hartid2scratch . . . 1029
30.261.2 Function Documentation . . . 1029
30.261.2.1 sbi_hartid_to_hartindex() . . . 1029
30.261.2.2 sbi_scratch_alloc_offset() . . . 1029
30.261.2.3 sbi_scratch_free_offset() . . . 1030

30.261.2.4 sbi_scratch_init() . . . 1031
30.261.2.5 sbi_scratch_used_space() . . . 1031
30.261.3 Variable Documentation . . . 1031
30.261.3.1 extra_lock . . . 1031
30.261.3.2 extra_offset . . . 1031
30.261.3.3 hartindex_to_hartid_table . . . 1031
30.261.3.4 hartindex_to_scratch_table . . . 1032
30.261.3.5 last_hartindex_having_scratch . . . 1032
30.262 lib/sbi/sbi_sse.c File Reference . . . 1032
30.262.1 Macro Definition Documentation . . . 1034
30.262.1.1 assert_field_offset . . . 1034
30.262.1.2 EVENT_COUNT . . . 1034
30.262.1.3 EVENT_IS_GLOBAL . . . 1034
30.262.1.4 sse_event_invoke_cb . . . 1034
30.262.1.5 sse_get_hart_state_ptr . . . 1035
30.262.1.6 sse_set_hart_state_ptr . . . 1035
30.262.1.7 sse_thishart_state_ptr . . . 1035
30.262.2 Function Documentation . . . 1035
30.262.2.1 assert_field_offset() [1/10] . . . 1035
30.262.2.2 assert_field_offset() [2/10] . . . 1035
30.262.2.3 assert_field_offset() [3/10] . . . 1035
30.262.2.4 assert_field_offset() [4/10] . . . 1035
30.262.2.5 assert_field_offset() [5/10] . . . 1035
30.262.2.6 assert_field_offset() [6/10] . . . 1035
30.262.2.7 assert_field_offset() [7/10] . . . 1035
30.262.2.8 assert_field_offset() [8/10] . . . 1036
30.262.2.9 assert_field_offset() [9/10] . . . 1036
30.262.2.10 assert_field_offset() [10/10] . . . 1036
30.262.2.11 copy_attrs() . . . 1036
30.262.2.12 sbi_sse_attr_check() . . . 1036
30.262.2.13 sbi_sse_complete() . . . 1037
30.262.2.14 sbi_sse_disable() . . . 1037
30.262.2.15 sbi_sse_enable() . . . 1038
30.262.2.16 sbi_sse_exit() . . . 1039
30.262.2.17 sbi_sse_hart_mask() . . . 1039
30.262.2.18 sbi_sse_hart_unmask() . . . 1040
30.262.2.19 sbi_sse_init() . . . 1040
30.262.2.20 sbi_sse_inject_event() . . . 1041
30.262.2.21 sbi_sse_inject_from_ecall() . . . 1041
30.262.2.22 sbi_sse_process_pending_events() . . . 1042
30.262.2.23 sbi_sse_read_attrs() . . . 1043
30.262.2.24 sbi_sse_register() . . . 1043

30.262.2.25 sbi_sse_set_cb_ops() . . . 1044
30.262.2.26 sbi_sse_unregister() . . . 1045
30.262.2.27 sbi_sse_write_attrs() . . . 1045
30.262.2.28 sse_enabled_event_lock() . . . 1046
30.262.2.29 sse_enabled_event_unlock() . . . 1047
30.262.2.30 sse_event_add_to_list() . . . 1047
30.262.2.31 sse_event_check_inject() . . . 1048
30.262.2.32 sse_event_complete() . . . 1048
30.262.2.33 sse_event_count_init() . . . 1049
30.262.2.34 sse_event_disable() . . . 1049
30.262.2.35 sse_event_enable() . . . 1049
30.262.2.36 sse_event_get() . . . 1050
30.262.2.37 sse_event_init() . . . 1051
30.262.2.38 sse_event_inject() . . . 1051
30.262.2.39 sse_event_is_global() . . . 1052
30.262.2.40 sse_event_is_local() . . . 1052
30.262.2.41 sse_event_is_ready() . . . 1053
30.262.2.42 sse_event_pending() . . . 1053
30.262.2.43 sse_event_put() . . . 1054
30.262.2.44 sse_event_register() . . . 1054
30.262.2.45 sse_event_remove_from_list() . . . 1055
30.262.2.46 sse_event_resume() . . . 1055
30.262.2.47 sse_event_set_attr() . . . 1056
30.262.2.48 sse_event_set_attr_check() . . . 1056
30.262.2.49 sse_event_set_hart_id_check() . . . 1057
30.262.2.50 sse_event_set_pending() . . . 1057
30.262.2.51 sse_event_set_state() . . . 1058
30.262.2.52 sse_event_state() . . . 1058
30.262.2.53 sse_event_unregister() . . . 1059
30.262.2.54 sse_get_global_event() . . . 1059
30.262.2.55 sse_get_hart_state() . . . 1059
30.262.2.56 sse_global_init() . . . 1060
30.262.2.57 sse_inject_event() . . . 1060
30.262.2.58 sse_interrupted_flags() . . . 1061
30.262.2.59 sse_ipi_inject_process() . . . 1061
30.262.2.60 sse_ipi_inject_send() . . . 1061
30.262.2.61 sse_local_init() . . . 1062
30.262.2.62 sse_write_attrs() . . . 1063
30.262.3 Variable Documentation . . . 1063
30.262.3.1 global_event_count . . . 1063
30.262.3.2 global_events . . . 1063
30.262.3.3 local_event_count . . . 1063

30.262.3.4 shs_ptr_off . . . 1063
30.262.3.5 sse_inject_fifo_mem_off . . . 1063
30.262.3.6 sse_inject_fifo_off . . . 1063
30.262.3.7 sse_ipi_inject_event . . . 1063
30.262.3.8 sse_ipi_inject_ops . . . 1064
30.262.3.9 supported_events . . . 1064
30.263 lib/sbi/sbi_string.c File Reference . . . 1064
30.263.1 Function Documentation . . . 1065
30.263.1.1 sbi_memchr() . . . 1065
30.263.1.2 sbi_memcmp() . . . 1065
30.263.1.3 sbi_memcpy() . . . 1065
30.263.1.4 sbi_memmove() . . . 1066
30.263.1.5 sbi_memset() . . . 1066
30.263.1.6 sbi_strchr() . . . 1066
30.263.1.7 sbi_strcmp() . . . 1066
30.263.1.8 sbi_strcpy() . . . 1067
30.263.1.9 sbi_strlen() . . . 1067
30.263.1.10 sbi_strncmp() . . . 1068
30.263.1.11 sbi_strncpy() . . . 1068
30.263.1.12 sbi_strnlen() . . . 1068
30.263.1.13 sbi_strrchr() . . . 1068
30.264 lib/sbi/sbi_system.c File Reference . . . 1068
30.264.1 Function Documentation . . . 1069
30.264.1.1 SBI_LIST_HEAD() . . . 1069
30.264.1.2 sbi_system_reset() . . . 1069
30.264.1.3 sbi_system_reset_add_device() . . . 1070
30.264.1.4 sbi_system_reset_get_device() . . . 1071
30.264.1.5 sbi_system_reset_supported() . . . 1071
30.264.1.6 sbi_system_suspend() . . . 1072
30.264.1.7 sbi_system_suspend_get_device() . . . 1073
30.264.1.8 sbi_system_suspend_set_device() . . . 1073
30.264.1.9 sbi_system_suspend_supported() . . . 1073
30.264.1.10 sbi_system_suspend_test_check() . . . 1074
30.264.1.11 sbi_system_suspend_test_enable() . . . 1074
30.264.1.12 sbi_system_suspend_test_suspend() . . . 1074
30.264.2 Variable Documentation . . . 1074
30.264.2.1 sbi_system_suspend_test . . . 1074
30.264.2.2 suspend_dev . . . 1075
30.265 lib/sbi/sbi_timer.c File Reference . . . 1075
30.265.1 Function Documentation . . . 1076
30.265.1.1 get_ticks() . . . 1076
30.265.1.2 nop_delay_fn() . . . 1076

30.265.1.3 sbi_timer_delay_loop() . . . 1076
30.265.1.4 sbi_timer_event_start() . . . 1077
30.265.1.5 sbi_timer_exit() . . . 1077
30.265.1.6 sbi_timer_get_delta() . . . 1078
30.265.1.7 sbi_timer_get_device() . . . 1078
30.265.1.8 sbi_timer_init() . . . 1078
30.265.1.9 sbi_timer_process() . . . 1079
30.265.1.10 sbi_timer_set_delta() . . . 1080
30.265.1.11 sbi_timer_set_device() . . . 1080
30.265.1.12 sbi_timer_value() . . . 1080
30.265.1.13 sbi_timer_virt_value() . . . 1080
30.265.1.14 sbi_timer_waitms_until() . . . 1081
30.265.2 Variable Documentation . . . 1082
30.265.2.1 get_time_val . . . 1082
30.265.2.2 time_delta_off . . . 1082
30.265.2.3 timer_dev . . . 1082
30.266 lib/sbi/sbi_tlb.c File Reference . . . 1082
30.266.1 Function Documentation . . . 1083
30.266.1.1 sbi_tlb_init() . . . 1083
30.266.1.2 sbi_tlb_local_fence_i() . . . 1084
30.266.1.3 sbi_tlb_local_hfence_gvma() . . . 1084
30.266.1.4 sbi_tlb_local_hfence_gvma_vmid() . . . 1085
30.266.1.5 sbi_tlb_local_hfence_vvma() . . . 1085
30.266.1.6 sbi_tlb_local_hfence_vvma_asid() . . . 1086
30.266.1.7 sbi_tlb_local_sfence_vma() . . . 1087
30.266.1.8 sbi_tlb_local_sfence_vma_asid() . . . 1087
30.266.1.9 sbi_tlb_request() . . . 1087
30.266.1.10 tlb_entry_local_process() . . . 1088
30.266.1.11 tlb_entry_process() . . . 1089
30.266.1.12 tlb_flush_all() . . . 1090
30.266.1.13 tlb_process() . . . 1090
30.266.1.14 tlb_process_once() . . . 1091
30.266.1.15 tlb_range_check() . . . 1091
30.266.1.16 tlb_sync() . . . 1092
30.266.1.17 tlb_update() . . . 1093
30.266.1.18 tlb_update_cb() . . . 1093
30.266.2 Variable Documentation . . . 1094
30.266.2.1 tlb_event . . . 1094
30.266.2.2 tlb_fifo_mem_off . . . 1094
30.266.2.3 tlb_fifo_off . . . 1094
30.266.2.4 tlb_ops . . . 1094
30.266.2.5 tlb_range_flush_limit . . . 1094

30.266.2.6 tlb_sync_off . . . 1094
30.266.2.7 tlb_type_to_pmu_fw_event . . . 1094
30.267 lib/sbi/sbi_trap.c File Reference . . . 1094
30.267.1 Function Documentation . . . 1095
30.267.1.1 sbi_trap_aia_irq() . . . 1095
30.267.1.2 sbi_trap_error() . . . 1096
30.267.1.3 sbi_trap_error_one() . . . 1096
30.267.1.4 sbi_trap_handler() . . . 1097
30.267.1.5 sbi_trap_nonaia_irq() . . . 1097
30.267.1.6 sbi_trap_redirect() . . . 1098
30.268 lib/sbi/sbi_trap_ldst.c File Reference . . . 1099
30.268.1 Typedef Documentation . . . 1100
30.268.1.1 sbi_trap_ld_emulator . . . 1100
30.268.1.2 sbi_trap_st_emulator . . . 1101
30.268.2 Function Documentation . . . 1101
30.268.2.1 sbi_ld_access_emulator() . . . 1101
30.268.2.2 sbi_load_access_handler() . . . 1101
30.268.2.3 sbi_misaligned_ld_emulator() . . . 1102
30.268.2.4 sbi_misaligned_load_handler() . . . 1102
30.268.2.5 sbi_misaligned_st_emulator() . . . 1103
30.268.2.6 sbi_misaligned_store_handler() . . . 1103
30.268.2.7 sbi_misaligned_tinst_fixup() . . . 1104
30.268.2.8 sbi_st_access_emulator() . . . 1104
30.268.2.9 sbi_store_access_handler() . . . 1105
30.268.2.10 sbi_trap_emulate_load() . . . 1105
30.268.2.11 sbi_trap_emulate_store() . . . 1106
30.269 lib/sbi/sbi_trap_v_ldst.c File Reference . . . 1106
30.269.1 Function Documentation . . . 1107
30.269.1.1 sbi_misaligned_v_ld_emulator() . . . 1107
30.269.1.2 sbi_misaligned_v_st_emulator() . . . 1107
30.270 lib/sbi/sbi_unpriv.c File Reference . . . 1108
30.270.1 Macro Definition Documentation . . . 1108
30.270.1.1 DEFINE_UNPRIVILEGED_LOAD_FUNCTION . . . 1108
30.270.1.2 DEFINE_UNPRIVILEGED_STORE_FUNCTION . . . 1109
30.270.2 Function Documentation . . . 1109
30.270.2.1 sbi_get_insn() . . . 1109
30.271 lib/sbi/tests/riscv_atomic_test.c File Reference . . . 1110
30.271.1 Macro Definition Documentation . . . 1111
30.271.1.1 ATOMIC_TEST_BIT_NUM . . . 1111
30.271.1.2 ATOMIC_TEST_RAW_BIT_CELL . . . 1111
30.271.1.3 ATOMIC_TEST_RAW_BIT_NUM . . . 1111
30.271.1.4 ATOMIC_TEST_VAL1 . . . 1111

30.271.1.5 ATOMIC_TEST_VAL2 . . . 1111
30.271.1.6 ATOMIC_TEST_VAL3 . . . 1111
30.271.2 Function Documentation . . . 1111
30.271.2.1 add_return_test() . . . 1111
30.271.2.2 atomic_clear_bit_test() . . . 1112
30.271.2.3 atomic_raw_clear_bit_test() . . . 1112
30.271.2.4 atomic_raw_set_bit_test() . . . 1112
30.271.2.5 atomic_rw_test() . . . 1113
30.271.2.6 atomic_set_bit_test() . . . 1113
30.271.2.7 atomic_test_suite_init() . . . 1113
30.271.2.8 atomic_xchg_test() . . . 1114
30.271.2.9 cmpxchg_test() . . . 1114
30.271.2.10 sub_return_test() . . . 1114
30.271.3 Variable Documentation . . . 1115
30.271.3.1 atomic_test_cases . . . 1115
30.271.3.2 atomic_test_suite . . . 1115
30.271.3.3 test_atomic . . . 1115
30.272 lib/sbi/tests/riscv_locks_test.c File Reference . . . 1115
30.272.1 Function Documentation . . . 1116
30.272.1.1 SBIUNIT_TEST_SUITE() . . . 1116
30.272.1.2 spin_lock_test() . . . 1116
30.272.1.3 spin_trylock_fail() . . . 1117
30.272.1.4 spin_trylock_success() . . . 1117
30.272.2 Variable Documentation . . . 1118
30.272.2.1 locks_test_cases . . . 1118
30.272.2.2 test_lock . . . 1118
30.273 lib/sbi/tests/sbi_bitmap_test.c File Reference . . . 1118
30.273.1 Macro Definition Documentation . . . 1119
30.273.1.1 DATA_BIT_SIZE . . . 1119
30.273.1.2 DATA_SIZE . . . 1119
30.273.2 Function Documentation . . . 1119
30.273.2.1 bitmap_and_test() . . . 1119
30.273.2.2 bitmap_or_test() . . . 1119
30.273.2.3 bitmap_xor_test() . . . 1120
30.273.2.4 SBIUNIT_TEST_SUITE() . . . 1120
30.273.3 Variable Documentation . . . 1120
30.273.3.1 bitmap_test_cases . . . 1120
30.273.3.2 data_a . . . 1120
30.273.3.3 data_b . . . 1120
30.273.3.4 data_zero . . . 1120
30.274 lib/sbi/tests/sbi_console_test.c File Reference . . . 1120
30.274.1 Macro Definition Documentation . . . 1121

30.274.1.1 PRINTF_TEST . . . 1121
30.274.1.2 PUTS_TEST . . . 1122
30.274.1.3 TEST_CONSOLE_BUF_LEN . . . 1122
30.274.2 Function Documentation . . . 1122
30.274.2.1 clear_test_console_buf() . . . 1122
30.274.2.2 printf_test() . . . 1122
30.274.2.3 putc_test() . . . 1122
30.274.2.4 puts_test() . . . 1123
30.274.2.5 SBIUNIT_TEST_SUITE() . . . 1123
30.274.2.6 test_console_begin() . . . 1123
30.274.2.7 test_console_end() . . . 1123
30.274.2.8 test_console_putc() . . . 1124
30.274.3 Variable Documentation . . . 1124
30.274.3.1 console_test_cases . . . 1124
30.274.3.2 old_dev . . . 1124
30.274.3.3 test_console_buf . . . 1124
30.274.3.4 test_console_buf_pos . . . 1124
30.274.3.5 test_console_dev . . . 1124
30.274.3.6 test_console_lock . . . 1125
30.275 lib/sbi/tests/sbi_math_test.c File Reference . . . 1125
30.275.1 Function Documentation . . . 1125
30.275.1.1 log2roundup_test() . . . 1125
30.275.1.2 SBIUNIT_TEST_SUITE() . . . 1125
30.275.2 Variable Documentation . . . 1126
30.275.2.1 math_test_cases . . . 1126
30.276 lib/sbi/tests/sbi_unit_test.c File Reference . . . 1126
30.276.1 Macro Definition Documentation . . . 1126
30.276.1.1 ANSI_COLOR_GREEN . . . 1126
30.276.1.2 ANSI_COLOR_RED . . . 1127
30.276.1.3 ANSI_COLOR_RESET . . . 1127
30.276.2 Function Documentation . . . 1127
30.276.2.1 run_all_tests() . . . 1127
30.276.2.2 run_test_suite() . . . 1127
30.276.3 Variable Documentation . . . 1127
30.276.3.1 sbi_unit_tests . . . 1127
30.277 lib/utils/cppc/fdt_cppc.c File Reference . . . 1128
30.277.1 Function Documentation . . . 1128
30.277.1.1 fdt_cppc_init() . . . 1128
30.277.2 Variable Documentation . . . 1129
30.277.2.1 fdt_cppc_drivers . . . 1129
30.278 lib/utils/cppc/fdt_cppc_rpmi.c File Reference . . . 1129
30.278.1 Macro Definition Documentation . . . 1130

30.278.1.1 RPMI_CPPC_HART_FASTCHAN_SIZE . . . 1130
30.278.2 Function Documentation . . . 1130
30.278.2.1 rpmi_cppc_cold_init() . . . 1130
30.278.2.2 rpmi_cppc_fc_db_trigger() . . . 1130
30.278.2.3 rpmi_cppc_get_pointer() . . . 1130
30.278.2.4 rpmi_cppc_probe() . . . 1131
30.278.2.5 rpmi_cppc_read() . . . 1131
30.278.2.6 rpmi_cppc_update_hart_scratch() . . . 1131
30.278.2.7 rpmi_cppc_write() . . . 1132
30.278.3 Variable Documentation . . . 1132
30.278.3.1 fdt_cppc_rpmi . . . 1132
30.278.3.2 rpmi_cppc_match . . . 1132
30.278.3.3 rpmi_cppc_offset . . . 1133
30.278.3.4 sbi_rpmi_cppc . . . 1133
30.279 lib/utils/fdt/fdt_domain.c File Reference . . . 1133
30.279.1 Macro Definition Documentation . . . 1134
30.279.1.1 DISABLE_DEVICES_MASK . . . 1134
30.279.1.2 FDT_DOMAIN_REGION_MAX_COUNT . . . 1134
30.279.2 Function Documentation . . . 1134
30.279.2.1 __fdt_parse_domain() . . . 1134
30.279.2.2 __fdt_parse_region() . . . 1135
30.279.2.3 __fixup_count_disable_devices() . . . 1135
30.279.2.4 __fixup_disable_devices() . . . 1136
30.279.2.5 __fixup_find_domain_offset() . . . 1137
30.279.2.6 fdt_domain_fixup() . . . 1137
30.279.2.7 fdt_domains_populate() . . . 1137
30.279.2.8 fdt_iterate_each_domain() . . . 1138
30.279.2.9 fdt_iterate_each_domain_ro() . . . 1138
30.279.2.10 fdt_iterate_each_memregion() . . . 1139
30.279.2.11 fdt_iterate_each_memregion_ro() . . . 1139
30.280 lib/utils/fdt/fdt_driver.c File Reference . . . 1140
30.280.1 Function Documentation . . . 1140
30.280.1.1 fdt_driver_init_all() . . . 1140
30.280.1.2 fdt_driver_init_by_offset() . . . 1142
30.280.1.3 fdt_driver_init_one() . . . 1143
30.280.1.4 fdt_driver_init_scan() . . . 1144
30.281 lib/utils/fdt/fdt_fixup.c File Reference . . . 1145
30.281.1 Function Documentation . . . 1146
30.281.1.1 fdt_add_cpu_idle_states() . . . 1146
30.281.1.2 fdt_aplic_fixup() . . . 1147
30.281.1.3 fdt_config_fixup() . . . 1147
30.281.1.4 fdt_cpu_fixup() . . . 1148

30.281.1.5 fdt_domain_based_fixup_one() . . . 1148
30.281.1.6 fdt_fixup_node() . . . 1149
30.281.1.7 fdt_fixups() . . . 1150
30.281.1.8 fdt_imsic_fixup() . . . 1150
30.281.1.9 fdt_plic_fixup() . . . 1151
30.281.1.10 fdt_register_general_fixup() . . . 1151
30.281.1.11 fdt_reserved_memory_fixup() . . . 1152
30.281.1.12 fdt_resv_memory_update_node() . . . 1153
30.281.1.13 fdt_unregister_general_fixup() . . . 1153
30.281.1.14 SBI_LIST_HEAD() . . . 1154
30.282 lib/utils/fdt/fdt_helper.c File Reference . . . 1154
30.282.1 Macro Definition Documentation . . . 1155
30.282.1.1 DEFAULT_RENESAS_SCIF_BAUD . . . 1155
30.282.1.2 DEFAULT_RENESAS_SCIF_FREQ . . . 1156
30.282.1.3 DEFAULT_SHAKTI_UART_BAUD . . . 1156
30.282.1.4 DEFAULT_SHAKTI_UART_FREQ . . . 1156
30.282.1.5 DEFAULT_SIFIVE_UART_BAUD . . . 1156
30.282.1.6 DEFAULT_SIFIVE_UART_FREQ . . . 1156
30.282.1.7 DEFAULT_UART_BAUD . . . 1156
30.282.1.8 DEFAULT_UART_FREQ . . . 1156
30.282.1.9 DEFAULT_UART_REG_IO_WIDTH . . . 1156
30.282.1.10 DEFAULT_UART_REG_OFFSET . . . 1156
30.282.1.11 DEFAULT_UART_REG_SHIFT . . . 1156
30.282.1.12 RISCV_ISA_EXT_NAME_LEN_MAX . . . 1156
30.282.1.13 set_multi_letter_ext . . . 1156
30.282.2 Function Documentation . . . 1156
30.282.2.1 fdt_check_imsic_mlevel() . . . 1156
30.282.2.2 fdt_find_match() . . . 1157
30.282.2.3 fdt_get_aclint_addr_size() . . . 1157
30.282.2.4 fdt_get_aclint_addr_size_by_name() . . . 1158
30.282.2.5 fdt_get_node_addr_size() . . . 1159
30.282.2.6 fdt_get_node_addr_size_by_name() . . . 1160
30.282.2.7 fdt_match_node() . . . 1160
30.282.2.8 fdt_node_is_enabled() . . . 1160
30.282.2.9 fdt_parse_aclint_node() . . . 1161
30.282.2.10 fdt_parse_aplic_node() . . . 1162
30.282.2.11 fdt_parse_compat_addr() . . . 1162
30.282.2.12 fdt_parse_gaisler_uart_node() . . . 1163
30.282.2.13 fdt_parse_hart_id() . . . 1164
30.282.2.14 fdt_parse_imsic_node() . . . 1164
30.282.2.15 fdt_parse_isa_all_harts() . . . 1165
30.282.2.16 fdt_parse_isa_extensions() . . . 1165

30.282.2.17 fdt_parse_isa_extensions_one_hart() . . . 1166
30.282.2.18 fdt_parse_isa_one_hart() . . . 1166
30.282.2.19 fdt_parse_max_enabled_hart_id() . . . 1167
30.282.2.20 fdt_parse_phandle_with_args() . . . 1167
30.282.2.21 fdt_parse_plic() . . . 1167
30.282.2.22 fdt_parse_plic_node() . . . 1168
30.282.2.23 fdt_parse_plicsw_node() . . . 1168
30.282.2.24 fdt_parse_plmt_node() . . . 1169
30.282.2.25 fdt_parse_renesas_scif_node() . . . 1170
30.282.2.26 fdt_parse_shakti_uart_node() . . . 1170
30.282.2.27 fdt_parse_sifive_uart_node() . . . 1171
30.282.2.28 fdt_parse_timebase_frequency() . . . 1172
30.282.2.29 fdt_parse_uart8250() . . . 1172
30.282.2.30 fdt_parse_uart_node() . . . 1172
30.282.2.31 fdt_parse_uart_node_common() . . . 1173
30.282.2.32 fdt_parse_xlnx_uartlite_node() . . . 1174
30.282.2.33 fdt_translate_address() . . . 1174
30.282.3 Variable Documentation . . . 1175
30.282.3.1 fdt_isa_bitmap_offset . . . 1175
30.283 lib/utils/fdt/fdt_pmu.c File Reference . . . 1175
30.283.1 Macro Definition Documentation . . . 1176
30.283.1.1 FDT_PMU_HW_EVENT_MAX . . . 1176
30.283.2 Function Documentation . . . 1176
30.283.2.1 fdt_pmu_fixup() . . . 1176
30.283.2.2 fdt_pmu_get_select_value() . . . 1176
30.283.2.3 fdt_pmu_setup() . . . 1177
30.283.3 Variable Documentation . . . 1177
30.283.3.1 fdt_pmu_evt_select . . . 1177
30.283.3.2 hw_event_count . . . 1177
30.284 lib/utils/gpio/fdt_gpio.c File Reference . . . 1177
30.284.1 Function Documentation . . . 1178
30.284.1.1 fdt_gpio_chip_find() . . . 1178
30.284.1.2 fdt_gpio_init() . . . 1178
30.284.1.3 fdt_gpio_pin_get() . . . 1179
30.284.1.4 fdt_gpio_simple_xlate() . . . 1179
30.284.2 Variable Documentation . . . 1179
30.284.2.1 fdt_gpio_drivers . . . 1179
30.285 lib/utils/gpio/fdt_gpio_designware.c File Reference . . . 1180
30.285.1 Macro Definition Documentation . . . 1180
30.285.1.1 DW_GPIO_BIT . . . 1180
30.285.1.2 DW_GPIO_DDR . . . 1180
30.285.1.3 DW_GPIO_DR . . . 1180

30.285.1.4 DW_GPIO_PINS_MAX . . . 1181
30.285.1.5 pin_to_chip . . . 1181
30.285.2 Function Documentation . . . 1181
30.285.2.1 dw_gpio_direction_output() . . . 1181
30.285.2.2 dw_gpio_init_bank() . . . 1181
30.285.2.3 dw_gpio_set() . . . 1181
30.285.3 Variable Documentation . . . 1182
30.285.3.1 dw_gpio_match . . . 1182
30.285.3.2 fdt_gpio_designware . . . 1182
30.286 lib/utils/gpio/fdt_gpio_sifive.c File Reference . . . 1182
30.286.1 Macro Definition Documentation . . . 1183
30.286.1.1 SIFIVE_GPIO_BIT . . . 1183
30.286.1.2 SIFIVE_GPIO_OUTEN . . . 1183
30.286.1.3 SIFIVE_GPIO_OUTVAL . . . 1183
30.286.1.4 SIFIVE_GPIO_PINS_DEF . . . 1183
30.286.1.5 SIFIVE_GPIO_PINS_MAX . . . 1183
30.286.1.6 SIFIVE_GPIO_PINS_MIN . . . 1183
30.286.2 Function Documentation . . . 1183
30.286.2.1 sifive_gpio_direction_output() . . . 1183
30.286.2.2 sifive_gpio_init() . . . 1184
30.286.2.3 sifive_gpio_set() . . . 1184
30.286.3 Variable Documentation . . . 1184
30.286.3.1 fdt_gpio_sifive . . . 1184
30.286.3.2 sifive_gpio_match . . . 1185
30.287 lib/utils/gpio/fdt_gpio_starfive.c File Reference . . . 1185
30.287.1 Macro Definition Documentation . . . 1186
30.287.1.1 STARFIVE_GPIO_MASK . . . 1186
30.287.1.2 STARFIVE_GPIO_OUTVAL . . . 1186
30.287.1.3 STARFIVE_GPIO_PINS_DEF . . . 1186
30.287.1.4 STARFIVE_GPIO_REG_SHIFT_MASK . . . 1186
30.287.1.5 STARFIVE_GPIO_SHIFT_BITS . . . 1186
30.287.2 Function Documentation . . . 1186
30.287.2.1 starfive_gpio_direction_output() . . . 1186
30.287.2.2 starfive_gpio_init() . . . 1186
30.287.2.3 starfive_gpio_set() . . . 1187
30.287.3 Variable Documentation . . . 1187
30.287.3.1 fdt_gpio_starfive . . . 1187
30.287.3.2 starfive_gpio_match . . . 1187
30.288 lib/utils/gpio/gpio.c File Reference . . . 1187
30.288.1 Function Documentation . . . 1188
30.288.1.1 gpio_chip_add() . . . 1188
30.288.1.2 gpio_chip_find() . . . 1189

30.288.1.3 gpio_chip_remove() . . . 1189
30.288.1.4 gpio_direction_input() . . . 1190
30.288.1.5 gpio_direction_output() . . . 1190
30.288.1.6 gpio_get() . . . 1190
30.288.1.7 gpio_get_direction() . . . 1190
30.288.1.8 gpio_set() . . . 1191
30.288.1.9 SBI_LIST_HEAD() . . . 1191
30.289 lib/utils/hsm/fdt_hsm.c File Reference . . . 1191
30.289.1 Function Documentation . . . 1192
30.289.1.1 fdt_hsm_init() . . . 1192
30.289.2 Variable Documentation . . . 1192
30.289.2.1 fdt_hsm_drivers . . . 1192
30.290 lib/utils/hsm/fdt_hsm_rpmi.c File Reference . . . 1192
30.290.1 Macro Definition Documentation . . . 1193
30.290.1.1 MAX_HSM_SUPSEND_STATE_NAMELEN . . . 1193
30.290.2 Function Documentation . . . 1193
30.290.2.1 is_rpmi_hsm_susp_supported() . . . 1193
30.290.2.2 rpmi_hsm_cold_init() . . . 1193
30.290.2.3 rpmi_hsm_do_fixup() . . . 1194
30.290.2.4 rpmi_hsm_get_num_suspend_states() . . . 1194
30.290.2.5 rpmi_hsm_get_pointer() . . . 1195
30.290.2.6 rpmi_hsm_get_suspend_states() . . . 1195
30.290.2.7 rpmi_hsm_start() . . . 1196
30.290.2.8 rpmi_hsm_stop() . . . 1196
30.290.2.9 rpmi_hsm_suspend() . . . 1196
30.290.2.10 rpmi_hsm_update_hart_scratch() . . . 1197
30.290.3 Variable Documentation . . . 1198
30.290.3.1 fdt_hsm_rpmi . . . 1198
30.290.3.2 rpmi_hsm_fixup . . . 1198
30.290.3.3 rpmi_hsm_match . . . 1198
30.290.3.4 rpmi_hsm_offset . . . 1198
30.290.3.5 sbi_hsm_rpmi . . . 1198
30.291 lib/utils/i2c/dw_i2c.c File Reference . . . 1198
30.291.1 Macro Definition Documentation . . . 1200
30.291.1.1 dw_i2c_adapter_poll_rxrdy . . . 1200
30.291.1.2 dw_i2c_adapter_poll_txfifo_ready . . . 1200
30.291.1.3 DW_I2C_STATUS_RXFIFO_NOT_EMPTY . . . 1200
30.291.1.4 DW_I2C_STATUS_TXFIFO_EMPTY . . . 1200
30.291.1.5 DW_IC_CLR_ACTIVITY . . . 1200
30.291.1.6 DW_IC_CLR_GEN_CALL . . . 1200
30.291.1.7 DW_IC_CLR_INTR . . . 1200
30.291.1.8 DW_IC_CLR_RD_REQ . . . 1200

30.291.1.9 DW_IC_CLR_RESTART_DET . . . 1200
30.291.1.10 DW_IC_CLR_RX_DONE . . . 1201
30.291.1.11 DW_IC_CLR_RX_OVER . . . 1201
30.291.1.12 DW_IC_CLR_RX_UNDER . . . 1201
30.291.1.13 DW_IC_CLR_START_DET . . . 1201
30.291.1.14 DW_IC_CLR_STOP_DET . . . 1201
30.291.1.15 DW_IC_CLR_TX_ABRT . . . 1201
30.291.1.16 DW_IC_CLR_TX_OVER . . . 1201
30.291.1.17 DW_IC_COMP_PARAM_1 . . . 1201
30.291.1.18 DW_IC_COMP_VERSION . . . 1201
30.291.1.19 DW_IC_CON . . . 1201
30.291.1.20 DW_IC_DATA_CMD . . . 1201
30.291.1.21 DW_IC_ENABLE . . . 1201
30.291.1.22 DW_IC_ENABLE_STATUS . . . 1201
30.291.1.23 DW_IC_FS_SCL_HCNT . . . 1201
30.291.1.24 DW_IC_FS_SCL_LCNT . . . 1201
30.291.1.25 DW_IC_HS_SCL_HCNT . . . 1202
30.291.1.26 DW_IC_HS_SCL_LCNT . . . 1202
30.291.1.27 DW_IC_INTR_MASK . . . 1202
30.291.1.28 DW_IC_INTR_STAT . . . 1202
30.291.1.29 DW_IC_RAW_INTR_STAT . . . 1202
30.291.1.30 DW_IC_RX_TL . . . 1202
30.291.1.31 DW_IC_RXFLR . . . 1202
30.291.1.32 DW_IC_SAR . . . 1202
30.291.1.33 DW_IC_SDA_HOLD . . . 1202
30.291.1.34 DW_IC_SS_SCL_HCNT . . . 1202
30.291.1.35 DW_IC_SS_SCL_LCNT . . . 1202
30.291.1.36 DW_IC_STATUS . . . 1202
30.291.1.37 DW_IC_TAR . . . 1202
30.291.1.38 DW_IC_TX_ABRT_SOURCE . . . 1202
30.291.1.39 DW_IC_TX_TL . . . 1202
30.291.1.40 DW_IC_TXFLR . . . 1203
30.291.1.41 IC_DATA_CMD_READ . . . 1203
30.291.1.42 IC_DATA_CMD_RESTART . . . 1203
30.291.1.43 IC_DATA_CMD_STOP . . . 1203
30.291.1.44 IC_INT_STATUS_STOPDET . . . 1203
30.291.2 Function Documentation . . . 1203
30.291.2.1 dw_i2c_adapter_poll() . . . 1203
30.291.2.2 dw_i2c_adapter_read() . . . 1203
30.291.2.3 dw_i2c_adapter_write() . . . 1204
30.291.2.4 dw_i2c_getreg() . . . 1205
30.291.2.5 dw_i2c_init() . . . 1205

30.291.2.6 dw_i2c_setreg() . . . 1205
30.291.2.7 dw_i2c_write_addr() . . . 1206
30.292 lib/utils/i2c/fdt_i2c.c File Reference . . . 1206
30.292.1 Function Documentation . . . 1207
30.292.1.1 fdt_i2c_adapter_find() . . . 1207
30.292.1.2 fdt_i2c_adapter_get() . . . 1208
30.292.2 Variable Documentation . . . 1208
30.292.2.1 fdt_i2c_adapter_drivers . . . 1208
30.293 lib/utils/i2c/fdt_i2c_dw.c File Reference . . . 1208
30.293.1 Function Documentation . . . 1209
30.293.1.1 fdt_dw_i2c_init() . . . 1209
30.293.2 Variable Documentation . . . 1210
30.293.2.1 fdt_dw_i2c_match . . . 1210
30.293.2.2 fdt_i2c_adapter_dw . . . 1210
30.294 lib/utils/i2c/fdt_i2c_sifive.c File Reference . . . 1210
30.294.1 Macro Definition Documentation . . . 1211
30.294.1.1 sifive_i2c_adapter_poll_busy . . . 1211
30.294.1.2 sifive_i2c_adapter_poll_tip . . . 1211
30.294.1.3 SIFIVE_I2C_CMD_ACK . . . 1211
30.294.1.4 SIFIVE_I2C_CMD_IACK . . . 1211
30.294.1.5 SIFIVE_I2C_CMD_RD . . . 1211
30.294.1.6 SIFIVE_I2C_CMD_STA . . . 1211
30.294.1.7 SIFIVE_I2C_CMD_STO . . . 1212
30.294.1.8 SIFIVE_I2C_CMD_WR . . . 1212
30.294.1.9 SIFIVE_I2C_CR . . . 1212
30.294.1.10 SIFIVE_I2C_CTR . . . 1212
30.294.1.11 SIFIVE_I2C_CTR_EN . . . 1212
30.294.1.12 SIFIVE_I2C_CTR_IEN . . . 1212
30.294.1.13 SIFIVE_I2C_PREHI . . . 1212
30.294.1.14 SIFIVE_I2C_PRELO . . . 1212
30.294.1.15 SIFIVE_I2C_READ_BIT . . . 1212
30.294.1.16 SIFIVE_I2C_RXR . . . 1212
30.294.1.17 SIFIVE_I2C_SR . . . 1212
30.294.1.18 SIFIVE_I2C_STATUS_AL . . . 1212
30.294.1.19 SIFIVE_I2C_STATUS_BUSY . . . 1212
30.294.1.20 SIFIVE_I2C_STATUS_IF . . . 1212
30.294.1.21 SIFIVE_I2C_STATUS_RXACK . . . 1212
30.294.1.22 SIFIVE_I2C_STATUS_TIP . . . 1213
30.294.1.23 SIFIVE_I2C_TXR . . . 1213
30.294.1.24 SIFIVE_I2C_WRITE_BIT . . . 1213
30.294.2 Function Documentation . . . 1213
30.294.2.1 sifive_i2c_adapter_poll() . . . 1213

30.294.2.2 sifive_i2c_adapter_read() . . . 1213
30.294.2.3 sifive_i2c_adapter_rxack() . . . 1214
30.294.2.4 sifive_i2c_adapter_start() . . . 1214
30.294.2.5 sifive_i2c_adapter_write() . . . 1215
30.294.2.6 sifive_i2c_getreg() . . . 1216
30.294.2.7 sifive_i2c_init() . . . 1216
30.294.2.8 sifive_i2c_setreg() . . . 1217
30.294.3 Variable Documentation . . . 1217
30.294.3.1 fdt_i2c_adapter_sifive . . . 1217
30.294.3.2 sifive_i2c_match . . . 1217
30.295 lib/utils/i2c/i2c.c File Reference . . . 1217
30.295.1 Function Documentation . . . 1218
30.295.1.1 i2c_adapter_add() . . . 1218
30.295.1.2 i2c_adapter_find() . . . 1219
30.295.1.3 i2c_adapter_read() . . . 1219
30.295.1.4 i2c_adapter_remove() . . . 1220
30.295.1.5 i2c_adapter_write() . . . 1220
30.295.1.6 SBI_LIST_HEAD() . . . 1220
30.296 lib/utils/ipi/aclint_mswi.c File Reference . . . 1220
30.296.1 Macro Definition Documentation . . . 1221
30.296.1.1 mswi_get_hart_data_ptr . . . 1221
30.296.1.2 mswi_set_hart_data_ptr . . . 1221
30.296.2 Function Documentation . . . 1221
30.296.2.1 aclint_mswi_cold_init() . . . 1221
30.296.2.2 mswi_ipi_clear() . . . 1222
30.296.2.3 mswi_ipi_send() . . . 1222
30.296.3 Variable Documentation . . . 1222
30.296.3.1 aclint_mswi . . . 1222
30.296.3.2 mswi_ptr_offset . . . 1222
30.297 lib/utils/ipi/andes_plicsw.c File Reference . . . 1222
30.297.1 Function Documentation . . . 1223
30.297.1.1 plicsw_cold_ipi_init() . . . 1223
30.297.1.2 plicsw_ipi_clear() . . . 1224
30.297.1.3 plicsw_ipi_send() . . . 1224
30.297.2 Variable Documentation . . . 1224
30.297.2.1 plicsw . . . 1224
30.297.2.2 plicsw_ipi . . . 1224
30.298 lib/utils/ipi/fdt_ipi.c File Reference . . . 1224
30.298.1 Function Documentation . . . 1225
30.298.1.1 fdt_ipi_init() . . . 1225
30.298.2 Variable Documentation . . . 1225
30.298.2.1 fdt_ipi_drivers . . . 1225

30.299 lib/utils/ipi/fdt_ipi_mswi.c File Reference . . . 1225
30.299.1 Function Documentation . . . 1226
30.299.1.1 ipi_mswi_cold_init() . . . 1226
30.299.2 Variable Documentation . . . 1226
30.299.2.1 clint_offset . . . 1226
30.299.2.2 fdt_ipi_mswi . . . 1226
30.299.2.3 ipi_mswi_match . . . 1226
30.300 lib/utils/ipi/fdt_ipi_plicsw.c File Reference . . . 1227
30.300.1 Function Documentation . . . 1227
30.300.1.1 fdt_plicsw_cold_ipi_init() . . . 1227
30.300.2 Variable Documentation . . . 1228
30.300.2.1 fdt_ipi_plicsw . . . 1228
30.300.2.2 ipi_plicsw_match . . . 1228
30.300.2.3 plicsw . . . 1228
30.301 lib/utils/irqchip/aplic.c File Reference . . . 1228
30.301.1 Macro Definition Documentation . . . 1230
30.301.1.1 APLIC_CLRIE_BASE . . . 1230
30.301.1.2 APLIC_CLRIENUM . . . 1230
30.301.1.3 APLIC_CLRIP_BASE . . . 1230
30.301.1.4 APLIC_CLRIPNUM . . . 1230
30.301.1.5 APLIC_DEFAULT_PRIORITY . . . 1230
30.301.1.6 APLIC_DISABLE_IDELIVERY . . . 1230
30.301.1.7 APLIC_DISABLE_ITHRESHOLD . . . 1230
30.301.1.8 APLIC_DOMAINCFG . . . 1230
30.301.1.9 APLIC_DOMAINCFG_BE . . . 1231
30.301.1.10 APLIC_DOMAINCFG_DM . . . 1231
30.301.1.11 APLIC_DOMAINCFG_IE . . . 1231
30.301.1.12 APLIC_ENABLE_IDELIVERY . . . 1231
30.301.1.13 APLIC_ENABLE_ITHRESHOLD . . . 1231
30.301.1.14 APLIC_IDC_BASE . . . 1231
30.301.1.15 APLIC_IDC_CLAIMI . . . 1231
30.301.1.16 APLIC_IDC_IDELIVERY . . . 1231
30.301.1.17 APLIC_IDC_IFORCE . . . 1231
30.301.1.18 APLIC_IDC_ITHRESHOLD . . . 1231
30.301.1.19 APLIC_IDC_SIZE . . . 1231
30.301.1.20 APLIC_IDC_TOPI . . . 1231
30.301.1.21 APLIC_IDC_TOPI_ID_MASK . . . 1231
30.301.1.22 APLIC_IDC_TOPI_ID_SHIFT . . . 1231
30.301.1.23 APLIC_IDC_TOPI_PRIO_MASK . . . 1231
30.301.1.24 APLIC_MAX_IDC . . . 1232
30.301.1.25 APLIC_MAX_SOURCE . . . 1232
30.301.1.26 APLIC_MMSICFGADDR . . . 1232

30.301.1.27 APLIC_MMSICFGADDRH . . . 1232
30.301.1.28 APLIC_SETIE_BASE . . . 1232
30.301.1.29 APLIC_SETIENUM . . . 1232
30.301.1.30 APLIC_SETIP_BASE . . . 1232
30.301.1.31 APLIC_SETIPNUM . . . 1232
30.301.1.32 APLIC_SETIPNUM_BE . . . 1232
30.301.1.33 APLIC_SETIPNUM_LE . . . 1232
30.301.1.34 APLIC_SMSICFGADDR . . . 1232
30.301.1.35 APLIC_SMSICFGADDRH . . . 1232
30.301.1.36 APLIC_SOURCECFG_BASE . . . 1232
30.301.1.37 APLIC_SOURCECFG_CHILDIDX_MASK . . . 1232
30.301.1.38 APLIC_SOURCECFG_D . . . 1232
30.301.1.39 APLIC_SOURCECFG_SM_DETACH . . . 1233
30.301.1.40 APLIC_SOURCECFG_SM_EDGE_FALL . . . 1233
30.301.1.41 APLIC_SOURCECFG_SM_EDGE_RISE . . . 1233
30.301.1.42 APLIC_SOURCECFG_SM_INACTIVE . . . 1233
30.301.1.43 APLIC_SOURCECFG_SM_LEVEL_HIGH . . . 1233
30.301.1.44 APLIC_SOURCECFG_SM_LEVEL_LOW . . . 1233
30.301.1.45 APLIC_SOURCECFG_SM_MASK . . . 1233
30.301.1.46 APLIC_TARGET_BASE . . . 1233
30.301.1.47 APLIC_TARGET_EIID_MASK . . . 1233
30.301.1.48 APLIC_TARGET_GUEST_IDX_MASK . . . 1233
30.301.1.49 APLIC_TARGET_GUEST_IDX_SHIFT . . . 1233
30.301.1.50 APLIC_TARGET_HART_IDX_MASK . . . 1233
30.301.1.51 APLIC_TARGET_HART_IDX_SHIFT . . . 1233
30.301.1.52 APLIC_TARGET_IPRIO_MASK . . . 1233
30.301.1.53 APLIC_xMSICFGADDR_PPN_HART . . . 1233
30.301.1.54 APLIC_xMSICFGADDR_PPN_HHX . . . 1234
30.301.1.55 APLIC_xMSICFGADDR_PPN_HHX_MASK . . . 1234
30.301.1.56 APLIC_xMSICFGADDR_PPN_HHX_SHIFT . . . 1234
30.301.1.57 APLIC_xMSICFGADDR_PPN_LHX . . . 1234
30.301.1.58 APLIC_xMSICFGADDR_PPN_LHX_MASK . . . 1234
30.301.1.59 APLIC_xMSICFGADDR_PPN_LHX_SHIFT . . . 1234
30.301.1.60 APLIC_xMSICFGADDR_PPN_SHIFT . . . 1234
30.301.1.61 APLIC_xMSICFGADDRH_BAPPN_MASK . . . 1234
30.301.1.62 APLIC_xMSICFGADDRH_HHXS_MASK . . . 1234
30.301.1.63 APLIC_xMSICFGADDRH_HHXS_SHIFT . . . 1234
30.301.1.64 APLIC_xMSICFGADDRH_HHXW_MASK . . . 1234
30.301.1.65 APLIC_xMSICFGADDRH_HHXW_SHIFT . . . 1234
30.301.1.66 APLIC_xMSICFGADDRH_L . . . 1235
30.301.1.67 APLIC_xMSICFGADDRH_LHXS_MASK . . . 1235
30.301.1.68 APLIC_xMSICFGADDRH_LHXS_SHIFT . . . 1235

30.301.1.69 APLIC_xMSICFGADDRH_LHXW_MASK . . . 1235
30.301.1.70 APLIC_xMSICFGADDRH_LHXW_SHIFT . . . 1235
30.301.2 Function Documentation . . . 1235
30.301.2.1 aplic_check_msicfg() . . . 1235
30.301.2.2 aplic_cold_irqchip_init() . . . 1235
30.301.2.3 aplic_writel_msicfg() . . . 1236
30.302 lib/utils/irqchip/fdt_irqchip.c File Reference . . . 1236
30.302.1 Function Documentation . . . 1237
30.302.1.1 fdt_irqchip_init() . . . 1237
30.302.2 Variable Documentation . . . 1237
30.302.2.1 fdt_irqchip_drivers . . . 1237
30.303 lib/utils/irqchip/fdt_irqchip_aplic.c File Reference . . . 1237
30.303.1 Function Documentation . . . 1238
30.303.1.1 irqchip_aplic_cold_init() . . . 1238
30.303.2 Variable Documentation . . . 1238
30.303.2.1 fdt_irqchip_aplic . . . 1238
30.303.2.2 irqchip_aplic_match . . . 1238
30.304 lib/utils/irqchip/fdt_irqchip_imsic.c File Reference . . . 1238
30.304.1 Function Documentation . . . 1239
30.304.1.1 irqchip_imsic_cold_init() . . . 1239
30.304.1.2 irqchip_imsic_update_hartid_table() . . . 1239
30.304.2 Variable Documentation . . . 1240
30.304.2.1 fdt_irqchip_imsic . . . 1240
30.304.2.2 irqchip_imsic_match . . . 1240
30.305 lib/utils/irqchip/fdt_irqchip_plic.c File Reference . . . 1240
30.305.1 Function Documentation . . . 1241
30.305.1.1 irqchip_plic_cold_init() . . . 1241
30.305.1.2 irqchip_plic_update_context_map() . . . 1241
30.305.2 Variable Documentation . . . 1242
30.305.2.1 fdt_irqchip_plic . . . 1242
30.305.2.2 irqchip_plic_match . . . 1242
30.306 lib/utils/irqchip/imsic.c File Reference . . . 1242
30.306.1 Macro Definition Documentation . . . 1244
30.306.1.1 imsic_csr_clear . . . 1244
30.306.1.2 imsic_csr_read . . . 1244
30.306.1.3 imsic_csr_set . . . 1244
30.306.1.4 imsic_csr_write . . . 1244
30.306.1.5 IMSIC_DISABLE_EIDELIVERY . . . 1244
30.306.1.6 IMSIC_DISABLE_EITHRESHOLD . . . 1245
30.306.1.7 IMSIC_EIDELIVERY . . . 1245
30.306.1.8 IMSIC_EIE0 . . . 1245
30.306.1.9 IMSIC_EIE63 . . . 1245

30.306.1.10 IMSIC_EIEx_BITS . . . 1245
30.306.1.11 IMSIC_EIP0 . . . 1245
30.306.1.12 IMSIC_EIP63 . . . 1245
30.306.1.13 IMSIC_EIPx_BITS . . . 1245
30.306.1.14 IMSIC_EITHRESHOLD . . . 1245
30.306.1.15 IMSIC_ENABLE_EIDELIVERY . . . 1245
30.306.1.16 IMSIC_ENABLE_EITHRESHOLD . . . 1245
30.306.1.17 imsic_get_hart_data_ptr . . . 1245
30.306.1.18 imsic_get_hart_file . . . 1245
30.306.1.19 IMSIC_IPI_ID . . . 1245
30.306.1.20 IMSIC_MAX_ID . . . 1245
30.306.1.21 IMSIC_MIN_ID . . . 1246
30.306.1.22 IMSIC_MMIO_PAGE_BE . . . 1246
30.306.1.23 IMSIC_MMIO_PAGE_LE . . . 1246
30.306.1.24 imsic_set_hart_data_ptr . . . 1246
30.306.1.25 imsic_set_hart_file . . . 1246
30.306.1.26 IMSIC_TOPEI . . . 1246
30.306.1.27 IMSIC_TOPEI_ID_MASK . . . 1246
30.306.1.28 IMSIC_TOPEI_ID_SHIFT . . . 1246
30.306.1.29 IMSIC_TOPEI_PRIO_MASK . . . 1246
30.306.2 Function Documentation . . . 1246
30.306.2.1 imsic_cold_irqchip_init() . . . 1246
30.306.2.2 imsic_data_check() . . . 1247
30.306.2.3 imsic_external_irqfn() . . . 1247
30.306.2.4 imsic_get_data() . . . 1247
30.306.2.5 imsic_get_target_file() . . . 1248
30.306.2.6 imsic_ipi_send() . . . 1248
30.306.2.7 imsic_local_eix_update() . . . 1248
30.306.2.8 imsic_local_irqchip_init() . . . 1248
30.306.2.9 imsic_map_hartid_to_data() . . . 1248
30.306.2.10 imsic_warm_irqchip_init() . . . 1249
30.306.3 Variable Documentation . . . 1249
30.306.3.1 imsic_device . . . 1249
30.306.3.2 imsic_file_offset . . . 1249
30.306.3.3 imsic_ipi_device . . . 1249
30.306.3.4 imsic_ptr_offset . . . 1249
30.307 lib/utils/irqchip/plic.c File Reference . . . 1249
30.307.1 Macro Definition Documentation . . . 1250
30.307.1.1 PLIC_CONTEXT_BASE . . . 1250
30.307.1.2 PLIC_CONTEXT_STRIDE . . . 1251
30.307.1.3 PLIC_ENABLE_BASE . . . 1251
30.307.1.4 PLIC_ENABLE_STRIDE . . . 1251

30.307.1.5 plic_get_hart_data_ptr . . . 1251
30.307.1.6 PLIC_PENDING_BASE . . . 1251
30.307.1.7 PLIC_PRIORITY_BASE . . . 1251
30.307.1.8 plic_set_hart_data_ptr . . . 1251
30.307.1.9 THEAD_PLIC_CTRL_REG . . . 1251
30.307.2 Function Documentation . . . 1251
30.307.2.1 plic_cold_irqchip_init() . . . 1251
30.307.2.2 plic_context_init() . . . 1252
30.307.2.3 plic_delegate() . . . 1252
30.307.2.4 plic_get() . . . 1253
30.307.2.5 plic_get_ie() . . . 1253
30.307.2.6 plic_get_priority() . . . 1253
30.307.2.7 plic_get_thresh() . . . 1254
30.307.2.8 plic_resume() . . . 1254
30.307.2.9 plic_set_ie() . . . 1254
30.307.2.10 plic_set_priority() . . . 1255
30.307.2.11 plic_set_thresh() . . . 1255
30.307.2.12 plic_suspend() . . . 1255
30.307.2.13 plic_warm_irqchip_init() . . . 1256
30.307.3 Variable Documentation . . . 1256
30.307.3.1 plic_ptr_offset . . . 1256
30.308 lib/utils/libfdt/fdt.c File Reference . . . 1256
30.308.1 Function Documentation . . . 1257
30.308.1.1 check_block_() . . . 1257
30.308.1.2 check_off_() . . . 1258
30.308.1.3 fdt_check_header() . . . 1258
30.308.1.4 fdt_check_node_offset_() . . . 1259
30.308.1.5 fdt_check_prop_offset_() . . . 1260
30.308.1.6 fdt_find_string_() . . . 1261
30.308.1.7 fdt_first_subnode() . . . 1261
30.308.1.8 fdt_header_size() . . . 1262
30.308.1.9 fdt_header_size_() . . . 1262
30.308.1.10 fdt_move() . . . 1263
30.308.1.11 fdt_next_node() . . . 1263
30.308.1.12 fdt_next_subnode() . . . 1264
30.308.1.13 fdt_next_tag() . . . 1265
30.308.1.14 fdt_offset_ptr() . . . 1266
30.308.1.15 fdt_ro_probe_() . . . 1266
30.309 lib/utils/libfdt/fdt.h File Reference . . . 1267
30.309.1 Macro Definition Documentation . . . 1267
30.309.1.1 FDT_BEGIN_NODE . . . 1267
30.309.1.2 FDT_END . . . 1268

30.309.1.3 FDT_END_NODE . . . 1268
30.309.1.4 FDT_MAGIC . . . 1268
30.309.1.5 FDT_NOP . . . 1268
30.309.1.6 FDT_PROP . . . 1268
30.309.1.7 FDT_TAGSIZE . . . 1268
30.309.1.8 FDT_V16_SIZE . . . 1268
30.309.1.9 FDT_V17_SIZE . . . 1268
30.309.1.10 FDT_V1_SIZE . . . 1268
30.309.1.11 FDT_V2_SIZE . . . 1268
30.309.1.12 FDT_V3_SIZE . . . 1268
30.310 fdt.h . . . 1268
30.311 lib/utils/libfdt/fdt_addresses.c File Reference . . . 1269
30.311.1 Function Documentation . . . 1270
30.311.1.1 fdt_address_cells() . . . 1270
30.311.1.2 fdt_appendprop_addrrange() . . . 1271
30.311.1.3 fdt_cells() . . . 1271
30.311.1.4 fdt_size_cells() . . . 1272
30.312 lib/utils/libfdt/fdt_check.c File Reference . . . 1273
30.312.1 Function Documentation . . . 1274
30.312.1.1 fdt_check_full() . . . 1274
30.313 lib/utils/libfdt/fdt_empty_tree.c File Reference . . . 1275
30.313.1 Function Documentation . . . 1275
30.313.1.1 fdt_create_empty_tree() . . . 1275
30.314 lib/utils/libfdt/fdt_overlay.c File Reference . . . 1276
30.314.1 Function Documentation . . . 1277
30.314.1.1 fdt_overlay_apply() . . . 1277
30.314.1.2 get_path_len() . . . 1278
30.314.1.3 overlay_adjust_local_phandles() . . . 1278
30.314.1.4 overlay_adjust_node_phandles() . . . 1279
30.314.1.5 overlay_apply_node() . . . 1279
30.314.1.6 overlay_fixup_one_phandle() . . . 1280
30.314.1.7 overlay_fixup_phandle() . . . 1281
30.314.1.8 overlay_fixup_phandles() . . . 1281
30.314.1.9 overlay_get_target() . . . 1282
30.314.1.10 overlay_get_target_phandle() . . . 1282
30.314.1.11 overlay_merge() . . . 1283
30.314.1.12 overlay_phandle_add_offset() . . . 1284
30.314.1.13 overlay_symbol_update() . . . 1284
30.314.1.14 overlay_update_local_node_references() . . . 1285
30.314.1.15 overlay_update_local_references() . . . 1286
30.315 lib/utils/libfdt/fdt_ro.c File Reference . . . 1286
30.315.1 Function Documentation . . . 1288

30.315.1.1 fdt_find_max_phandle() . . . 1288
30.315.1.2 fdt_first_property_offset() . . . 1289
30.315.1.3 fdt_generate_phandle() . . . 1290
30.315.1.4 fdt_get_alias() . . . 1291
30.315.1.5 fdt_get_alias_namelen() . . . 1291
30.315.1.6 fdt_get_mem_rsv() . . . 1292
30.315.1.7 fdt_get_name() . . . 1292
30.315.1.8 fdt_get_path() . . . 1293
30.315.1.9 fdt_get_phandle() . . . 1294
30.315.1.10 fdt_get_property() . . . 1295
30.315.1.11 fdt_get_property_by_offset() . . . 1296
30.315.1.12 fdt_get_property_by_offset_() . . . 1296
30.315.1.13 fdt_get_property_namelen() . . . 1297
30.315.1.14 fdt_get_property_namelen_() . . . 1298
30.315.1.15 fdt_get_string() . . . 1299
30.315.1.16 fdt_getprop() . . . 1300
30.315.1.17 fdt_getprop_by_offset() . . . 1301
30.315.1.18 fdt_getprop_namelen() . . . 1302
30.315.1.19 fdt_mem_rsv() . . . 1303
30.315.1.20 fdt_next_property_offset() . . . 1304
30.315.1.21 fdt_node_check_compatible() . . . 1305
30.315.1.22 fdt_node_depth() . . . 1306
30.315.1.23 fdt_node_offset_by_compatible() . . . 1307
30.315.1.24 fdt_node_offset_by_phandle() . . . 1308
30.315.1.25 fdt_node_offset_by_prop_value() . . . 1309
30.315.1.26 fdt_nodename_eq_() . . . 1310
30.315.1.27 fdt_num_mem_rsv() . . . 1311
30.315.1.28 fdt_parent_offset() . . . 1311
30.315.1.29 fdt_path_offset() . . . 1312
30.315.1.30 fdt_path_offset_namelen() . . . 1313
30.315.1.31 fdt_string() . . . 1314
30.315.1.32 fdt_string_eq_() . . . 1315
30.315.1.33 fdt_stringlist_contains() . . . 1316
30.315.1.34 fdt_stringlist_count() . . . 1317
30.315.1.35 fdt_stringlist_get() . . . 1318
30.315.1.36 fdt_stringlist_search() . . . 1318
30.315.1.37 fdt_subnode_offset() . . . 1319
30.315.1.38 fdt_subnode_offset_namelen() . . . 1319
30.315.1.39 fdt_supernode_atdepth_offset() . . . 1320
30.315.1.40 nextprop_() . . . 1321
30.316 lib/utils/libfdt/fdt_rw.c File Reference . . . 1322
30.316.1 Macro Definition Documentation . . . 1324

30.316.1.1 FDT_RW_PROBE . . . 1324
30.316.2 Function Documentation . . . 1324
30.316.2.1 fdt_add_mem_rsv() . . . 1324
30.316.2.2 fdt_add_property_() . . . 1324
30.316.2.3 fdt_add_subnode() . . . 1325
30.316.2.4 fdt_add_subnode_namelen() . . . 1326
30.316.2.5 fdt_appendprop() . . . 1326
30.316.2.6 fdt_blocks_misordered_() . . . 1327
30.316.2.7 fdt_data_size_() . . . 1327
30.316.2.8 fdt_del_last_string_() . . . 1328
30.316.2.9 fdt_del_mem_rsv() . . . 1328
30.316.2.10 fdt_del_node() . . . 1329
30.316.2.11 fdt_delprop() . . . 1329
30.316.2.12 fdt_find_add_string_() . . . 1330
30.316.2.13 fdt_open_into() . . . 1330
30.316.2.14 fdt_pack() . . . 1331
30.316.2.15 fdt_packblocks_() . . . 1331
30.316.2.16 fdt_resize_property_() . . . 1332
30.316.2.17 fdt_rw_probe_() . . . 1332
30.316.2.18 fdt_set_name() . . . 1333
30.316.2.19 fdt_setprop() . . . 1333
30.316.2.20 fdt_setprop_placeholder() . . . 1334
30.316.2.21 fdt_splice_() . . . 1334
30.316.2.22 fdt_splice_mem_rsv_() . . . 1335
30.316.2.23 fdt_splice_string_() . . . 1336
30.316.2.24 fdt_splice_struct_() . . . 1336
30.317 lib/utils/libfdt/fdt_strerror.c File Reference . . . 1337
30.317.1 Macro Definition Documentation . . . 1338
30.317.1.1 FDT_ERRTABENT . . . 1338
30.317.1.2 FDT_ERRTABSIZE . . . 1338
30.317.2 Function Documentation . . . 1338
30.317.2.1 fdt_strerror() . . . 1338
30.317.3 Variable Documentation . . . 1338
30.317.3.1 fdt_errtable . . . 1338
30.318 lib/utils/libfdt/fdt_sw.c File Reference . . . 1338
30.318.1 Macro Definition Documentation . . . 1340
30.318.1.1 FDT_SW_PROBE . . . 1340
30.318.1.2 FDT_SW_PROBE_MEMRSV . . . 1340
30.318.1.3 FDT_SW_PROBE_STRUCT . . . 1340
30.318.2 Function Documentation . . . 1340
30.318.2.1 fdt_add_reservemap_entry() . . . 1340
30.318.2.2 fdt_add_string_() . . . 1341

30.318.2.3 fdt_begin_node() . . . 1341
30.318.2.4 fdt_create() . . . 1341
30.318.2.5 fdt_create_with_flags() . . . 1342
30.318.2.6 fdt_del_last_string_() . . . 1342
30.318.2.7 fdt_end_node() . . . 1343
30.318.2.8 fdt_find_add_string_() . . . 1343
30.318.2.9 fdt_finish() . . . 1344
30.318.2.10 fdt_finish_reservemap() . . . 1344
30.318.2.11 fdt_grab_space_() . . . 1345
30.318.2.12 fdt_property() . . . 1345
30.318.2.13 fdt_property_placeholder() . . . 1346
30.318.2.14 fdt_resize() . . . 1346
30.318.2.15 fdt_sw_probe_() . . . 1347
30.318.2.16 fdt_sw_probe_memrsv_() . . . 1347
30.318.2.17 fdt_sw_probe_struct_() . . . 1347
30.318.2.18 sw_flags() . . . 1347
30.319 lib/utils/libfdt/fdt_wip.c File Reference . . . 1348
30.319.1 Function Documentation . . . 1349
30.319.1.1 fdt_node_end_offset_() . . . 1349
30.319.1.2 fdt_nop_node() . . . 1349
30.319.1.3 fdt_nop_property() . . . 1350
30.319.1.4 fdt_nop_region_() . . . 1350
30.319.1.5 fdt_setprop_inplace() . . . 1351
30.319.1.6 fdt_setprop_inplace_namelen_partial() . . . 1351
30.320 lib/utils/libfdt/libfdt.h File Reference . . . 1352
30.320.1 Macro Definition Documentation . . . 1358
30.320.1.1 fdt_appendprop_string . . . 1358
30.320.1.2 fdt_boot_cpuid_phys . . . 1358
30.320.1.3 FDT_CREATE_FLAG_NO_NAME_DEDUP . . . 1358
30.320.1.4 FDT_CREATE_FLAGS_ALL . . . 1358
30.320.1.5 FDT_ERR_ALIGNMENT . . . 1358
30.320.1.6 FDT_ERR_BADFLAGS . . . 1358
30.320.1.7 FDT_ERR_BADLAYOUT . . . 1358
30.320.1.8 FDT_ERR_BADMAGIC . . . 1358
30.320.1.9 FDT_ERR_BADNCELLS . . . 1358
30.320.1.10 FDT_ERR_BADOFFSET . . . 1358
30.320.1.11 FDT_ERR_BADOVERLAY . . . 1358
30.320.1.12 FDT_ERR_BADPATH . . . 1358
30.320.1.13 FDT_ERR_BADPHANDLE . . . 1358
30.320.1.14 FDT_ERR_BADSTATE . . . 1359
30.320.1.15 FDT_ERR_BADSTRUCTURE . . . 1359
30.320.1.16 FDT_ERR_BADVALUE . . . 1359

30.320.1.17 FDT_ERR_BADVERSION . . . 1359
30.320.1.18 FDT_ERR_EXISTS . . . 1359
30.320.1.19 FDT_ERR_INTERNAL . . . 1359
30.320.1.20 FDT_ERR_MAX . . . 1359
30.320.1.21 FDT_ERR_NOPHANDLES . . . 1359
30.320.1.22 FDT_ERR_NOSPACE . . . 1359
30.320.1.23 FDT_ERR_NOTFOUND . . . 1359
30.320.1.24 FDT_ERR_TRUNCATED . . . 1359
30.320.1.25 FDT_FIRST_SUPPORTED_VERSION . . . 1359
30.320.1.26 fdt_for_each_property_offset . . . 1359
30.320.1.27 fdt_for_each_subnode . . . 1360
30.320.1.28 fdt_get_header . . . 1360
30.320.1.29 fdt_last_comp_version . . . 1360
30.320.1.30 FDT_LAST_COMPATIBLE_VERSION . . . 1360
30.320.1.31 FDT_LAST_SUPPORTED_VERSION . . . 1360
30.320.1.32 fdt_magic . . . 1360
30.320.1.33 FDT_MAX_NCELLS . . . 1360
30.320.1.34 FDT_MAX_PHANDLE . . . 1360
30.320.1.35 fdt_off_dt_strings . . . 1360
30.320.1.36 fdt_off_dt_struct . . . 1361
30.320.1.37 fdt_off_mem_rsvmap . . . 1361
30.320.1.38 fdt_property_string . . . 1361
30.320.1.39 fdt_set_hdr_ . . . 1361
30.320.1.40 fdt_setprop_empty . . . 1361
30.320.1.41 fdt_setprop_string . . . 1361
30.320.1.42 fdt_size_dt_strings . . . 1361
30.320.1.43 fdt_size_dt_struct . . . 1361
30.320.1.44 fdt_totalsize . . . 1361
30.320.1.45 fdt_version . . . 1361
30.320.2 Function Documentation . . . 1362
30.320.2.1 fdt32_ld() . . . 1362
30.320.2.2 fdt32_st() . . . 1362
30.320.2.3 fdt64_ld() . . . 1362
30.320.2.4 fdt64_st() . . . 1362
30.320.2.5 fdt_add_mem_rsv() . . . 1362
30.320.2.6 fdt_add_reservemap_entry() . . . 1363
30.320.2.7 fdt_add_subnode() . . . 1363
30.320.2.8 fdt_add_subnode_namelen() . . . 1364
30.320.2.9 fdt_address_cells() . . . 1365
30.320.2.10 fdt_appendprop() . . . 1365
30.320.2.11 fdt_appendprop_addrrange() . . . 1366
30.320.2.12 fdt_appendprop_cell() . . . 1366

30.320.2.13 fdt_appendprop_u32() . . . 1367
30.320.2.14 fdt_appendprop_u64() . . . 1367
30.320.2.15 fdt_begin_node() . . . 1368
30.320.2.16 fdt_check_full() . . . 1368
30.320.2.17 fdt_check_header() . . . 1369
30.320.2.18 fdt_create() . . . 1370
30.320.2.19 fdt_create_empty_tree() . . . 1370
30.320.2.20 fdt_create_with_flags() . . . 1371
30.320.2.21 fdt_del_mem_rsv() . . . 1371
30.320.2.22 fdt_del_node() . . . 1372
30.320.2.23 fdt_delprop() . . . 1372
30.320.2.24 fdt_end_node() . . . 1373
30.320.2.25 fdt_find_max_phandle() . . . 1373
30.320.2.26 fdt_finish() . . . 1374
30.320.2.27 fdt_finish_reservemap() . . . 1375
30.320.2.28 fdt_first_property_offset() . . . 1375
30.320.2.29 fdt_first_subnode() . . . 1376
30.320.2.30 fdt_generate_phandle() . . . 1377
30.320.2.31 fdt_get_alias() . . . 1377
30.320.2.32 fdt_get_alias_namelen() . . . 1377
30.320.2.33 fdt_get_max_phandle() . . . 1378
30.320.2.34 fdt_get_mem_rsv() . . . 1378
30.320.2.35 fdt_get_name() . . . 1379
30.320.2.36 fdt_get_path() . . . 1380
30.320.2.37 fdt_get_phandle() . . . 1381
30.320.2.38 fdt_get_property() . . . 1382
30.320.2.39 fdt_get_property_by_offset() . . . 1382
30.320.2.40 fdt_get_property_namelen() . . . 1382
30.320.2.41 fdt_get_property_w() . . . 1383
30.320.2.42 fdt_get_string() . . . 1384
30.320.2.43 fdt_getprop() . . . 1385
30.320.2.44 fdt_getprop_by_offset() . . . 1385
30.320.2.45 fdt_getprop_namelen() . . . 1386
30.320.2.46 fdt_getprop_namelen_w() . . . 1387
30.320.2.47 fdt_getprop_w() . . . 1388
30.320.2.48 fdt_header_size() . . . 1388
30.320.2.49 fdt_header_size_() . . . 1389
30.320.2.50 fdt_move() . . . 1389
30.320.2.51 fdt_next_node() . . . 1390
30.320.2.52 fdt_next_property_offset() . . . 1391
30.320.2.53 fdt_next_subnode() . . . 1393
30.320.2.54 fdt_next_tag() . . . 1393

30.320.2.55 fdt_node_check_compatible() . . . 1394
30.320.2.56 fdt_node_depth() . . . 1395
30.320.2.57 fdt_node_offset_by_compatible() . . . 1396
30.320.2.58 fdt_node_offset_by_phandle() . . . 1397
30.320.2.59 fdt_node_offset_by_prop_value() . . . 1398
30.320.2.60 fdt_nop_node() . . . 1399
30.320.2.61 fdt_nop_property() . . . 1399
30.320.2.62 fdt_num_mem_rsv() . . . 1400
30.320.2.63 fdt_offset_ptr() . . . 1401
30.320.2.64 fdt_offset_ptr_w() . . . 1401
30.320.2.65 fdt_open_into() . . . 1402
30.320.2.66 fdt_overlay_apply() . . . 1402
30.320.2.67 fdt_pack() . . . 1403
30.320.2.68 fdt_parent_offset() . . . 1403
30.320.2.69 fdt_path_offset() . . . 1404
30.320.2.70 fdt_path_offset_namelen() . . . 1405
30.320.2.71 fdt_property() . . . 1406
30.320.2.72 fdt_property_cell() . . . 1406
30.320.2.73 fdt_property_placeholder() . . . 1407
30.320.2.74 fdt_property_u32() . . . 1407
30.320.2.75 fdt_property_u64() . . . 1408
30.320.2.76 fdt_resize() . . . 1408
30.320.2.77 fdt_set_hdr_() [1/10] . . . 1408
30.320.2.78 fdt_set_hdr_() [2/10] . . . 1409
30.320.2.79 fdt_set_hdr_() [3/10] . . . 1409
30.320.2.80 fdt_set_hdr_() [4/10] . . . 1409
30.320.2.81 fdt_set_hdr_() [5/10] . . . 1409
30.320.2.82 fdt_set_hdr_() [6/10] . . . 1409
30.320.2.83 fdt_set_hdr_() [7/10] . . . 1409
30.320.2.84 fdt_set_hdr_() [8/10] . . . 1409
30.320.2.85 fdt_set_hdr_() [9/10] . . . 1409
30.320.2.86 fdt_set_hdr_() [10/10] . . . 1409
30.320.2.87 fdt_set_name() . . . 1409
30.320.2.88 fdt_setprop() . . . 1410
30.320.2.89 fdt_setprop_cell() . . . 1410
30.320.2.90 fdt_setprop_inplace() . . . 1411
30.320.2.91 fdt_setprop_inplace_cell() . . . 1411
30.320.2.92 fdt_setprop_inplace_namelen_partial() . . . 1411
30.320.2.93 fdt_setprop_inplace_u32() . . . 1412
30.320.2.94 fdt_setprop_inplace_u64() . . . 1412
30.320.2.95 fdt_setprop_placeholder() . . . 1413
30.320.2.96 fdt_setprop_u32() . . . 1413

30.320.2.97 fdt_setprop_u64() . . . 1414
30.320.2.98 fdt_size_cells() . . . 1414
30.320.2.99 fdt_strerror() . . . 1415
30.320.2.100 fdt_string() . . . 1415
30.320.2.101 fdt_stringlist_contains() . . . 1416
30.320.2.102 fdt_stringlist_count() . . . 1417
30.320.2.103 fdt_stringlist_get() . . . 1418
30.320.2.104 fdt_stringlist_search() . . . 1418
30.320.2.105 fdt_subnode_offset() . . . 1419
30.320.2.106 fdt_subnode_offset_namelen() . . . 1419
30.320.2.107 fdt_supernode_atdepth_offset() . . . 1420
30.321 libfdt.h . . . 1421
30.322 lib/utils/libfdt/libfdt_env.h File Reference . . . 1428
30.322.1 Macro Definition Documentation . . . 1429
30.322.1.1 FDT_BITWISE . . . 1429
30.322.1.2 FDT_FORCE . . . 1429
30.322.1.3 INT32_MAX . . . 1429
30.322.1.4 INT_MAX . . . 1429
30.322.1.5 memchr . . . 1429
30.322.1.6 memcmp . . . 1429
30.322.1.7 memcpy . . . 1429
30.322.1.8 memmove . . . 1429
30.322.1.9 memset . . . 1429
30.322.1.10 strchr . . . 1430
30.322.1.11 strcmp . . . 1430
30.322.1.12 strcpy . . . 1430
30.322.1.13 strlen . . . 1430
30.322.1.14 strncmp . . . 1430
30.322.1.15 strncpy . . . 1430
30.322.1.16 strnlen . . . 1430
30.322.1.17 strrchr . . . 1430
30.322.1.18 UINT32_MAX . . . 1430
30.322.1.19 UINT_MAX . . . 1430
30.322.2 Typedef Documentation . . . 1430
30.322.2.1 fdt16_t . . . 1430
30.322.2.2 fdt32_t . . . 1430
30.322.2.3 fdt64_t . . . 1430
30.322.3 Function Documentation . . . 1430
30.322.3.1 cpu_to_fdt16() . . . 1430
30.322.3.2 cpu_to_fdt32() . . . 1431
30.322.3.3 cpu_to_fdt64() . . . 1431
30.322.3.4 fdt16_to_cpu() . . . 1431

30.322.3.5 fdt32_to_cpu() . . . 1431
30.322.3.6 fdt64_to_cpu() . . . 1432
30.323 libfdt_env.h . . . 1433
30.324 lib/utils/libfdt/libfdt_internal.h File Reference . . . 1434
30.324.1 Macro Definition Documentation . . . 1435
30.324.1.1 can_assume . . . 1435
30.324.1.2 FDT_ALIGN . . . 1435
30.324.1.3 FDT_ASSUME_MASK . . . 1435
30.324.1.4 FDT_RO_PROBE . . . 1435
30.324.1.5 FDT_SW_MAGIC . . . 1435
30.324.1.6 FDT_TAGALIGN . . . 1435
30.324.2 Enumeration Type Documentation . . . 1435
30.324.2.1 anonymous enum . . . 1435
30.324.3 Function Documentation . . . 1436
30.324.3.1 can_assume_() . . . 1436
30.324.3.2 fdt32_ld_() . . . 1436
30.324.3.3 fdt64_ld_() . . . 1437
30.324.3.4 fdt_check_node_offset_() . . . 1437
30.324.3.5 fdt_check_prop_offset_() . . . 1438
30.324.3.6 fdt_find_string_() . . . 1439
30.324.3.7 fdt_mem_rsv_() . . . 1440
30.324.3.8 fdt_mem_rsv_w_() . . . 1440
30.324.3.9 fdt_node_end_offset_() . . . 1441
30.324.3.10 fdt_offset_ptr_() . . . 1441
30.324.3.11 fdt_offset_ptr_w_() . . . 1442
30.324.3.12 fdt_ro_probe_() . . . 1443
30.325 libfdt_internal.h . . . 1443
30.326 lib/utils/libquad/divdi3.c File Reference . . . 1445
30.326.1 Function Documentation . . . 1446
30.326.1.1 __divdi3() . . . 1446
30.326.1.2 __FBSDID() . . . 1446
30.327 lib/utils/libquad/include/limits.h File Reference . . . 1447
30.327.1 Macro Definition Documentation . . . 1447
30.327.1.1 CHAR_BIT . . . 1447
30.328 limits.h . . . 1447
30.329 lib/utils/libquad/include/sys/cdefs.h File Reference . . . 1447
30.329.1 Macro Definition Documentation . . . 1447
30.329.1.1 __FBSDID . . . 1447
30.330 cdefs.h . . . 1448
30.331 lib/utils/libquad/include/sys/types.h File Reference . . . 1448
30.331.1 Macro Definition Documentation . . . 1449
30.331.1.1 _QUAD_HIGHWORD . . . 1449

30.331.1.2 _QUAD_LOWWORD . . . 1449
30.331.2 Typedef Documentation . . . 1449
30.331.2.1 quad_t . . . 1449
30.331.2.2 u_long . . . 1449
30.331.2.3 u_quad_t . . . 1449
30.332 types.h . . . 1449
30.333 lib/utils/libquad/moddi3.c File Reference . . . 1449
30.333.1 Function Documentation . . . 1450
30.333.1.1 __FBSDID() . . . 1450
30.333.1.2 __moddi3() . . . 1450
30.334 lib/utils/libquad/qdivrem.c File Reference . . . 1450
30.334.1 Macro Definition Documentation . . . 1451
30.334.1.1 B . . . 1451
30.334.1.2 COMBINE . . . 1451
30.334.2 Typedef Documentation . . . 1451
30.334.2.1 digit . . . 1451
30.334.3 Function Documentation . . . 1452
30.334.3.1 __FBSDID() . . . 1452
30.334.3.2 __qdivrem() . . . 1452
30.334.3.3 shl() . . . 1452
30.335 lib/utils/libquad/quad.h File Reference . . . 1453
30.335.1 Macro Definition Documentation . . . 1454
30.335.1.1 H . . . 1454
30.335.1.2 HALF_BITS . . . 1454
30.335.1.3 HHALF . . . 1454
30.335.1.4 L . . . 1454
30.335.1.5 LHALF . . . 1454
30.335.1.6 LHUP . . . 1454
30.335.1.7 LONG_BITS . . . 1455
30.335.1.8 QUAD_BITS . . . 1455
30.335.2 Typedef Documentation . . . 1455
30.335.2.1 qshift_t . . . 1455
30.335.3 Function Documentation . . . 1455
30.335.3.1 __cmpdi2() . . . 1455
30.335.3.2 __divdi3() . . . 1455
30.335.3.3 __moddi3() . . . 1455
30.335.3.4 __qdivrem() . . . 1456
30.335.3.5 __ucmpdi2() . . . 1456
30.335.3.6 __udivdi3() . . . 1456
30.335.3.7 __umoddi3() . . . 1457
30.336 quad.h . . . 1457
30.337 lib/utils/libquad/udivdi3.c File Reference . . . 1458

30.337.1 Function Documentation . . . 1459
30.337.1.1 __FBSDID() . . . 1459
30.337.1.2 __udivdi3() . . . 1459
30.338 lib/utils/libquad/umoddi3.c File Reference . . . 1459
30.338.1 Function Documentation . . . 1460
30.338.1.1 __FBSDID() . . . 1460
30.338.1.2 __umoddi3() . . . 1460
30.339 lib/utils/mailbox/fdt_mailbox.c File Reference . . . 1461
30.339.1 Function Documentation . . . 1461
30.339.1.1 fdt_mailbox_request_chan() . . . 1461
30.339.1.2 fdt_mailbox_simple_xlate() . . . 1462
30.339.1.3 fdt_mbox_controller_find() . . . 1462
30.339.2 Variable Documentation . . . 1463
30.339.2.1 fdt_mailbox_drivers . . . 1463
30.340 lib/utils/mailbox/fdt_mailbox_rpmi_shmem.c File Reference . . . 1463
30.340.1 Macro Definition Documentation . . . 1464
30.340.1.1 GET_DLEN . . . 1464
30.340.1.2 GET_FLAGS . . . 1465
30.340.1.3 GET_MESSAGE_ID . . . 1465
30.340.1.4 GET_MESSAGE_TYPE . . . 1465
30.340.1.5 GET_SERVICE_ID . . . 1465
30.340.1.6 GET_SERVICEGROUP_ID . . . 1465
30.340.1.7 GET_TOKEN . . . 1465
30.340.1.8 MAKE_MESSAGE_ID . . . 1465
30.340.1.9 RPMI_BASE_VERSION_MIN . . . 1466
30.340.1.10 to_srvgrp_chan . . . 1466
30.340.2 Enumeration Type Documentation . . . 1466
30.340.2.1 rpmi_queue_idx . . . 1466
30.340.2.2 rpmi_queue_type . . . 1466
30.340.2.3 rpmi_reg_idx . . . 1466
30.340.3 Function Documentation . . . 1466
30.340.3.1 __smq_queue_empty() . . . 1466
30.340.3.2 __smq_queue_full() . . . 1467
30.340.3.3 __smq_rx() . . . 1467
30.340.3.4 __smq_tx() . . . 1467
30.340.3.5 rpmi_get_platform_info() . . . 1468
30.340.3.6 rpmi_shmem_mbox_free_chan() . . . 1469
30.340.3.7 rpmi_shmem_mbox_init() . . . 1469
30.340.3.8 rpmi_shmem_mbox_request_chan() . . . 1470
30.340.3.9 rpmi_shmem_mbox_xfer() . . . 1471
30.340.3.10 rpmi_shmem_transport_init() . . . 1471
30.340.3.11 smq_base_get_two_u32() . . . 1472

30.340.3.12 smq_rx() . . . 1473
30.340.3.13 smq_tx() . . . 1473
30.340.4 Variable Documentation . . . 1474
30.340.4.1 __packed . . . 1474
30.340.4.2 fdt_mailbox_rpmi_shmem . . . 1474
30.340.4.3 rpmi_shmem_mbox_match . . . 1474
30.341 lib/utils/mailbox/mailbox.c File Reference . . . 1474
30.341.1 Function Documentation . . . 1475
30.341.1.1 mbox_chan_get_attribute() . . . 1475
30.341.1.2 mbox_chan_set_attribute() . . . 1475
30.341.1.3 mbox_chan_xfer() . . . 1475
30.341.1.4 mbox_controller_add() . . . 1476
30.341.1.5 mbox_controller_find() . . . 1477
30.341.1.6 mbox_controller_free_chan() . . . 1477
30.341.1.7 mbox_controller_remove() . . . 1477
30.341.1.8 mbox_controller_request_chan() . . . 1478
30.341.1.9 SBI_LIST_HEAD() . . . 1479
30.342 lib/utils/mailbox/rpmi_mailbox.c File Reference . . . 1479
30.342.1 Function Documentation . . . 1480
30.342.1.1 rpmi_normal_request_with_status() . . . 1480
30.342.1.2 rpmi_posted_request() . . . 1481
30.342.1.3 rpmi_xlate_error() . . . 1482
30.343 lib/utils/mpxy/fdt_mpxy.c File Reference . . . 1482
30.343.1 Function Documentation . . . 1483
30.343.1.1 fdt_mpxy_init() . . . 1483
30.343.2 Variable Documentation . . . 1483
30.343.2.1 fdt_mpxy_drivers . . . 1483
30.344 lib/utils/mpxy/fdt_mpxy_rpmi_mbox.c File Reference . . . 1484
30.344.1 Macro Definition Documentation . . . 1485
30.344.1.1 assert_field_offset . . . 1485
30.344.1.2 attr_id2index . . . 1485
30.344.1.3 RPMI_MAJOR_VER . . . 1485
30.344.1.4 RPMI_MINOR_VER . . . 1485
30.344.2 Enumeration Type Documentation . . . 1485
30.344.2.1 mpxy_msgprot_rpmi_attr_id . . . 1485
30.344.3 Function Documentation . . . 1485
30.344.3.1 __mpxy_mbox_send_message() . . . 1485
30.344.3.2 assert_field_offset() . . . 1486
30.344.3.3 mpxy_check_write_attr() . . . 1486
30.344.3.4 mpxy_copy_attrs() . . . 1487
30.344.3.5 mpxy_find_rpmi_srvid() . . . 1487
30.344.3.6 mpxy_mbox_get_notifications() . . . 1487

30.344.3.7 mpxy_mbox_init() . . . 1487
30.344.3.8 mpxy_mbox_read_attributes() . . . 1488
30.344.3.9 mpxy_mbox_send_message_withoutresp() . . . 1488
30.344.3.10 mpxy_mbox_send_message_withresp() . . . 1489
30.344.3.11 mpxy_mbox_write_attributes() . . . 1489
30.344.3.12 mpxy_write_attr() . . . 1490
30.344.4 Variable Documentation . . . 1490
30.344.4.1 clock_data . . . 1490
30.344.4.2 clock_services . . . 1490
30.344.4.3 fdt_mpxy_rpmi_mbox . . . 1491
30.344.4.4 mpxy_mbox_match . . . 1491
30.345 lib/utils/regmap/fdt_regmap.c File Reference . . . 1491
30.345.1 Function Documentation . . . 1491
30.345.1.1 fdt_regmap_find() . . . 1491
30.345.1.2 fdt_regmap_get() . . . 1492
30.345.1.3 fdt_regmap_get_by_phandle() . . . 1493
30.345.2 Variable Documentation . . . 1493
30.345.2.1 fdt_regmap_drivers . . . 1493
30.346 lib/utils/regmap/fdt_regmap_syscon.c File Reference . . . 1493
30.346.1 Macro Definition Documentation . . . 1494
30.346.1.1 to_syscon_regmap . . . 1494
30.346.2 Enumeration Type Documentation . . . 1495
30.346.2.1 syscon_regmap_endian . . . 1495
30.346.3 Function Documentation . . . 1495
30.346.3.1 regmap_syscon_init() . . . 1495
30.346.3.2 regmap_syscon_read_16() . . . 1495
30.346.3.3 regmap_syscon_read_32() . . . 1496
30.346.3.4 regmap_syscon_read_8() . . . 1496
30.346.3.5 regmap_syscon_read_be16() . . . 1496
30.346.3.6 regmap_syscon_read_be32() . . . 1497
30.346.3.7 regmap_syscon_read_le16() . . . 1497
30.346.3.8 regmap_syscon_read_le32() . . . 1497
30.346.3.9 regmap_syscon_write_16() . . . 1498
30.346.3.10 regmap_syscon_write_32() . . . 1498
30.346.3.11 regmap_syscon_write_8() . . . 1498
30.346.3.12 regmap_syscon_write_be16() . . . 1499
30.346.3.13 regmap_syscon_write_be32() . . . 1499
30.346.3.14 regmap_syscon_write_le16() . . . 1499
30.346.3.15 regmap_syscon_write_le32() . . . 1500
30.346.4 Variable Documentation . . . 1500
30.346.4.1 fdt_regmap_syscon . . . 1500
30.346.4.2 regmap_syscon_match . . . 1500

30.347 lib/utils/regmap/regmap.c File Reference . . . 1500
30.347.1 Function Documentation . . . 1501
30.347.1.1 regmap_add() . . . 1501
30.347.1.2 regmap_find() . . . 1502
30.347.1.3 regmap_read() . . . 1502
30.347.1.4 regmap_reg_addr() . . . 1503
30.347.1.5 regmap_reg_valid() . . . 1503
30.347.1.6 regmap_remove() . . . 1503
30.347.1.7 regmap_update_bits() . . . 1504
30.347.1.8 regmap_write() . . . 1504
30.347.1.9 SBI_LIST_HEAD() . . . 1505
30.348 lib/utils/reset/fdt_reset.c File Reference . . . 1505
30.348.1 Function Documentation . . . 1506
30.348.1.1 fdt_reset_init() . . . 1506
30.348.2 Variable Documentation . . . 1506
30.348.2.1 fdt_reset_drivers . . . 1506
30.349 lib/utils/reset/fdt_reset_atcwdt200.c File Reference . . . 1506
30.349.1 Macro Definition Documentation . . . 1507
30.349.1.1 ATCWDT200_WP_NUM . . . 1507
30.349.1.2 CLK_PCLK . . . 1507
30.349.1.3 CTRL_REG . . . 1507
30.349.1.4 INT_CLK_1024 . . . 1507
30.349.1.5 INT_CLK_16384 . . . 1507
30.349.1.6 INT_CLK_2048 . . . 1507
30.349.1.7 INT_CLK_256 . . . 1508
30.349.1.8 INT_CLK_32768 . . . 1508
30.349.1.9 INT_CLK_4096 . . . 1508
30.349.1.10 INT_CLK_64 . . . 1508
30.349.1.11 INT_CLK_8192 . . . 1508
30.349.1.12 INT_EN . . . 1508
30.349.1.13 INT_TIME_MSK . . . 1508
30.349.1.14 INT_TIME_OFF . . . 1508
30.349.1.15 RST_CLK_1024 . . . 1508
30.349.1.16 RST_CLK_128 . . . 1508
30.349.1.17 RST_CLK_256 . . . 1508
30.349.1.18 RST_CLK_512 . . . 1508
30.349.1.19 RST_EN . . . 1508
30.349.1.20 RST_TIME_MSK . . . 1508
30.349.1.21 RST_TIME_OFF . . . 1508
30.349.1.22 WDT_EN . . . 1509
30.349.1.23 WREN_REG . . . 1509
30.349.2 Function Documentation . . . 1509

30.349.2.1 ae350_system_reset() . . . 1509
30.349.2.2 ae350_system_reset_check() . . . 1509
30.349.2.3 atcwdt200_reset_init() . . . 1509
30.349.3 Variable Documentation . . . 1509
30.349.3.1 atcwdt200_reset . . . 1509
30.349.3.2 atcwdt200_reset_match . . . 1510
30.349.3.3 fdt_reset_atcwdt200 . . . 1510
30.349.3.4 smu . . . 1510
30.349.3.5 wdt_addr . . . 1510
30.350 lib/utils/reset/fdt_reset_gpio.c File Reference . . . 1510
30.350.1 Function Documentation . . . 1511
30.350.1.1 gpio_reset_exec() . . . 1511
30.350.1.2 gpio_reset_get() . . . 1511
30.350.1.3 gpio_reset_init() . . . 1512
30.350.1.4 gpio_system_poweroff() . . . 1512
30.350.1.5 gpio_system_poweroff_check() . . . 1513
30.350.1.6 gpio_system_restart() . . . 1513
30.350.1.7 gpio_system_restart_check() . . . 1513
30.350.2 Variable Documentation . . . 1513
30.350.2.1 fdt_poweroff_gpio . . . 1513
30.350.2.2 fdt_reset_gpio . . . 1514
30.350.2.3 gpio_poweroff . . . 1514
30.350.2.4 gpio_poweroff_match . . . 1514
30.350.2.5 gpio_reset_match . . . 1514
30.350.2.6 gpio_restart . . . 1514
30.350.2.7 poweroff . . . 1514
30.350.2.8 restart . . . 1514
30.351 lib/utils/reset/fdt_reset_htif.c File Reference . . . 1515
30.351.1 Function Documentation . . . 1515
30.351.1.1 htif_reset_init() . . . 1515
30.351.2 Variable Documentation . . . 1516
30.351.2.1 fdt_reset_htif . . . 1516
30.351.2.2 htif_reset_match . . . 1516
30.352 lib/utils/reset/fdt_reset_rpmi.c File Reference . . . 1516
30.352.1 Function Documentation . . . 1517
30.352.1.1 rpmi_do_system_reset() . . . 1517
30.352.1.2 rpmi_reset_init() . . . 1517
30.352.1.3 rpmi_system_reset() . . . 1518
30.352.1.4 rpmi_system_reset_check() . . . 1518
30.352.1.5 rpmi_system_reset_type_check() . . . 1518
30.352.2 Variable Documentation . . . 1519
30.352.2.1 fdt_reset_rpmi . . . 1519

30.352.2.2 rpmi_reset_dev . . . 1519
30.352.2.3 rpmi_reset_match . . . 1519
30.352.2.4 sysreset_ctx . . . 1519
30.353 lib/utils/reset/fdt_reset_sg2042_hwmon_mcu.c File Reference . . . 1519
30.353.1 Macro Definition Documentation . . . 1520
30.353.1.1 CMD_POWEROFF . . . 1520
30.353.1.2 CMD_REBOOT . . . 1520
30.353.1.3 CMD_RESET . . . 1520
30.353.1.4 MANGO_BOARD_TYPE_MASK . . . 1520
30.353.1.5 REG_BOARD_TYPE . . . 1521
30.353.1.6 REG_CMD . . . 1521
30.353.2 Function Documentation . . . 1521
30.353.2.1 sg2042_mcu_reset() . . . 1521
30.353.2.2 sg2042_mcu_reset_check() . . . 1521
30.353.2.3 sg2042_mcu_reset_check_board() . . . 1521
30.353.2.4 sg2042_mcu_reset_init() . . . 1522
30.353.3 Variable Documentation . . . 1522
30.353.3.1 fdt_reset_sg2042_mcu . . . 1522
30.353.3.2 mcu_adapter . . . 1522
30.353.3.3 mcu_reg . . . 1522
30.353.3.4 sg2042_mcu_reset_device . . . 1522
30.353.3.5 sg2042_mcu_reset_match . . . 1522
30.354 lib/utils/reset/fdt_reset_sunxi_wdt.c File Reference . . . 1522
30.354.1 Macro Definition Documentation . . . 1523
30.354.1.1 WDT_KEY_VAL . . . 1523
30.354.1.2 WDT_MODE_REG . . . 1523
30.354.1.3 WDT_SOFT_RST_EN . . . 1523
30.354.1.4 WDT_SOFT_RST_REG . . . 1523
30.354.2 Function Documentation . . . 1524
30.354.2.1 sunxi_wdt_reset_init() . . . 1524
30.354.2.2 sunxi_wdt_system_reset() . . . 1524
30.354.2.3 sunxi_wdt_system_reset_check() . . . 1524
30.354.3 Variable Documentation . . . 1524
30.354.3.1 fdt_reset_sunxi_wdt . . . 1524
30.354.3.2 sunxi_wdt_base . . . 1524
30.354.3.3 sunxi_wdt_reset . . . 1524
30.354.3.4 sunxi_wdt_reset_match . . . 1524
30.355 lib/utils/reset/fdt_reset_syscon.c File Reference . . . 1525
30.355.1 Function Documentation . . . 1526
30.355.1.1 syscon_do_poweroff() . . . 1526
30.355.1.2 syscon_do_reboot() . . . 1526
30.355.1.3 syscon_poweroff_check() . . . 1526

30.355.1.4 syscon_reboot_check() . . . 1526
30.355.1.5 syscon_reset_exec() . . . 1527
30.355.1.6 syscon_reset_get() . . . 1527
30.355.1.7 syscon_reset_init() . . . 1528
30.355.2 Variable Documentation . . . 1528
30.355.2.1 fdt_syscon_poweroff . . . 1528
30.355.2.2 fdt_syscon_reboot . . . 1528
30.355.2.3 poweroff . . . 1529
30.355.2.4 reboot . . . 1529
30.355.2.5 syscon_poweroff . . . 1529
30.355.2.6 syscon_poweroff_match . . . 1529
30.355.2.7 syscon_reboot . . . 1529
30.355.2.8 syscon_reboot_match . . . 1529
30.356 lib/utils/serial/cadence-uart.c File Reference . . . 1529
30.356.1 Macro Definition Documentation . . . 1530
30.356.1.1 UART_BRGR_CD_CLKDIVISOR . . . 1530
30.356.1.2 UART_CSR_REMPTY . . . 1531
30.356.1.3 UART_CSR_TFUL . . . 1531
30.356.1.4 UART_CTRL_RXDIS . . . 1531
30.356.1.5 UART_CTRL_RXEN . . . 1531
30.356.1.6 UART_CTRL_RXRES . . . 1531
30.356.1.7 UART_CTRL_TXDIS . . . 1531
30.356.1.8 UART_CTRL_TXEN . . . 1531
30.356.1.9 UART_CTRL_TXRES . . . 1531
30.356.1.10 UART_MODE_PAR_NONE . . . 1531
30.356.1.11 UART_REG_BDIVR . . . 1531
30.356.1.12 UART_REG_BRGR . . . 1531
30.356.1.13 UART_REG_CSR . . . 1531
30.356.1.14 UART_REG_CTRL . . . 1531
30.356.1.15 UART_REG_IDR . . . 1531
30.356.1.16 UART_REG_MODE . . . 1531
30.356.1.17 UART_REG_RFIFO_TFIFO . . . 1532
30.356.2 Function Documentation . . . 1532
30.356.2.1 cadence_uart_getc() . . . 1532
30.356.2.2 cadence_uart_init() . . . 1532
30.356.2.3 cadence_uart_putc() . . . 1532
30.356.2.4 get_reg() . . . 1533
30.356.2.5 set_reg() . . . 1533
30.356.2.6 uart_min_clk_divisor() . . . 1533
30.356.3 Variable Documentation . . . 1534
30.356.3.1 cadence_console . . . 1534
30.356.3.2 uart_base . . . 1534

30.356.3.3 uart_baudrate . . . 1534
30.356.3.4 uart_in_freq . . . 1534
30.357 lib/utils/serial/fdt_serial.c File Reference . . . 1534
30.357.1 Function Documentation . . . 1535
30.357.1.1 fdt_serial_init() . . . 1535
30.357.2 Variable Documentation . . . 1535
30.357.2.1 fdt_serial_drivers . . . 1535
30.358 lib/utils/serial/fdt_serial_cadence.c File Reference . . . 1535
30.358.1 Function Documentation . . . 1536
30.358.1.1 serial_cadence_init() . . . 1536
30.358.2 Variable Documentation . . . 1537
30.358.2.1 fdt_serial_cadence . . . 1537
30.358.2.2 serial_cadence_match . . . 1537
30.359 lib/utils/serial/fdt_serial_gaisler.c File Reference . . . 1537
30.359.1 Function Documentation . . . 1538
30.359.1.1 serial_gaisler_init() . . . 1538
30.359.2 Variable Documentation . . . 1538
30.359.2.1 fdt_serial_gaisler . . . 1538
30.359.2.2 serial_gaisler_match . . . 1538
30.360 lib/utils/serial/fdt_serial_htif.c File Reference . . . 1538
30.360.1 Function Documentation . . . 1539
30.360.1.1 serial_htif_init() . . . 1539
30.360.2 Variable Documentation . . . 1540
30.360.2.1 fdt_serial_htif . . . 1540
30.360.2.2 serial_htif_match . . . 1540
30.361 lib/utils/serial/fdt_serial_litex.c File Reference . . . 1540
30.361.1 Function Documentation . . . 1541
30.361.1.1 serial_litex_init() . . . 1541
30.361.2 Variable Documentation . . . 1541
30.361.2.1 fdt_serial_litex . . . 1541
30.361.2.2 serial_litex_match . . . 1541
30.362 lib/utils/serial/fdt_serial_renesas_scif.c File Reference . . . 1541
30.362.1 Function Documentation . . . 1542
30.362.1.1 serial_renesas_scif_init() . . . 1542
30.362.2 Variable Documentation . . . 1543
30.362.2.1 fdt_serial_renesas_scif . . . 1543
30.362.2.2 serial_renesas_scif_match . . . 1543
30.363 lib/utils/serial/fdt_serial_shakti.c File Reference . . . 1543
30.363.1 Function Documentation . . . 1544
30.363.1.1 serial_shakti_init() . . . 1544
30.363.2 Variable Documentation . . . 1544
30.363.2.1 fdt_serial_shakti . . . 1544

30.363.2.2 serial_shakti_match . . . 1544
30.364 lib/utils/serial/fdt_serial_sifive.c File Reference . . . 1544
30.364.1 Function Documentation . . . 1545
30.364.1.1 serial_sifive_init() . . . 1545
30.364.2 Variable Documentation . . . 1546
30.364.2.1 fdt_serial_sifive . . . 1546
30.364.2.2 serial_sifive_match . . . 1546
30.365 lib/utils/serial/fdt_serial_uart8250.c File Reference . . . 1546
30.365.1 Function Documentation . . . 1547
30.365.1.1 serial_uart8250_init() . . . 1547
30.365.2 Variable Documentation . . . 1547
30.365.2.1 fdt_serial_uart8250 . . . 1547
30.365.2.2 serial_uart8250_match . . . 1547
30.366 lib/utils/serial/fdt_serial_xlnx_uartlite.c File Reference . . . 1547
30.366.1 Function Documentation . . . 1548
30.366.1.1 serial_xlnx_uartlite_init() . . . 1548
30.366.2 Variable Documentation . . . 1549
30.366.2.1 fdt_serial_xlnx_uartlite . . . 1549
30.366.2.2 serial_xlnx_uartlite_match . . . 1549
30.367 lib/utils/serial/gaisler-uart.c File Reference . . . 1549
30.367.1 Macro Definition Documentation . . . 1550
30.367.1.1 UART_CTRL_DB . . . 1550
30.367.1.2 UART_CTRL_FL . . . 1550
30.367.1.3 UART_CTRL_RE . . . 1550
30.367.1.4 UART_CTRL_TE . . . 1550
30.367.1.5 UART_DATA_DATA . . . 1550
30.367.1.6 UART_REG_CTRL . . . 1550
30.367.1.7 UART_REG_DATA . . . 1550
30.367.1.8 UART_REG_SCALER . . . 1550
30.367.1.9 UART_REG_STATUS . . . 1550
30.367.1.10 UART_STATUS_DATAREADY . . . 1550
30.367.1.11 UART_STATUS_FIFOFULL . . . 1550
30.367.2 Function Documentation . . . 1551
30.367.2.1 gaisler_uart_getc() . . . 1551
30.367.2.2 gaisler_uart_init() . . . 1551
30.367.2.3 gaisler_uart_putc() . . . 1551
30.367.2.4 get_reg() . . . 1552
30.367.2.5 set_reg() . . . 1552
30.367.3 Variable Documentation . . . 1553
30.367.3.1 gaisler_console . . . 1553
30.367.3.2 uart_base . . . 1553
30.368 lib/utils/serial/litex-uart.c File Reference . . . 1553

30.368.1 Macro Definition Documentation . . . 1554
30.368.1.1 UART_EV_RX . . . 1554
30.368.1.2 UART_REG_EV_ENABLE . . . 1554
30.368.1.3 UART_REG_EV_PENDING . . . 1554
30.368.1.4 UART_REG_EV_STATUS . . . 1554
30.368.1.5 UART_REG_RXEMPTY . . . 1554
30.368.1.6 UART_REG_RXTX . . . 1554
30.368.1.7 UART_REG_TXFULL . . . 1554
30.368.2 Function Documentation . . . 1554
30.368.2.1 get_reg() . . . 1554
30.368.2.2 litex_uart_getc() . . . 1555
30.368.2.3 litex_uart_init() . . . 1555
30.368.2.4 litex_uart_putc() . . . 1555
30.368.2.5 set_reg() . . . 1556
30.368.3 Variable Documentation . . . 1556
30.368.3.1 litex_console . . . 1556
30.368.3.2 uart_base . . . 1556
30.369 lib/utils/serial/renesas_scif.c File Reference . . . 1556
30.369.1 Macro Definition Documentation . . . 1558
30.369.1.1 SCBRR_VALUE . . . 1558
30.369.1.2 SCIF_FCR_RFRST . . . 1558
30.369.1.3 SCIF_FCR_RST_ASSRT_RFTF . . . 1558
30.369.1.4 SCIF_FCR_RST_NGATE_RFTF . . . 1558
30.369.1.5 SCIF_FCR_TFRST . . . 1558
30.369.1.6 SCIF_FSR_BRK . . . 1558
30.369.1.7 SCIF_FSR_DR . . . 1558
30.369.1.8 SCIF_FSR_ER . . . 1558
30.369.1.9 SCIF_FSR_TDFE . . . 1558
30.369.1.10 SCIF_FSR_TEND . . . 1558
30.369.1.11 SCIF_FSR_TXD_CHK . . . 1558
30.369.1.12 SCIF_REG_8BIT . . . 1558
30.369.1.13 SCIF_REG_BRR . . . 1558
30.369.1.14 SCIF_REG_FCR . . . 1559
30.369.1.15 SCIF_REG_FSR . . . 1559
30.369.1.16 SCIF_REG_FTDR . . . 1559
30.369.1.17 SCIF_REG_LSR . . . 1559
30.369.1.18 SCIF_REG_SCR . . . 1559
30.369.1.19 SCIF_REG_SEMR . . . 1559
30.369.1.20 SCIF_REG_SMR . . . 1559
30.369.1.21 SCIF_SCR_RCV_TRN_DIS . . . 1559
30.369.1.22 SCIF_SCR_RCV_TRN_EN . . . 1559
30.369.1.23 SCIF_SCR_RE . . . 1559

30.369.1.24 SCIF_SCR_TE . . . 1559
30.369.1.25 SCIF_SEMR_MDDRS . . . 1559
30.369.2 Function Documentation . . . 1559
30.369.2.1 get_reg() . . . 1559
30.369.2.2 renesas_scif_init() . . . 1560
30.369.2.3 renesas_scif_putc() . . . 1560
30.369.2.4 set_reg() . . . 1561
30.369.3 Variable Documentation . . . 1561
30.369.3.1 renesas_scif_console . . . 1561
30.369.3.2 scif_base . . . 1561
30.370 lib/utils/serial/semihosting.c File Reference . . . 1561
30.370.1 Macro Definition Documentation . . . 1562
30.370.1.1 SYSERRNO . . . 1562
30.370.1.2 SYSOPEN . . . 1562
30.370.1.3 SYSREAD . . . 1563
30.370.1.4 SYSREADC . . . 1563
30.370.1.5 SYSWRITE . . . 1563
30.370.1.6 SYSWRITEC . . . 1563
30.370.2 Function Documentation . . . 1563
30.370.2.1 semihosting_enabled() . . . 1563
30.370.2.2 semihosting_errno() . . . 1563
30.370.2.3 semihosting_getc() . . . 1563
30.370.2.4 semihosting_init() . . . 1564
30.370.2.5 semihosting_open() . . . 1564
30.370.2.6 semihosting_puts() . . . 1565
30.370.2.7 semihosting_read() . . . 1565
30.370.2.8 semihosting_trap() . . . 1565
30.370.2.9 semihosting_write() . . . 1566
30.370.3 Variable Documentation . . . 1566
30.370.3.1 _semihosting_enabled . . . 1566
30.370.3.2 semihosting_console . . . 1566
30.370.3.3 semihosting_infd . . . 1567
30.370.3.4 semihosting_outfd . . . 1567
30.370.3.5 try_semihosting . . . 1567
30.371 lib/utils/serial/shakti-uart.c File Reference . . . 1567
30.371.1 Macro Definition Documentation . . . 1568
30.371.1.1 REG_BAUD . . . 1568
30.371.1.2 REG_CONTROL . . . 1568
30.371.1.3 REG_DELAY . . . 1568
30.371.1.4 REG_INT_EN . . . 1568
30.371.1.5 REG_IQ_CYCLES . . . 1568
30.371.1.6 REG_RX . . . 1568

30.371.1.7 REG_RX_THRES . . . 1568
30.371.1.8 REG_STATUS . . . 1568
30.371.1.9 REG_TX . . . 1568
30.371.1.10 UART_RX_FULL . . . 1568
30.371.1.11 UART_RX_NOT_EMPTY . . . 1568
30.371.1.12 UART_TX_FULL . . . 1568
30.371.2 Function Documentation . . . 1568
30.371.2.1 shakti_uart_getc() . . . 1568
30.371.2.2 shakti_uart_init() . . . 1569
30.371.2.3 shakti_uart_putc() . . . 1569
30.371.3 Variable Documentation . . . 1569
30.371.3.1 shakti_console . . . 1569
30.371.3.2 uart_base . . . 1569
30.372 lib/utils/serial/sifive-uart.c File Reference . . . 1569
30.372.1 Macro Definition Documentation . . . 1570
30.372.1.1 UART_REG_DIV . . . 1570
30.372.1.2 UART_REG_IE . . . 1571
30.372.1.3 UART_REG_IP . . . 1571
30.372.1.4 UART_REG_RXCTRL . . . 1571
30.372.1.5 UART_REG_RXFIFO . . . 1571
30.372.1.6 UART_REG_TXCTRL . . . 1571
30.372.1.7 UART_REG_TXFIFO . . . 1571
30.372.1.8 UART_RXCTRL_RXEN . . . 1571
30.372.1.9 UART_RXFIFO_DATA . . . 1571
30.372.1.10 UART_RXFIFO_EMPTY . . . 1571
30.372.1.11 UART_TXCTRL_TXEN . . . 1571
30.372.1.12 UART_TXFIFO_FULL . . . 1571
30.372.2 Function Documentation . . . 1571
30.372.2.1 get_reg() . . . 1571
30.372.2.2 set_reg() . . . 1572
30.372.2.3 sifive_uart_getc() . . . 1572
30.372.2.4 sifive_uart_init() . . . 1572
30.372.2.5 sifive_uart_putc() . . . 1573
30.372.2.6 uart_min_clk_divisor() . . . 1573
30.372.3 Variable Documentation . . . 1574
30.372.3.1 sifive_console . . . 1574
30.372.3.2 uart_base . . . 1574
30.372.3.3 uart_baudrate . . . 1574
30.372.3.4 uart_in_freq . . . 1574
30.373 lib/utils/serial/uart8250.c File Reference . . . 1574
30.373.1 Macro Definition Documentation . . . 1575
30.373.1.1 UART_DLL_OFFSET . . . 1575

30.373.1.2 UART_DLM_OFFSET . . . 1575
30.373.1.3 UART_FCR_OFFSET . . . 1575
30.373.1.4 UART_IER_OFFSET . . . 1576
30.373.1.5 UART_IIR_OFFSET . . . 1576
30.373.1.6 UART_LCR_OFFSET . . . 1576
30.373.1.7 UART_LSR_BI . . . 1576
30.373.1.8 UART_LSR_BRK_ERROR_BITS . . . 1576
30.373.1.9 UART_LSR_DR . . . 1576
30.373.1.10 UART_LSR_FE . . . 1576
30.373.1.11 UART_LSR_FIFOE . . . 1576
30.373.1.12 UART_LSR_OE . . . 1576
30.373.1.13 UART_LSR_OFFSET . . . 1576
30.373.1.14 UART_LSR_PE . . . 1576
30.373.1.15 UART_LSR_TEMT . . . 1576
30.373.1.16 UART_LSR_THRE . . . 1576
30.373.1.17 UART_MCR_OFFSET . . . 1576
30.373.1.18 UART_MDR1_OFFSET . . . 1576
30.373.1.19 UART_MSR_OFFSET . . . 1577
30.373.1.20 UART_RBR_OFFSET . . . 1577
30.373.1.21 UART_SCR_OFFSET . . . 1577
30.373.1.22 UART_THR_OFFSET . . . 1577
30.373.2 Function Documentation . . . 1577
30.373.2.1 get_reg() . . . 1577
30.373.2.2 set_reg() . . . 1577
30.373.2.3 uart8250_getc() . . . 1578
30.373.2.4 uart8250_init() . . . 1578
30.373.2.5 uart8250_putc() . . . 1578
30.373.3 Variable Documentation . . . 1579
30.373.3.1 uart8250_base . . . 1579
30.373.3.2 uart8250_baudrate . . . 1579
30.373.3.3 uart8250_console . . . 1579
30.373.3.4 uart8250_in_freq . . . 1579
30.373.3.5 uart8250_reg_shift . . . 1579
30.373.3.6 uart8250_reg_width . . . 1579
30.374 lib/utils/serial/xlnx-uartlite.c File Reference . . . 1579
30.374.1 Macro Definition Documentation . . . 1580
30.374.1.1 UART_CTRL_IE . . . 1580
30.374.1.2 UART_CTRL_OFFSET . . . 1580
30.374.1.3 UART_CTRL_RST_RX . . . 1581
30.374.1.4 UART_CTRL_RST_TX . . . 1581
30.374.1.5 UART_RX_OFFSET . . . 1581
30.374.1.6 UART_STATUS_FRAME . . . 1581

30.374.1.7 UART_STATUS_IE . . . 1581
30.374.1.8 UART_STATUS_OFFSET . . . 1581
30.374.1.9 UART_STATUS_OVERRUN . . . 1581
30.374.1.10 UART_STATUS_PARITY . . . 1581
30.374.1.11 UART_STATUS_RXFULL . . . 1581
30.374.1.12 UART_STATUS_RXVALID . . . 1581
30.374.1.13 UART_STATUS_TXEMPTY . . . 1581
30.374.1.14 UART_STATUS_TXFULL . . . 1581
30.374.1.15 UART_TX_OFFSET . . . 1581
30.374.2 Function Documentation . . . 1581
30.374.2.1 xlnx_uartlite_getc() . . . 1581
30.374.2.2 xlnx_uartlite_init() . . . 1582
30.374.2.3 xlnx_uartlite_putc() . . . 1582
30.374.3 Variable Documentation . . . 1582
30.374.3.1 xlnx_uartlite_base . . . 1582
30.374.3.2 xlnx_uartlite_console . . . 1582
30.375 lib/utils/suspend/fdt_suspend.c File Reference . . . 1582
30.375.1 Function Documentation . . . 1583
30.375.1.1 fdt_suspend_init() . . . 1583
30.375.2 Variable Documentation . . . 1583
30.375.2.1 fdt_suspend_drivers . . . 1583
30.376 lib/utils/suspend/fdt_suspend_rpmi.c File Reference . . . 1584
30.376.1 Function Documentation . . . 1584
30.376.1.1 rpmi_suspend_init() . . . 1584
30.376.1.2 rpmi_syssusp() . . . 1585
30.376.1.3 rpmi_syssusp_attrs() . . . 1585
30.376.1.4 rpmi_system_suspend() . . . 1586
30.376.1.5 rpmi_system_suspend_check() . . . 1586
30.376.2 Variable Documentation . . . 1586
30.376.2.1 fdt_suspend_rpmi . . . 1586
30.376.2.2 rpmi_suspend_dev . . . 1586
30.376.2.3 rpmi_suspend_match . . . 1586
30.376.2.4 syssusp_ctx . . . 1587
30.377 lib/utils/sys/atcsmu.c File Reference . . . 1587
30.377.1 Function Documentation . . . 1587
30.377.1.1 smu_set_command() . . . 1587
30.377.1.2 smu_set_reset_vector() . . . 1587
30.377.1.3 smu_set_wakeup_events() . . . 1588
30.377.1.4 smu_support_sleep_mode() . . . 1588
30.378 lib/utils/sys/htif.c File Reference . . . 1588
30.378.1 Macro Definition Documentation . . . 1590
30.378.1.1 FROMHOST_CMD . . . 1590

30.378.1.2 FROMHOST_DATA . . . 1590
30.378.1.3 FROMHOST_DEV . . . 1590
30.378.1.4 HTIF_CMD_BITS . . . 1590
30.378.1.5 HTIF_CMD_MASK . . . 1590
30.378.1.6 HTIF_CMD_SHIFT . . . 1590
30.378.1.7 HTIF_CONSOLE_CMD_GETC . . . 1590
30.378.1.8 HTIF_CONSOLE_CMD_PUTC . . . 1590
30.378.1.9 HTIF_DATA_BITS . . . 1590
30.378.1.10 HTIF_DATA_MASK . . . 1590
30.378.1.11 HTIF_DATA_SHIFT . . . 1590
30.378.1.12 HTIF_DEV_BITS . . . 1590
30.378.1.13 HTIF_DEV_CONSOLE . . . 1590
30.378.1.14 HTIF_DEV_MASK . . . 1590
30.378.1.15 HTIF_DEV_SHIFT . . . 1591
30.378.1.16 HTIF_DEV_SYSTEM . . . 1591
30.378.1.17 PK_SYS_write . . . 1591
30.378.1.18 TOHOST_CMD . . . 1591
30.378.2 Function Documentation . . . 1591
30.378.2.1 __attribute__() . . . 1591
30.378.2.2 __check_fromhost() . . . 1591
30.378.2.3 __read_fromhost() . . . 1592
30.378.2.4 __read_tohost() . . . 1592
30.378.2.5 __set_tohost() . . . 1592
30.378.2.6 __write_fromhost() . . . 1593
30.378.2.7 __write_tohost() . . . 1593
30.378.2.8 htif_getc() . . . 1593
30.378.2.9 htif_putc() . . . 1594
30.378.2.10 htif_serial_init() . . . 1594
30.378.2.11 htif_system_reset() . . . 1595
30.378.2.12 htif_system_reset_check() . . . 1595
30.378.2.13 htif_system_reset_init() . . . 1595
30.378.2.14 set_custom_addr() . . . 1596
30.378.3 Variable Documentation . . . 1596
30.378.3.1 htif_console . . . 1596
30.378.3.2 htif_console_buf . . . 1596
30.378.3.3 htif_custom . . . 1596
30.378.3.4 htif_fromhost . . . 1596
30.378.3.5 htif_lock . . . 1596
30.378.3.6 htif_reset . . . 1597
30.378.3.7 htif_tohost . . . 1597
30.379 lib/utils/timer/aclint_mtimer.c File Reference . . . 1597
30.379.1 Macro Definition Documentation . . . 1598

30.379.1.1 mtimer_get_hart_data_ptr . . . 1598
30.379.1.2 mtimer_set_hart_data_ptr . . . 1598
30.379.2 Function Documentation . . . 1598
30.379.2.1 aclint_mtimer_cold_init() . . . 1598
30.379.2.2 aclint_mtimer_set_reference() . . . 1598
30.379.2.3 aclint_mtimer_sync() . . . 1599
30.379.2.4 aclint_mtimer_warm_init() . . . 1599
30.379.2.5 mtimer_event_start() . . . 1600
30.379.2.6 mtimer_event_stop() . . . 1600
30.379.2.7 mtimer_time_rd32() . . . 1600
30.379.2.8 mtimer_time_rd64() . . . 1600
30.379.2.9 mtimer_time_wr32() . . . 1600
30.379.2.10 mtimer_time_wr64() . . . 1600
30.379.2.11 mtimer_value() . . . 1601
30.379.3 Variable Documentation . . . 1601
30.379.3.1 mtimer . . . 1601
30.379.3.2 mtimer_ptr_offset . . . 1601
30.380 lib/utils/timer/andes_plmt.c File Reference . . . 1601
30.380.1 Function Documentation . . . 1602
30.380.1.1 plmt_cold_timer_init() . . . 1602
30.380.1.2 plmt_timer_event_start() . . . 1602
30.380.1.3 plmt_timer_event_stop() . . . 1602
30.380.1.4 plmt_timer_value() . . . 1603
30.380.1.5 plmt_warm_timer_init() . . . 1603
30.380.2 Variable Documentation . . . 1603
30.380.2.1 plmt . . . 1603
30.380.2.2 plmt_timer . . . 1603
30.381 lib/utils/timer/fdt_timer.c File Reference . . . 1603
30.381.1 Function Documentation . . . 1604
30.381.1.1 fdt_timer_init() . . . 1604
30.381.2 Variable Documentation . . . 1604
30.381.2.1 fdt_timer_drivers . . . 1604
30.382 lib/utils/timer/fdt_timer_mtimer.c File Reference . . . 1605
30.382.1 Function Documentation . . . 1605
30.382.1.1 SBI_LIST_HEAD() . . . 1605
30.382.1.2 timer_mtimer_cold_init() . . . 1605
30.382.2 Variable Documentation . . . 1606
30.382.2.1 fdt_timer_mtimer . . . 1606
30.382.2.2 mt_reference . . . 1606
30.382.2.3 sifive_clint_quirks . . . 1606
30.382.2.4 thead_aclint_quirks . . . 1606
30.382.2.5 thead_clint_quirks . . . 1606

30.382.2.6 timer_mtimer_match 1607
30.383 lib/utils/timer/fdt_timer_plmt.c File Reference 1607
30.383.1 Function Documentation 1607
30.383.1.1 fdt_plmt_cold_timer_init() 1607
30.383.2 Variable Documentation 1608
30.383.2.1 fdt_timer_plmt 1608
30.383.2.2 plmt 1608
30.383.2.3 timer_plmt_match 1608
30.384 README.md File Reference 1608

Index **1609**

Chapter 1

RISC-V Open Source Supervisor Binary Interface (OpenSBI)

Copyright and License

The OpenSBI project is:

- Copyright (c) 2019 Western Digital Corporation or its affiliates
- Copyright (c) 2023 RISC-V International

It is distributed under the terms of the BSD 2-clause license ("Simplified BSD License" or "FreeBSD License", SPDX: *BSD-2-Clause*). A copy of this license with OpenSBI copyright can be found in the file `COPYING.BSD`.

All source files in OpenSBI contain the 2-Clause BSD license SPDX short identifier in place of the full license text.
`SPDX-License-Identifier: BSD-2-Clause`

This enables machine processing of license information based on the SPDX License Identifiers that are available on the `SPDX` web site.

OpenSBI source code also contains code reused from other projects as listed below. The original license text of these projects is included in the source files where the reused code is present.

- The libfdt source code is disjunctively dual licensed (GPL-2.0+ OR BSD-2-Clause). Some of this project code is used in OpenSBI under the terms of the BSD 2-Clause license. Any contributions to this code must be made under the terms of both licenses.

See also the third party notices file for more information.

Introduction

The **RISC-V Supervisor Binary Interface (SBI)** is the recommended interface between:

1. A platform-specific firmware running in M-mode and a bootloader, a hypervisor or a general-purpose OS executing in S-mode or HS-mode.
2. A hypervisor running in HS-mode and a bootloader or a general-purpose OS executing in VS-mode.

The *RISC-V SBI specification* is maintained as an independent project by the RISC-V Foundation on `Github`.

The goal of the OpenSBI project is to provide an open-source reference implementation of the RISC-V SBI specifications for platform-specific firmwares executing in M-mode (case 1 mentioned above). An OpenSBI implementation can be easily extended by RISC-V platform and system-on-chip vendors to fit a particular hardware configuration.

The main component of OpenSBI is provided in the form of a platform-independent static library **libsbi.a** implementing the SBI interface. A firmware or bootloader implementation can link against this library to ensure conformance with the SBI interface specifications. *libsbi.a* also defines an interface for integrating with platform-specific operations provided by the platform firmware implementation (e.g. console access functions, inter-processor interrupt control, etc).

To illustrate the use of the *libsbi.a* library, OpenSBI also provides a set of platform-specific support examples. For each example, a platform-specific static library *libplatsbi.a* can be compiled. This library implements SBI call processing by integrating *libsbi.a* with the necessary platform-dependent hardware manipulation functions. For all supported platforms, OpenSBI also provides several runtime firmware examples built using the platform *libplatsbi.a*. These example firmwares can be used to replace the legacy *riscv-pk* bootloader (aka BBL) and enable the use of well-known bootloaders such as `U-Boot`.

Supported SBI version

Currently, OpenSBI fully supports SBI specification *v0.2*. OpenSBI also supports Hart State Management (HSM) SBI extension starting from OpenSBI v0.7. HSM extension allows S-mode software to boot all the harts a defined order rather than legacy method of random booting of harts. As a result, many required features such as CPU hotplug, kexec/kdump can also be supported easily in S-mode. HSM extension in OpenSBI is implemented in a non-backward compatible manner to reduce the maintenance burden and avoid confusion. That's why, any S-mode software using OpenSBI will not be able to boot more than 1 hart if HSM extension is not supported in S-mode.

Linux kernel already supports SBI v0.2 and HSM SBI extension starting from **v5.7-rc1**. If you are using an Linux kernel older than **5.7-rc1** or any other S-mode software without HSM SBI extension, you should stick to OpenSBI v0.6 to boot all the harts. For a UMP systems, it doesn't matter.

N.B. Any S-mode boot loader (i.e. U-Boot) doesn't need to support HSM extension, as it doesn't need to boot all the harts. The operating system should be capable enough to bring up all other non-booting harts using HSM extension.

Required Toolchain and Packages

OpenSBI can be compiled natively or cross-compiled on a x86 host. For cross-compilation, you can build your own toolchain, download a prebuilt one from the `Bootlin toolchain repository` or install a distribution-provided toolchain; if you opt to use LLVM/Clang, most distribution toolchains will support cross-compiling for RISC-V using the same toolchain as your native LLVM/Clang toolchain due to LLVM's ability to support multiple backends in the same binary, so is often an easy way to obtain a working cross-compilation toolchain.

Basically, we prefer toolchains with Position Independent Executable (PIE) support like *riscv64-linux-gnu-gcc*, *riscv64-unknown-freebsd-gcc*, or *Clang/LLVM* as they generate PIE firmware images that can run at arbitrary address with appropriate alignment. If a bare-metal GNU toolchain (e.g. *riscv64-unknown-elf-gcc*) is used, static linked firmware images are generated instead. *Clang/LLVM* can still generate PIE images if a bare-metal triple is used (e.g. *-target riscv64-unknown-elf*).

Please note that only a 64-bit version of the toolchain is available in the Bootlin toolchain repository for now.

In addition to a toolchain, OpenSBI also requires the following packages on the host:

1. device-tree-compiler: The device tree compiler for compiling device tree sources (DTS files).
2. python3: The python 3.0 (or compatible) language support for various scripts.

Building and Installing the OpenSBI Platform-Independent Library

The OpenSBI platform-independent static library *libsbi.a* can be compiled natively or it can be cross-compiled on a host with a different base architecture than RISC-V.

For cross-compiling, the environment variable *CROSS_COMPILE* must be defined to specify the name prefix of the RISC-V compiler toolchain executables, e.g. *riscv64-linux-gnu-* if the gcc executable used is *riscv64-linux-gnu-gcc*.

To build *libsbi.a* simply execute:
```
make
```

All compiled binaries as well as the resulting *libsbi.a* static library file will be placed in the *build/lib* directory. To specify an alternate build root directory path, run:
```
make O=<build_directory>
```

To generate files to be installed for using *libsbi.a* in other projects, run:
```
make install
```

This will create the *install* directory with all necessary include files copied under the *install/include* directory and the library file copied into the *install/lib* directory. To specify an alternate installation root directory path, run:
```
make I=<install_directory> install
```

Building and Installing a Reference Platform Static Library and Firmware

When the *PLATFORM=<platform_subdir>* argument is specified on the make command line, the platform-specific static library *libplatsbi.a* and firmware examples are built for the platform ∗<platform_subdir>∗ present in the directory *platform* in the OpenSBI top directory. For example, to compile the platform library and the firmware examples for the QEMU RISC-V *virt* machine, ∗<platform_subdir>∗ should be *generic*.

To build *libsbi.a*, *libplatsbi.a* and the firmware for one of the supported platforms, run:
```
make PLATFORM=<platform_subdir>
```

An alternate build directory path can also be specified:
```
make PLATFORM=<platform_subdir> O=<build_directory>
```

The platform-specific library *libplatsbi.a* will be generated in the *build/platform/<platform_subdir>/lib* directory. The platform firmware files will be under the *build/platform/<platform_subdir>/firmware* directory. The compiled firmwares will be available in two different formats: an ELF file and an expanded image file.

To install *libsbi.a*, *libplatsbi.a*, and the compiled firmwares, run:
```
make PLATFORM=<platform_subdir> install
```

This will copy the compiled platform-specific libraries and firmware files under the *install/platform/<platform_↩ subdir>/* directory. An alternate install root directory path can be specified as follows:
```
make PLATFORM=<platform_subdir> I=<install_directory> install
```

In addition, platform-specific configuration options can be specified with the top-level make command line. These options, such as *PLATFORM_<xyz>* or *FW_<abc>*, are platform-specific and described in more details in the *docs/platform/<platform_name>.md* files and *docs/firmware/<firmware_name>.md* files.

All OpenSBI platforms support Kconfig style build-time configuration. Users can change the build-time configuration of a platform using a graphical interface as follows:
```
make PLATFORM=<platform_subdir> menuconfig
```

Alternately, an OpenSBI platform can have multiple default configurations and users can select a custom default configuration as follows:
```
make PLATFORM=<platform_subdir> PLATFORM_DEFCONFIG=<platform_custom_defconfig>
```

Building 32-bit / 64-bit OpenSBI Images

By default, building OpenSBI generates 32-bit or 64-bit images based on the supplied RISC-V cross-compile toolchain. For example if *CROSS_COMPILE* is set to *riscv64-linux-gnu-*, 64-bit OpenSBI images will be generated. If building 32-bit OpenSBI images, *CROSS_COMPILE* should be set to a toolchain that is pre-configured to generate 32-bit RISC-V codes, like *riscv32-linux-gnu-*.

However it's possible to explicitly specify the image bits we want to build with a given RISC-V toolchain. This can be done by setting the environment variable *PLATFORM_RISCV_XLEN* to the desired width, for example:
```
export CROSS_COMPILE=riscv64-linux-gnu-
export PLATFORM_RISCV_XLEN=32
```

will generate 32-bit OpenSBI images. And vice vesa.

Building with Clang/LLVM

OpenSBI can also be built with Clang/LLVM. To build with just Clang but keep the default binutils (which will still use the *CROSS_COMPILE* prefix if defined), override the *CC* make variable with:

```
make CC=clang
```

To build with a full LLVM-based toolchain, not just Clang, enable the *LLVM* option with:

```
make LLVM=1
```

When using Clang, *CROSS_COMPILE* often does not need to be defined unless using GNU binutils with prefixed binary names. *PLATFORM_RISCV_XLEN* will be used to infer a default triple to pass to Clang, so if *PLATFORM↩_RISCV_XLEN* itself defaults to an undesired value then prefer setting that rather than the full triple via *CROSS↩_COMPILE*. If *CROSS_COMPILE* is nonetheless defined, rather than being used as a prefix for the executable name, it will instead be passed via the `--target` option with the trailing `-` removed, so must be a valid triple.

These can also be mixed; for example using a GCC cross-compiler but LLVM binutils would be:

```
make CC=riscv64-linux-gnu-gcc LLVM=1
```

These variables must be passed for all the make invocations described in this document.

NOTE: Using Clang with a `riscv*-linux-gnu` GNU binutils linker has been seen to produce broken binaries with missing relocations; it is therefore currently recommended that this combination be avoided.

Building with timestamp and compiler info

When doing development, we may want to know the build time and compiler info for debug purpose. OpenSBI can also be built with timestamp and compiler info. To build with those info and print it out at boot time, we can just simply add `BUILD_INFO=y`, like:

```
make BUILD_INFO=y
```

But if you have used `BUILD_INFO=y`, and want to switch back to `BUILD_INFO=n`, you must do

```
make clean
```

before the next build.

NOTE: Using `BUILD_INFO=y` without specifying SOURCE_DATE_EPOCH will violate `reproducible builds`. This definition is ONLY for development and debug purpose, and should NOT be used in a product which follows "reproducible builds".

Building with optimization off for debugging

When debugging OpenSBI, we may want to turn off the compiler optimization and make debugging produce the expected results for a better debugging experience. To build with optimization off we can just simply add `DEBUG=1`, like:

```
make DEBUG=1
```

This definition is ONLY for development and debug purpose, and should NOT be used in a product build.

Contributing to OpenSBI

The OpenSBI project encourages and welcomes contributions. Contributions should follow the rules described in the OpenSBI Contribution Guideline document. In particular, all patches sent should contain a Signed-off-by tag.

The Contributors List document provides a list of individuals and organizations actively contributing to the OpenSBI project.

Documentation

Detailed documentation of various aspects of OpenSBI can be found under the *docs* directory. The documentation covers the following topics.

- [Contribution Guideline]: Guideline for contributing code to OpenSBI project
- [Library Usage]: API documentation of OpenSBI static library *libsbi.a*
- [Platform Requirements]: Requirements for using OpenSBI on a platform
- [Platform Support Guide]: Guideline for implementing support for new platforms
- [Platform Documentation]: Documentation of the platforms currently supported.
- [Firmware Documentation]: Documentation for the different types of firmware examples build supported by OpenSBI.
- [Domain Support]: Documentation for the OpenSBI domain support which helps users achieve system-level partitioning using OpenSBI.

OpenSBI source code is also well documented. For source level documentation, doxygen style is used. Please refer to the `Doxygen manual` for details on this format.

Doxygen can be installed on Linux distributions using ∗.deb∗ packages using the following command.
```
sudo apt-get install doxygen doxygen-latex doxygen-doc doxygen-gui graphviz
```

For ∗.rpm∗ based Linux distributions, the following commands can be used.
```
sudo yum install doxygen doxygen-latex doxywizard graphviz
```

or
```
sudo yum install doxygen doxygen-latex doxywizard graphviz
```

To build a consolidated *refman.pdf* of all documentation, run:
```
make docs
```

or
```
make O=<build_directory> docs
```

the resulting *refman.pdf* will be available under the directory ∗<build_directory>/docs/latex∗. To install this file, run:
```
make install_docs
```

or
```
make I=<install_directory> install_docs
```

refman.pdf will be installed under ∗<install_directory>/docs∗.

Chapter 2

OpenSBI Contribution Guideline

All contributions to OpenSBI must be sent via email patches to the OpenSBI mailing list at `opensbi@lists.infradead.org`

To join the OpenSBI mailing list, please visit the `OpenSBI infradead page`.

The mailing list based patch approach is preferred over github PRs so that they are visible to a wider audience. All accepted patches on the OpenSBI mailing list will be taken by one of the OpenSBI maintainers and merged into the `OpenSBI main repository`.

All contributed work must follow the following rules:

1. OpenSBI code should be written in accordance to the `Linux coding style`.
2. This project embraces the `Developer Certificate of Origin (DCO)` for contributions. This means that you must agree to the following prior to submitting patches: if you agree with this developer certificate you acknowledge this by adding a Signed-off-by tag to your patch commit log. Every submitted patch must have this tag.
3. A commit message must have a subject line, followed by a blank line, followed by a description of the patch content. A blank line and the author Signed-off-by tag must follow this description.
4. A commit subject line must start with a prefix followed by a ":". Common prefixes are for example "lib:", "platform:", "firmware:", "docs:", "utils:" and "top:".
5. Maintainers should use "Rebase and Merge" when using GitHub to merge pull requests to avoid creating unnecessary merge commits.
6. Maintainers should avoid creating branches directly in the main riscv/opensbi repository. Instead, prefer using a fork of the riscv/opensbi main repository and branches within that fork to create pull requests.
7. A maintainer cannot merge his own pull requests in the riscv/opensbi main repository.

Developer Certificate of Origin Version 1.1
Copyright (C) 2004, 2006 The Linux Foundation and its contributors. 660 York Street, Suite 102, San Francisco, CA 94110 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
Developer's Certificate of Origin 1.1
By making a contribution to this project, I certify that:
(a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or
(b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created

in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or
(c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.
(d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

Chapter 3

OpenSBI Platform Support Guideline

The OpenSBI platform support allows an implementation to define a set of platform-specific hooks (hardware manipulation functions) in the form of a *struct sbi_platform* data structure instance. This instance is required by the platform-independent *libsbi.a* to execute platform-specific operations.
Each of the reference platform supports provided by OpenSBI defines an instance of the *struct sbi_platform* data structure. For each supported platform, *libplatsbi.a* integrates this instance with *libsbi.a* to create a platform-specific OpenSBI static library. This library is installed in ∗<install_directory>/platform/<platform_subdir>/lib/libplatsbi.a∗
OpenSBI also provides implementation examples of bootable runtime firmwares for the supported platforms. These firmwares are linked against *libplatsbi.a*. Firmware binaries are installed in ∗<install_directory>/platform/<platform_subdir>/bin∗. These firmwares can be used as executable runtime firmwares on the supported platforms as a replacement for the legacy *riscv-pk* boot loader (BBL).
A complete doxygen-style documentation of *struct sbi_platform* and related APIs is available in the file *include/sbi/sbi_platform.h*.

Adding support for a new platform

Support for a new platform named <*xyz*> can be added as follows:

1. Create a directory named <*xyz*> under the *platform/* directory.
2. Create platform configuration files named *Kconfig* and *configs/defconfig* under the *platform/*<*xyz*>*/* directory. These configuration files will provide the build time configuration for the sources to be compiled.
3. Create a *platform/*<*xyz*>*/objects.mk* file for listing the platform object files to be compiled. This file also provides platform-specific compiler flags, and select firmware options.
4. Create a *platform/*<*xyz*>*/platform.c* file providing a *struct sbi_platform* instance.

A platform support code template is available under the *platform/template* directory. Copying this directory and its content as a new directory named <*xyz*> under the *platform/* directory will create all the files mentioned above.

Chapter 4

OpenSBI Platform Requirements

The RISC-V platform requirements for OpenSBI can change over time with advances in RISC-V specifications and ecosystem.
To handle this, we have two types of RISC-V platform requirements:

1. **Base platform requirements** which apply to all OpenSBI releases
2. **Release specific platform requirements** which apply to a OpenSBI release and later releases

Currently, we don't have any **Release specific platform requirements**, but such platform requirements will be added in future.

Base Platform Requirements

The base RISC-V platform requirements for OpenSBI are as follows:

1. At least rv32ima_zicsr or rv64ima_zicsr required on all HARTs
2. At least one HART should have S-mode support because:
   - SBI calls are meant for RISC-V S-mode (Supervisor mode)
   - OpenSBI implements SBI calls for S-mode software
3. The MTVEC CSR on all HARTs must support direct mode
4. The PMP CSRs are optional. If PMP CSRs are not implemented then OpenSBI cannot protect M-mode firmware and secured memory regions
5. The TIME CSR is optional. If TIME CSR is not implemented in hardware then a 64-bit MMIO counter is required to track time and emulate TIME CSR
6. Hardware support for injecting M-mode software interrupts on a multi-HART platform

The RISC-V extensions not covered by rv32ima_zicsr or rv64ima_zicsr are optional for OpenSBI. Although, Open↩ SBI will detect and handle some of these optional RISC-V extensions at runtime.
The optional RISC-V extensions handled by OpenSBI at runtime are:

- D-extension: Double precision floating point
- F-extension: Single precision floating point
- H-extension: Hypervisor

Chapter 5

OpenSBI Library Usage

OpenSBI provides two types of static libraries:

1. *libsbi.a* - A platform-independent generic static library implementing the interface defined by the SBI specifications. Platform-specific processing hooks for the execution of this interface must be provided by the firmware or bootloader linking with this library. This library is installed as ∗<install_directory>/lib/libsbi.a∗

2. *libplatsbi.a* - An example platform-specific static library integrating *libsbi.a* with platform-specific hooks. This library is available only for the platforms supported by OpenSBI. This library is installed as ∗<install_↩directory>/platform/<platform_subdir>/lib/libplatsbi.a∗

Implementations may choose either *libsbi.a* or *libplatsbi.a* to link with their firmware or bootloader. In the case of *libsbi.a*, platform-specific hooks in the form of a *struct sbi_platform* instance need to be provided.
The platform-specific example firmwares provided by OpenSBI are not mandatory. An implementation may choose to link the OpenSBI generic static library together with an M-mode firmware or bootloader providing the hardware-specific hooks. Since OpenSBI is a statically linked library, users must ensure that the license of these external components is compatible with the OpenSBI license.

Constraints on OpenSBI usage from external firmware

Users have to ensure that an external firmware or bootloader linking against OpenSBI static libraries (*libsbi.a* or *libplatsbi.a*) is compiled with the same GCC target options *-mabi*, *-march*, and *-mcmodel*.
There are only two constraints on calling any OpenSBI library function from an external M-mode firmware or bootloader:

1. The RISC-V *MSCRATCH* CSR must point to a valid OpenSBI scratch space (i.e. a *struct sbi_scratch* instance).

2. The RISC-V *SP* register (i.e. the stack pointer) must be set per-HART pointing to distinct non-overlapping stacks.

The most important functions from an external firmware or bootloader perspective are *sbi_init()* and *sbi_trap_handler()*.
In addition to the above constraints, the external firmware or bootloader must ensure that interrupts are disabled in the *MSTATUS* and *MIE* CSRs when calling the functions *sbi_init()* and *sbi_trap_handler()*.
The *sbi_init()* function should be called by the external firmware or bootloader for each HART that is powered-up at boot-time or in response to a CPU hotplug event.
The *sbi_trap_handler()* function should be called by the external firmware or bootloader to service the following interrupts and traps:

1. M-mode timer interrupt

2. M-mode software interrupt

3. Illegal instruction trap

4. Misaligned load trap

5. Misaligned store trap
6. Supervisor ecall trap
7. Hypervisor ecall trap

Note: external firmwares or bootloaders can be more conservative by forwarding all traps and interrupts to *sbi_trap_handler()*.

Definitions of OpenSBI Data Types for the External Firmware

OpenSBI can be built as library using external firmware build system such as EDK2 code base (The open source of UEFI firmware implementation) and linked with external firmware drivers based on the external firmware architecture.
OPENSBI_EXTERNAL_SBI_TYPES identifier is introduced to *sbi_types.h* for selecting external header file during the build preprocess in order to define OpensSBI data types based on external firmware data type binding. For example, *bool* is declared as *int* in sbi_types.h. However, in EDK2 build system, *bool* is declared as *BOOLEAN* which is defined as *unsigned char* data type.
External firmware can define **OPENSBI_EXTERNAL_SBI_TYPES** in CFLAGS and specify it to the header file maintained in its code tree. However, the external build system has to address the additional include directory for the external header file based on its own build system. For example, *-D∗∗∗OPENSBI_EXTERNAL_SBI_↩TYPES∗∗∗=OpensbiTypes.h* Above tells *sbi_types.h* to refer to *OpensbiTypes.h* instead of using original definitions of data types.

Chapter 6

OpenSBI Domain Support

An OpenSBI domain is a system-level partition (subset) of underlying hardware having its own memory regions (RAM and MMIO devices) and HARTs. The OpenSBI will try to achieve secure isolation between domains using RISC-V platform features such as PMP, ePMP, IOPMP, SiFive Shield, etc.
Important entities which help implement OpenSBI domain support are:

- **struct sbi_domain_memregion** - Representation of a domain memory region
- **struct sbi_hartmask** - Representation of domain HART set
- **struct sbi_domain** - Representation of a domain instance

Each HART of a RISC-V platform must have an OpenSBI domain assigned to it. The OpenSBI platform support is responsible for populating domains and providing HART id to domain mapping. The OpenSBI domain support will by default assign **the ROOT domain** to all HARTs of a RISC-V platform, so it is not mandatory for the OpenSBI platform support to populate domains.

Domain Memory Region

A domain memory region is represented by **struct sbi_domain_memregion** in OpenSBI and has following details:

- **order** - The size of a memory region is **2 ^ order** where **order** must be **3 <= order <= __riscv_xlen**
- **base** - The base address of a memory region is **2 ^ order** aligned start address
- **flags** - The flags of a memory region represent memory type (i.e. RAM or MMIO) and allowed accesses (i.e. READ, WRITE, EXECUTE, etc.)

Domain Instance

A domain instance is represented by **struct sbi_domain** in OpenSBI and has following details:

- **index** - Logical index of this domain
- **name** - Name of this domain
- **assigned_harts** - HARTs assigned to this domain
- **possible_harts** - HARTs possible in this domain
- **regions** - Array of memory regions terminated by a memory region with order zero
- **boot_hartid** - HART id of the HART booting this domain. The domain boot HART will be started at boot-time if boot HART is possible and assigned for this domain.
- **next_addr** - Address of the next booting stage for this domain
- **next_arg1** - Arg1 (or 'a1' register) of the next booting stage for this domain

- **next_mode** - Privilege mode of the next booting stage for this domain. This can be either S-mode or U-mode.
- **system_reset_allowed** - Is domain allowed to reset the system?
- **system_suspend_allowed** - Is domain allowed to suspend the system?

The memory regions represented by **regions** in **struct sbi_domain** have following additional constraints to align with RISC-V PMP requirements:

- A memory region to protect OpenSBI firmware from S-mode and U-mode should always be present
- For two overlapping memory regions, one should be sub-region of another
- Two overlapping memory regions should not be of same size
- Two overlapping memory regions cannot have same flags
- Memory access checks on overlapping address should prefer smallest overlapping memory region flags.

ROOT Domain

The ROOT domain is the default OpenSBI domain which is assigned by default to all HARTs of a RISC-V platform. The OpenSBI domain support will hand-craft **the ROOT domain** very early at boot-time in the following manner:

- **index** - Logical index of the ROOT domain is always zero
- **name** - Name of the ROOT domain is "root"
- **assigned_harts** - At boot-time all valid HARTs of a RISC-V platform are assigned the ROOT domain which changes later based on OpenSBI platform support
- **possible_harts** - All valid HARTs of a RISC-V platform are possible HARTs of the ROOT domain
- **regions** - Two memory regions available to the ROOT domain: **A)** A memory region to protect OpenSBI firmware from S-mode and U-mode **B)** A memory region of **order=__riscv_xlen** allowing S-mode and U-mode access to full memory address space
- **boot_hartid** - Coldboot HART is the HART booting the ROOT domain
- **next_addr** - Next booting stage address in coldboot HART scratch space is the next address for the ROOT domain
- **next_arg1** - Next booting stage arg1 in coldboot HART scratch space is the next arg1 for the ROOT domain
- **next_mode** - Next booting stage mode in coldboot HART scratch space is the next mode for the ROOT domain
- **system_reset_allowed** - The ROOT domain is allowed to reset the system
- **system_suspend_allowed** - The ROOT domain is allowed to suspend the system

Domain Effects

Few noteworthy effects of a system partitioned into domains are as follows:

- At any point in time, a HART is running in exactly one OpenSBI domain context
- The SBI IPI and RFENCE calls from HART A are restricted to the HARTs in domain assigned to HART A
- The SBI HSM calls which try to change/read state of HART B from HART A will only work if both HART A and HART B are assigned same domain
- A HART running in S-mode or U-mode can only access memory based on the memory regions of the domain assigned to the HART

Domain Device Tree Bindings

The OpenSBI domains can be described in the **device tree (DT) blob** (or flattened device tree) passed to the OpenSBI firmwares by the previous booting stage. This allows OpenSBI platform support to parse and populate OpenSBI domains from the device tree blob (or flattened device tree).

Domain Configuration Node

All OpenSBI domain description related DT nodes should be under the domain configuration DT node. The ∗∗/chosen∗∗ DT node is the preferred parent of the domain configuration DT node.
The DT properties of a domain configuration DT node are as follows:

- **compatible** (Mandatory) - The compatible string of the domain configuration. This DT property should have value ∗"opensbi,domain,config"∗

Domain Memory Region Node

The domain memory region DT node describes details of a memory region and can be pointed by multiple domain instance DT nodes. The access permissions of the memory region are specified separately in domain instance node.
The DT properties of a domain memory region DT node are as follows:

- **compatible** (Mandatory) - The compatible string of the domain memory region. This DT property should have value ∗"opensbi,domain,memregion"∗
- **base** (Mandatory) - The base address of the domain memory region. This DT property should have a **2** ^ **order** aligned 64 bit address (i.e. two DT cells).
- **order** (Mandatory) - The order of the domain memory region. This DT property should have a 32 bit value (i.e. one DT cell) in the range **3** <= **order** <= **__riscv_xlen**.
- **mmio** (Optional) - A boolean flag representing whether the domain memory region is a memory-mapped I/O (MMIO) region.
- **devices** (Optional) - The list of device DT node phandles for devices which fall under this domain memory region.

Domain Instance Node

The domain instance DT node describes set of possible HARTs, set of memory regions, and other details of a domain instance.
The DT properties of a domain instance DT node are as follows:

- **compatible** (Mandatory) - The compatible string of the domain instance. This DT property should have value ∗"opensbi,domain,instance"∗
- **possible-harts** (Optional) - The list of CPU DT node phandles for the the domain instance. This list represents the possible HARTs of the domain instance.
- **regions** (Optional) - The list of domain memory region DT node phandle and access permissions for the domain instance. Each list entry is a pair of DT node phandle and access permissions. The access permissions are represented as a 32bit bitmask having bits: **M readable** (BIT[0]), **M writeable** (BIT[1]), **M executable** (BIT[2]), **SU readable** (BIT[3]), **SU writable** (BIT[4]), and **SU executable** (BIT[5]). The enforce permission bit (BIT[6]), if set, will lock the permissions in the PMP. This will enforce the permissions on M-mode as well which otherwise will have unrestricted access. This bit must be used with caution because no changes can be made to a PMP entry once its locked until the hart is reset. Any region of a domain defined in DT node cannot have only M-bits set in access permissions i.e. it cannot be an m-mode only accessible region.
- **boot-hart** (Optional) - The DT node phandle of the HART booting the domain instance. If coldboot HART is assigned to the domain instance then this DT property is ignored and the coldboot HART is assumed to be the boot HART of the domain instance.

- **next-arg1** (Optional) - The 64 bit next booting stage arg1 for the domain instance. If this DT property is not available and coldboot HART is not assigned to the domain instance then **next booting stage arg1 of coldboot HART** is used as default value.
- **next-addr** (Optional) - The 64 bit next booting stage address for the domain instance. If this DT property is not available and coldboot HART is not assigned to the domain instance then **0x0** is used as default value. If this DT property is not available and coldboot HART is assigned to the domain instance then **next booting stage address of coldboot HART** is used as default value.
- **next-mode** (Optional) - The 32 bit next booting stage mode for the domain instance. The possible values of this DT property are: **0x1** (S-mode), and **0x0** (U-mode). If this DT property is not available and coldboot HART is not assigned to the domain instance then **0x1** is used as default value. If this DT property is not available and coldboot HART is assigned to the domain instance then **next booting stage mode of coldboot HART** is used as default value.
- **system-reset-allowed** (Optional) - A boolean flag representing whether the domain instance is allowed to do system reset.
- **system-suspend-allowed** (Optional) - A boolean flag representing whether the domain instance is allowed to do system suspend.

Assigning HART To Domain Instance

By default, all HARTs are assigned to **the ROOT domain**. The OpenSBI platform support can provide the HART to domain instance assignment using platform specific callback.
The HART to domain instance assignment can be parsed from the device tree using optional DT property **opensbi-domain** in each CPU DT node. The value of DT property **opensbi-domain** is the DT phandle of the domain instance DT node. If **opensbi-domain** DT property is not specified then corresponding HART is assigned to **the ROOT domain**.

Domain Configuration Only Accessible to OpenSBI

The software running inside a domain instance should only be aware of devices and hardware resources accessible to itself.
To hide domain configuration from domain instances, the following should be done:

- The previous booting stage should preferably provide a separate device tree for each domain instance and mention location of device tree in respective domain instance DT nodes using **next-arg1** DT property.
- If domain assigned to a HART does not have separate device tree then OpenSBI platform support should remove all domain configuration details from the device tree passed by previous booting stage before passing it to the next booting stage.

Example

```
chosen {
    opensbi-domains {
        compatible = "opensbi,domain,config";

        tmem: tmem {
            compatible = "opensbi,domain,memregion";
            base = <0x0 0x80100000>;
            order = <20>;
        };

        tuart: tuart {
            compatible = "opensbi,domain,memregion";
            base = <0x0 0x10011000>;
            order = <12>;
            mmio;
            devices = <&uart1>;
        };

        allmem: allmem {
            compatible = "opensbi,domain,memregion";
            base = <0x0 0x0>;
            order = <64>;
```

```
        };

        tdomain: trusted-domain {
            compatible = "opensbi,domain,instance";
            possible-harts = <&cpu0>;
            regions = <&tmem 0x3f>, <&tuart 0x3f>;
            boot-hart = <&cpu0>;
            next-arg1 = <0x0 0x0>;
            next-addr = <0x0 0x80100000>;
            next-mode = <0x0>;
            system-reset-allowed;
            system-suspend-allowed;
        };

        udomain: untrusted-domain {
            compatible = "opensbi,domain,instance";
            possible-harts = <&cpu1 &cpu2 &cpu3 &cpu4>;
            regions = <&tmem 0x0>, <&tuart 0x0>, <&allmem 0x3f>;
        };
    };
};

cpus {
    #address-cells = <1>;
    #size-cells = <0>;
    timebase-frequency = <10000000>;

    cpu0: cpu@0 {
        device_type = "cpu";
        reg = <0x00>;
        compatible = "riscv";
        opensbi-domain = <&tdomain>;
        ...
    };

    cpu1: cpu@1 {
        device_type = "cpu";
        reg = <0x01>;
        compatible = "riscv";
        opensbi-domain = <&udomain>;
        ...
    };

    cpu2: cpu@2 {
        device_type = "cpu";
        reg = <0x02>;
        compatible = "riscv";
        opensbi-domain = <&udomain>;
        ...
    };

    cpu3: cpu@3 {
        device_type = "cpu";
        reg = <0x03>;
        compatible = "riscv";
        opensbi-domain = <&udomain>;
        ...
    };

    cpu4: cpu@4 {
        device_type = "cpu";
        reg = <0x04>;
        compatible = "riscv";
        opensbi-domain = <&udomain>;
        ...
    };
};

uart1: serial@10011000 {
    ...
};
```

Chapter 7

OpenSBI Device Tree Configuration Guideline

Some configurations of OpenSBI's Generic Platform can be described in the **device tree (DT) blob** (or flattened device tree) passed to the OpenSBI firmwares by the previous booting stage. OpenSBI will parse and use these configurations during the boot phase, but delete them from the device tree at the end of cold boot.

OpenSBI Configuration Node

All nodes related to OpenSBI configuration should be under the OpenSBI configuration DT node. The ∗∗/chosen∗∗ DT node is the preferred parent of the OpenSBI configuration DT node.
The DT properties of a domain configuration DT node are as follows:

- **compatible** (Mandatory) - The compatible string of the OpenSBI configuration. This DT property should have value ∗"opensbi,config"∗
- **cold-boot-harts** (Optional) - If a platform lacks an override cold_boot_allowed() mechanism, this DT property specifies that a set of harts is permitted to perform a cold boot. Otherwise, all harts are allowed to cold boot.
- **heap-size** (Optional) - When present, the specified value is used as the size of the heap in bytes.
- **system-suspend-test** (Optional) - When present, enable a system suspend test implementation which simply waits five seconds and issues a WFI.

The OpenSBI Configuration Node will be deleted at the end of cold boot (replace the node (subtree) with nop tags).

Example

```
chosen {
    opensbi-config {
        compatible = "opensbi,config";
        cold-boot-harts = <&cpu1 &cpu2 &cpu3 &cpu4>;
        heap-size = <0x400000>;
        system-suspend-test;
    };
};

cpus {
    #address-cells = <1>;
    #size-cells = <0>;
    timebase-frequency = <10000000>;

    cpu0: cpu@0 {
        device_type = "cpu";
        reg = <0x00>;
        compatible = "riscv";
        ...
    };

    cpu1: cpu@1 {
        device_type = "cpu";
        reg = <0x01>;
        compatible = "riscv";
        ...
```

```
    };

    cpu2: cpu@2 {
        device_type = "cpu";
        reg = <0x02>;
        compatible = "riscv";
        ...
    };

    cpu3: cpu@3 {
        device_type = "cpu";
        reg = <0x03>;
        compatible = "riscv";
        ...
    };

    cpu4: cpu@4 {
        device_type = "cpu";
        reg = <0x04>;
        compatible = "riscv";
        ...
    };
};

uart1: serial@10011000 {
    ...
};
```

Chapter 8

Writing tests for OpenSBI

SBIUnit

SBIUnit is a set of macros and functions which simplify the test development and automate the test execution and evaluation. All of the SBIUnit definitions are in the `include/sbi/sbi_unit_test.h` header file, and implementations are available in `lib/sbi/tests/sbi_unit_test.c`.

Simple SBIUnit test

For instance, we would like to test the following function from `lib/sbi/sbi_string.c`:

```
size_t sbi_strlen(const char *str)
{
    unsigned long ret = 0;

    while (*str != '\0') {
        ret++;
        str++;
    }

    return ret;
}
```

which calculates the string length.

Create the file `lib/sbi/tests/sbi_string_test.c` with the following content:

```
#include <sbi/sbi_unit_test.h>
#include <sbi/sbi_string.h>

static void strlen_test(struct sbiunit_test_case *test)
{
    SBIUNIT_EXPECT_EQ(test, sbi_strlen("Hello"), 5);
    SBIUNIT_EXPECT_EQ(test, sbi_strlen("Hell\0o"), 4);
}

static struct sbiunit_test_case string_test_cases[] = {
    SBIUNIT_TEST_CASE(strlen_test),
    SBIUNIT_END_CASE,
};

SBIUNIT_TEST_SUITE(string_test_suite, string_test_cases);
```

Then, add the corresponding Makefile entries to `lib/sbi/tests/objects.mk`:

```
 -makefile
...
carray-sbi_unit_tests-$(CONFIG_SBIUNIT) += string_test_suite
libsbi-objs-$(CONFIG_SBIUNIT) += tests/sbi_string_test.o
```

Now, run `make clean` in order to regenerate the carray-related files.

Recompile OpenSBI with the CONFIG_SBIUNIT option enabled and run it in QEMU. You will see something like this:

```
# make PLATFORM=generic run
...
# Running SBIUNIT tests #
...
## Running test suite: string_test_suite
[PASSED] strlen_test
1 PASSED / 0 FAILED / 1 TOTAL
```

Now let's try to change this test in the way that it will fail:

```
- SBIUNIT_EXPECT_EQ(test, sbi_strlen("Hello"), 5);
+ SBIUNIT_EXPECT_EQ(test, sbi_strlen("Hello"), 100);
```

`make all` and `make run` it again:

```
...
# Running SBIUNIT tests #
...
## Running test suite: string_test_suite
[SBIUnit] [.../opensbi/lib/sbi/tests/sbi_string_test.c:6]: strlen_test: Condition "(sbi_strlen("Hello")) ==
      (100)" expected to be true!
[FAILED] strlen_test
0 PASSED / 1 FAILED / 1 TOTAL
```

Covering the static functions / using the static definitions

SBIUnit also allows you to test static functions. In order to do so, simply include your test source in the file you would like to test. Complementing the example above, just add this to the `lib/sbi/sbi_string.c` file:

```
#ifdef CONFIG_SBIUNIT
#include "tests/sbi_string_test.c"
#endif
```

In this case you should only add a new carray entry pointing to the test suite to `lib/sbi/tests/objects.mk`:

```
 -makefile
...
carray-sbi_unit_tests-$(CONFIG_SBIUNIT) += string_test_suite
```

You don't have to compile the `sbi_string_test.o` separately, because the test code will be included into the `sbi_string` object file.

"Mocking" the structures

See the example of structure "mocking" in `lib/sbi/tests/sbi_console_test.c`, where the sbi_console_device structure was mocked to be used in various console-related functions in order to test them.

API Reference

All of the `SBIUNIT_EXPECT_*` macros will cause a test case to fail if the corresponding conditions are not met, however, the execution of a particular test case will not be stopped.
All of the `SBIUNIT_ASSERT_*` macros will cause a test case to fail and stop immediately, triggering a panic.

Chapter 9

OpenSBI Platform Firmwares

OpenSBI provides firmware builds for specific platforms. Different types of firmwares are supported to deal with the differences between different platforms early boot stage. All firmwares will execute the same initialization procedure of the platform hardware according to the platform specific code as well as OpenSBI generic library code. The supported firmwares type will differ in how the arguments passed by the platform early boot stage are handled, as well as how the boot stage following the firmware will be handled and executed.
The previous booting stage will pass information via the following registers of RISC-V CPU:

- hartid via *a0* register
- device tree blob address in memory via *a1* register. The address must be aligned to 8 bytes.

OpenSBI currently supports three different types of firmwares.

Firmware with Dynamic Information (*FW_DYNAMIC*)

The *FW_DYNAMIC* firmware gets information about the next booting stage entry, e.g. a bootloader or an OS kernel, from previous booting stage at runtime.
A *FW_DYNAMIC* firmware is particularly useful when the booting stage executed prior to OpenSBI firmware is capable of loading both the OpenSBI firmware and the booting stage binary to follow OpenSBI firmware.

Firmware with Jump Address (*FW_JUMP*)

The *FW_JUMP* firmware assumes a fixed address of the next booting stage entry, e.g. a bootloader or an OS kernel, without directly including the binary code for this next stage.
A *FW_JUMP* firmware is particularly useful when the booting stage executed prior to OpenSBI firmware is capable of loading both the OpenSBI firmware and the booting stage binary to follow OpenSBI firmware.

Firmware with Payload (*FW_PAYLOAD*)

The *FW_PAYLOAD* firmware directly includes the binary code for the booting stage to follow OpenSBI firmware execution. Typically, this payload will be a bootloader or an OS kernel.
A *FW_PAYLOAD* firmware is particularly useful when the booting stage executed prior to OpenSBI firmware is not capable of loading both OpenSBI firmware and the booting stage to follow OpenSBI firmware.
A *FW_PAYLOAD* firmware is also useful for cases where the booting stage prior to OpenSBI firmware does not pass a *flattened device tree (FDT file)*. In such case, a *FW_PAYLOAD* firmware allows embedding a flattened device tree in the .rodata section of the final firmware.

Firmware Configuration and Compilation

All firmware types support the following common compile time configuration parameters:

- **FW_TEXT_START** - Defines the compile time address of the OpenSBI firmware. This configuration parameter is optional and assumed to be 0 if not specified.
- **FW_FDT_PATH** - Path to an external flattened device tree binary file to be embedded in the ∗.rodata∗ section of the final firmware. If this option is not provided then the firmware will expect the FDT to be passed as an argument by the prior booting stage.
- **FW_FDT_PADDING** - Optional zero bytes padding to the embedded flattened device tree binary file specified by **FW_FDT_PATH** option.

Additionally, each firmware type as a set of type specific configuration parameters. Detailed information for each firmware type can be found in the following documents.

- *[FW_DYNAMIC]*: The *Firmware with Dynamic Information (FW_DYNAMIC)* is described in more details in the file *fw_dynamic.md*.
- *[FW_JUMP]*: The *Firmware with Jump Address (FW_JUMP)* is described in more details in the file *fw_jump.md*.
- *[FW_PAYLOAD]*: The *Firmware with Payload (FW_PAYLOAD)* is described in more details in the file *fw_payload.md*.

Providing different payloads to OpenSBI Firmware

OpenSBI firmware can accept various payloads using a compile time option. Typically, these payloads refer to the next stage boot loader (e.g. U-Boot) or operating system kernel images (e.g. Linux). By default, OpenSBI automatically provides a test payload if no specific payload is specified at compile time.
To specify a payload at compile time, the make variable *FW_PAYLOAD_PATH* is used.

```
make PLATFORM=<platform_subdir> FW_PAYLOAD_PATH=<payload path>
```

The instructions to build each payload is different and the details can be found in the *docs/firmware/payload_↩ <payload_name>.md* files.

Options for OpenSBI Firmware behaviors

An optional compile time flag FW_OPTIONS can be used to control the OpenSBI firmware run-time behaviors.

```
make PLATFORM=<platform_subdir> FW_OPTIONS=<options>
```

FW_OPTIONS is a bitwise or'ed value of various options, eg: *FW_OPTIONS=0x1* stands for disabling boot prints from the OpenSBI library.
For all supported options, please check "enum sbi_scratch_options" in the *include/sbi/sbi_scratch.h* header file.

Chapter 10

OpenSBI Firmware with Dynamic Information (FW_DYNAMIC)

OpenSBI **firmware with dynamic info (FW_DYNAMIC)** is a firmware which gets information about next booting stage (e.g. a bootloader or an OS) and runtime OpenSBI library options from previous booting stage.
The previous booting stage will pass information to *FW_DYNAMIC* by creating *struct fw_dynamic_info* in memory and passing its address to *FW_DYNAMIC* via *a2* register of RISC-V CPU. The address must be aligned to 8 bytes on RV64 and 4 bytes on RV32.
A *FW_DYNAMIC* firmware is particularly useful when the booting stage executed prior to OpenSBI firmware is capable of loading both the OpenSBI firmware and the booting stage binary to follow OpenSBI firmware.

FW_DYNAMIC Compilation

A platform can enable *FW_DYNAMIC* firmware using any of the following methods.

1. Specifying `FW_DYNAMIC=y` on the top level `make` command line.
2. Specifying `FW_DYNAMIC=y` in the target platform *objects.mk* configuration file.

The compiled *FW_DYNAMIC* firmware ELF file is named *fw_dynamic.elf*. It's expanded image file is *fw_dynamic.bin*. Both files are created in the platform specific build directory under the *build/platform/<platform_subdir>/firmware* directory.

FW_DYNAMIC Firmware Configuration Options

The *FW_DYNAMIC* firmware does not require any platform specific configuration parameters because all required information is passed by previous booting stage at runtime via *struct fw_dynamic_info*.

Chapter 11

OpenSBI Firmware with Jump Address (FW_JUMP)

OpenSBI **firmware with Jump Address (FW_JUMP)** is a firmware which only handles the address of the next booting stage entry, e.g. a bootloader or an OS kernel, without directly including the binary code for this next stage. A *FW_JUMP* firmware is particularly useful when the booting stage executed prior to the OpenSBI firmware is capable of loading both the OpenSBI firmware and the booting stage binary to follow the OpenSBI firmware.

FW_JUMP Compilation

A platform *FW_JUMP* firmware can be enabled by any of the following methods:

1. Specifying `FW_JUMP=y` on the top level `make` command line.
2. Specifying `FW_JUMP=y` in the target platform *objects.mk* configuration file.

The compiled *FW_JUMP* firmware ELF file is named *fw_jump.elf*. Its expanded image file is *fw_jump.bin*. Both files are created in the platform-specific build directory under the *build/platform/<platform_subdir>/firmware* directory.

FW_JUMP Firmware Configuration Options

To operate correctly, a *FW_JUMP* firmware requires some configuration parameters to be defined using either the top level `make` command line or the target platform *objects.mk* configuration file. The possible parameters are as follows:

- **FW_JUMP_ADDR** - Address of the entry point of the booting stage to be executed following OpenSBI firmware. This address generally corresponds exactly to the address where this next booting stage was loaded. At least one of *FW_JUMP_ADDR* and *FW_JUMP_OFFSET* (see below) should be defined. Compilation errors will result from not defining one of them.
- **FW_JUMP_OFFSET** - Address offset from the opensbi load address where the entry point of the next booting stage is located. This offset is used as relocatable address of the next booting stage entry point. If *FW_JUMP_ADDR* is also defined, the firmware will prefer *FW_JUMP_ADDR*.
- **FW_JUMP_FDT_ADDR** - Address where the *flattened device tree (FDT file)* passed by the prior booting stage will be placed in memory before executing the booting stage following the OpenSBI firmware. If this option is not provided, then the OpenSBI firmware will pass the FDT address passed by the previous booting stage to the next booting stage.

  When using the default *FW_JUMP_FDT_ADDR* with *PLATFORM=generic*, you must ensure *FW_JUMP_FDT_ADDR* is set high enough to avoid overwriting the kernel. You can use the following method (e.g., using bash or zsh):

```
${CROSS_COMPILE}objdump -h $KERNEL_ELF | sort -k 5,5 | awk -n '
/^ +[0-9]+ / {addr="0x"$3; size="0x"$5; printf "0x""%x\n",addr+size}' |
(( `tail -1` > (FW_JUMP_FDT_ADDR - FW_JUMP_ADDR) )) &&
```

```
echo fdt overlaps kernel, increase FW_JUMP_FDT_ADDR

${LLVM}objdump -h --show-lma $KERNEL_ELF | sort -k 5,5 | awk -n '
/^ +[0-9]+ / {addr="0x"$3; size="0x"$5; printf "0x""%x\n",addr+size}' |
(( `tail -1` > (FW_JUMP_FDT_ADDR - FW_JUMP_ADDR) )) &&
echo fdt overlaps kernel, increase FW_JUMP_FDT_ADDR
```

- **FW_JUMP_FDT_OFFSET** - Address offset from the opensbi load address where the FDT will be passed to the next booting stage. This offset is used as relocatable address of the FDT passed to the next booting stage. If *FW_JUMP_FDT_ADDR* is also defined, the firmware will prefer *FW_JUMP_FDT_ADDR*.

FW_JUMP Example

The *[qemu/virt]* platform illustrates how to configure and use a *FW_JUMP* firmware. Detailed information regarding these platforms can be found in the platform documentation files.

Chapter 12

OpenSBI Firmware with Payload (FW_PAYLOAD)

OpenSBI **firmware with Payload (FW_PAYLOAD)** is a firmware which directly includes the binary for the booting stage to follow the OpenSBI firmware execution. Typically, this payload will be a bootloader or an OS kernel.
A *FW_PAYLOAD* firmware is particularly useful when the booting stage executed prior to the OpenSBI firmware is not capable of loading both the OpenSBI firmware and the booting stage to follow OpenSBI firmware.
A *FW_PAYLOAD* firmware is also useful for cases where the booting stage prior to the OpenSBI firmware does not pass a *flattened device tree (FDT file)*. In such a case, a *FW_PAYLOAD* firmware allows embedding a flattened device tree in the .rodata section of the final firmware.

Enabling *FW_PAYLOAD* compilation

The *FW_PAYLOAD* firmware can be enabled by any of the following methods:

1. Specifying `FW_PAYLOAD=y` on the top level `make` command line.
2. Specifying `FW_PAYLOAD=y` in the target platform *objects.mk* configuration file.

The compiled *FW_PAYLOAD* firmware ELF file is named *fw_payload.elf*. Its expanded image file is *fw_payload.bin*. Both files are created in the platform-specific build directory under the *build/platform/<platform_subdir>/firmware* directory.

Configuration Options

A *FW_PAYLOAD* firmware is built according to configuration parameters and options. These configuration parameters can be defined using either the top level `make` command line or the target platform *objects.mk* configuration file. The parameters currently defined are as follows:

- **FW_PAYLOAD_OFFSET** - Offset from the opensbi load address where the payload binary will be linked in the final *FW_PAYLOAD* firmware binary image. This configuration parameter is mandatory if *FW_PAYLOAD_ALIGN* is not defined. Compilation errors will result from an incorrect definition of *FW_PAYLOAD_OFFSET* or of *FW_PAYLOAD_ALIGN*, or if neither of these parameters are defined.
- **FW_PAYLOAD_ALIGN** - Address alignment constraint where the payload binary will be linked after the end of the base firmware binary in the final *FW_PAYLOAD* firmware binary image. This configuration parameter is mandatory if *FW_PAYLOAD_OFFSET* is not defined. If both *FW_PAYLOAD_OFFSET* and *FW_PAYLOAD_ALIGN* are defined, *FW_PAYLOAD_OFFSET* is used and *FW_PAYLOAD_ALIGN* is ignored.
- **FW_PAYLOAD_PATH** - Path to the image file of the next booting stage binary. If this option is not provided then a simple test payload is automatically generated and used as a payload. This test payload executes an infinite `while (1)` loop after printing a message on the platform console.
- **FW_PAYLOAD_FDT_ADDR** - Address where the FDT passed by the prior booting stage or specified by the *FW_FDT_PATH* parameter and embedded in the ∗.rodata∗ section will be placed before executing the next

booting stage, that is, the payload firmware. If this option is not provided, then the firmware will pass the FDT address passed by the previous booting stage to the next booting stage.

- **FW_PAYLOAD_FDT_OFFSET** - Address offset from the opensbi load address where the FDT will be passed to the next booting stage. This offset is used as relocatable address of the FDT passed to the next booting stage. If *FW_PAYLOAD_FDT_ADDR* is also defined, the firmware will prefer *FW_PAYLOAD_FDT_ADDR*.

FW_PAYLOAD Example

The *[qemu/virt]* platforms illustrate how to configure and use a *FW_PAYLOAD* firmware. Detailed information regarding these platforms can be found in the platform documentation files.

Chapter 13

Linux as a direct payload to OpenSBI

OpenSBI has the capability to load a Linux kernel image directly in supervisor mode. The flattened image generated by the Linux kernel build process can be provided as a payload to OpenSBI.
Detailed examples can be found in both the QEMU and the HiFive Unleashed platform guides.

Chapter 14

U-Boot as a payload to OpenSBI

`U-Boot` is an open-source primary boot loader. It can be used as first and/or second stage boot loader in an embedded environment. In the context of OpenSBI, U-Boot can be specified as a payload to the OpenSBI firmware, becoming the boot stage following the OpenSBI firmware execution.

Building and Generating U-Boot images

Please refer to the U-Boot build documentation for detailed instructions on how to build U-Boot image and boot high level operating systems from U-Boot prompt.

Chapter 15

Andes AE350 SoC Platform

The AE350 AXI/AHB-based platform N25(F)/NX25(F)/D25F/A25/AX25 CPU with level-one memories, interrupt controller, debug module, AXI and AHB Bus Matrix Controller, AXI-to-AHB Bridge and a collection of fundamental AHB/APB bus IP components pre-integrated together as a system design. The high-quality and configurable AHB/↩ APB IPs suites a majority embedded systems, and the verified platform serves as a starting point to jump start SoC designs.
To build platform specific library and firmwares, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The Andes AE350 platform does not have any platform-specific options.

Building Andes AE350 Platform

AE350's dts is included in https://github.com/andestech/linux/tree/RISCV-Linux-5.↩ 4-ast-v5_1_0-branch

Linux Kernel Payload

```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image FW_FDT_PATH=<ae350.dtb
      path>
```

DTS Example: (Quad-core AX45MP)

```
compatible = "andestech,ae350";
cpus {
    #address-cells = <1>;
    #size-cells = <0>;
    timebase-frequency = <60000000>;

    CPU0: cpu@0 {
        device_type = "cpu";
        reg = <0>;
        status = "okay";
        compatible = "riscv";
        riscv,isa = "rv64imafdc";
        riscv,priv-major = <1>;
        riscv,priv-minor = <10>;
        mmu-type = "riscv,sv48";
        clock-frequency = <60000000>;
        i-cache-size = <0x8000>;
        i-cache-sets = <256>;
        i-cache-line-size = <64>;
        i-cache-block-size = <64>;
        d-cache-size = <0x8000>;
        d-cache-sets = <128>;
        d-cache-line-size = <64>;
        d-cache-block-size = <64>;
        next-level-cache = <&L2>;
        CPU0_intc: interrupt-controller {
            #interrupt-cells = <1>;
            interrupt-controller;
            compatible = "riscv,cpu-intc";
        };
```

```
    };
    CPU1: cpu@1 {
        device_type = "cpu";
        reg = <1>;
        status = "okay";
        compatible = "riscv";
        riscv,isa = "rv64imafdc";
        riscv,priv-major = <1>;
        riscv,priv-minor = <10>;
        mmu-type = "riscv,sv48";
        clock-frequency = <60000000>;
        i-cache-size = <0x8000>;
        i-cache-sets = <256>;
        i-cache-line-size = <64>;
        i-cache-block-size = <64>;
        d-cache-size = <0x8000>;
        d-cache-sets = <128>;
        d-cache-line-size = <64>;
        d-cache-block-size = <64>;
        next-level-cache = <&L2>;
        CPU1_intc: interrupt-controller {
            #interrupt-cells = <1>;
            interrupt-controller;
            compatible = "riscv,cpu-intc";
        };
    };
    CPU2: cpu@2 {
        device_type = "cpu";
        reg = <2>;
        status = "okay";
        compatible = "riscv";
        riscv,isa = "rv64imafdc";
        riscv,priv-major = <1>;
        riscv,priv-minor = <10>;
        mmu-type = "riscv,sv48";
        clock-frequency = <60000000>;
        i-cache-size = <0x8000>;
        i-cache-sets = <256>;
        i-cache-line-size = <64>;
        i-cache-block-size = <64>;
        d-cache-size = <0x8000>;
        d-cache-sets = <128>;
        d-cache-line-size = <64>;
        d-cache-block-size = <64>;
        next-level-cache = <&L2>;
        CPU2_intc: interrupt-controller {
            #interrupt-cells = <1>;
            interrupt-controller;
            compatible = "riscv,cpu-intc";
        };
    };
    CPU3: cpu@3 {
        device_type = "cpu";
        reg = <3>;
        status = "okay";
        compatible = "riscv";
        riscv,isa = "rv64imafdc";
        riscv,priv-major = <1>;
        riscv,priv-minor = <10>;
        mmu-type = "riscv,sv48";
        clock-frequency = <60000000>;
        i-cache-size = <0x8000>;
        i-cache-sets = <256>;
        i-cache-line-size = <64>;
        i-cache-block-size = <64>;
        d-cache-size = <0x8000>;
        d-cache-sets = <128>;
        d-cache-line-size = <64>;
        d-cache-block-size = <64>;
        next-level-cache = <&L2>;
        CPU3_intc: interrupt-controller {
            #interrupt-cells = <1>;
            interrupt-controller;
            compatible = "riscv,cpu-intc";
        };
    };
};

soc {
    #address-cells = <2>;
    #size-cells = <2>;
    compatible = "andestech,riscv-ae350-soc", "simple-bus";
    ranges;

    plic0: interrupt-controller@e4000000 {
        compatible = "riscv,plic0";
        reg = <0x00000000 0xe4000000 0x00000000 0x02000000>;
```

```
        interrupts-extended = < &CPU0_intc 11 &CPU0_intc 9
                                &CPU1_intc 11 &CPU1_intc 9
                                &CPU2_intc 11 &CPU2_intc 9
                                &CPU3_intc 11 &CPU3_intc 9 >;
        interrupt-controller;
        #address-cells = <2>;
        #interrupt-cells = <2>;
        riscv,ndev = <71>;
    };

    plicsw: interrupt-controller@e6400000 {
        compatible = "andestech,plicsw";
        reg = <0x00000000 0xe6400000 0x00000000 0x00400000>;
        interrupts-extended = < &CPU0_intc 3
                                &CPU1_intc 3
                                &CPU2_intc 3
                                &CPU3_intc 3 >;
        interrupt-controller;
        #address-cells = <2>;
        #interrupt-cells = <2>;
    };

    plmt0: plmt0@e6000000 {
        compatible = "andestech,plmt0";
        reg = <0x00000000 0xe6000000 0x00000000 0x00100000>;
        interrupts-extended = < &CPU0_intc 7
                                &CPU1_intc 7
                                &CPU2_intc 7
                                &CPU3_intc 7 >;
    };

    wdt: watchdog@f0500000 {
        compatible = "andestech,atcwdt200";
        reg = <0x00000000 0xf0500000 0x00000000 0x00001000>;
        interrupts = <3 4>;
        interrupt-parent = <&plic0>;
        clock-frequency = <15000000>;
    };

    serial0: serial@f0300000 {
        compatible = "andestech,uart16550", "ns16550a";
        reg = <0x00000000 0xf0300000 0x00000000 0x00001000>;
        interrupts = <9 4>;
        interrupt-parent = <&plic0>;
        clock-frequency = <19660800>;
        current-speed = <38400>;
        reg-shift = <2>;
        reg-offset = <32>;
        reg-io-width = <4>;
        no-loopback-test = <1>;
    };

    smu: smu@f0100000 {
        compatible = "andestech,atcsmu";
        reg = <0x00000000 0xf0100000 0x00000000 0x00001000>;
    };
};
```

Chapter 16

Ariane FPGA SoC Platform

Ariane is a 6-stage, single issue, in-order CPU which implements the 64-bit RISC-V instruction set. The Ariane FPGA development platform is based on FPGA SoC (which currently supports only Genesys 2 board) and is capable of running Linux.
The FPGA SoC currently contains the following peripherals:

- DDR3 memory controller
- SPI controller to connect to an SDCard
- Ethernet controller
- JTAG port (see debugging section below)
- Bootrom containing zero stage bootloader and device tree.

To build platform specific library and firmwares, provide the *PLATFORM=fpga/ariane* parameter to the top level `make` command.

Platform Options

The *Ariane FPGA* platform does not have any platform-specific options.

Building Ariane FPGA Platform

Linux Kernel Payload

```
make PLATFORM=fpga/ariane FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```

Booting Ariane FPGA Platform

Linux Kernel Payload

As Linux kernel image is embedded in the OpenSBI firmware binary, Ariane will directly boot into Linux directly after powered on.

Chapter 17

OpenPiton FPGA SoC Platform

OpenPiton is the world's first open source, general purpose, multithreaded manycore processor. It is a tiled manycore framework scalable from one to 1/2 billion cores. Currently, OpenPiton supports the 64bit Ariane RISC-V processor from ETH Zurich. To this end, Ariane has been equipped with a different L1 cache subsystem that follows a write-through protocol and that has support for cache invalidations and atomics.
To build platform specific library and firmwares, provide the *PLATFORM=fpga/openpiton* parameter to the top level `make` command.

Platform Options

The *OpenPiton* platform does not have any platform-specific options.

Building Ariane FPGA Platform

Linux Kernel Payload

```
make PLATFORM=fpga/openpiton FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```

Booting Ariane FPGA Platform

Linux Kernel Payload

As Linux kernel image is embedded in the OpenSBI firmware binary, Ariane will directly boot into Linux directly after powered on.

Chapter 18

Generic Platform

The **Generic** platform is a flattened device tree (FDT) based platform where all platform specific functionality is provided based on FDT passed by previous booting stage. The **Generic** platform allows us to use same OpenSBI firmware binaries on various emulators, simulators, FPGAs, and boards.
By default, the generic FDT platform makes following assumptions:

1. platform features are default
2. platform stack size is default
3. platform has no quirks or work-arounds

The above assumptions (except 1) can be overridden by adding special platform callbacks which will be called based on FDT root node compatible string.
Users of the generic FDT platform will have to ensure that:

1. Various FDT based drivers under lib/utils directory are upto date based on their platform requirements
2. The FDT passed by previous booting stage has DT compatible strings and DT properties in sync with the FDT based drivers under lib/utils directory
3. The FDT must have "stdout-path" DT property in the "/chosen" DT node when a platform has multiple serial ports or consoles
4. On multi-HART platform, the FDT must have a DT node for IPI device and lib/utils/ipi directory must have corresponding FDT based IPI driver
5. The FDT must have a DT node for timer device and lib/utils/timer directory must have corresponding FDT based timer driver

To build the platform-specific library and firmware images, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The *Generic* platform does not have any platform-specific options.

RISC-V Platforms Using Generic Platform

- **Andes AE350 Platform** (*[andes-ae350.md]*)
- **QEMU RISC-V Virt Machine** (*[qemu_virt.md]*)
- **Renesas RZ/Five SoC** (*[renesas-rzfive.md]*)
- **Shakti C-class SoC Platform** (*[shakti_cclass.md]*)
- **SiFive HiFive Unleashed** (*[sifive_fu540.md]*)

- **Spike** ([*spike.md*])
- **T-HEAD C9xx series Processors** ([*thead-c9xx.md*])

Chapter 19

Nuclei UX600 Platform

The **Nuclei UX600** is a 64-bit RISC-V Core which is capable of running Linux.

> Nuclei UX600: single core, pipeline as single-issue and 6∼9 variable stages, in-order dispatch and out-of-order write-back, running up to >1.2GHz

To build the platform-specific library and firmware images, provide the *PLATFORM=nuclei/ux600* parameter to the top level `make` command.

Platform Options

The *Nuclei UX600* platform does not have any platform-specific options.

Building Nuclei UX600 Platform

```
make PLATFORM=nuclei/ux600 clean all
```

Chapter 20

OpenSBI Supported Platforms

OpenSBI currently supports the following virtual and hardware platforms:

- **Generic**: Flattened device tree (FDT) based platform where platform specific functionality is provided based on the FDT passed by previous booting stage. More details on this platform can be found in the file *[generic.md]*.
- **QEMU RISC-V Virt Machine**: Platform support for the QEMU *virt* virtual RISC-V machine. This virtual machine is intended for RISC-V software development and tests. More details on this platform can be found in the file *[qemu_virt.md]*.
- **SiFive FU540 SoC**: Platform support for SiFive FU540 SoC used on the HiFive Unleashed board, as well as the *sifive_u* QEMU virtual RISC-V machine. More details on this platform can be found in the file *[sifive_fu540.md]*.
- **Kendryte K210 SoC**: Platform support for the Kendryte K210 SoC used on boards such as the Kendryte KD233 or the Sipeed MAIX Dock.
- **Ariane FPGA SoC**: Platform support for the Ariane FPGA SoC used on Genesys 2 board. More details on this platform can be found in the file *[fpga-ariane.md]*.
- **Andes AE350 SoC**: Platform support for the Andes's SoC (AE350). More details on this platform can be found in the file *[andes-ae350.md]*.
- **Spike**: Platform support for the Spike emulator. More details on this platform can be found in the file *[spike.md]*.
- **OpenPiton FPGA SoC**: Platform support OpenPiton research platform based on ariane core. More details on this platform can be found in the file *[fpga-openpiton.md]*.
- **Shakti C-class SoC Platform**: Platform support for Shakti C-class processor based SOCs. More details on this platform can be found in the file *[shakti_cclass.md]*.
- **Renesas RZ/Five SoC**: Platform support for Renesas RZ/Five (R9A07G043F) SoC used on the Renesas RZ/Five SMARC EVK board. More details on this platform can be found in the file *[renesas-rzfive.md]*.

The code for these supported platforms can be used as example to implement support for other platforms. The *platform/template* directory also provides template files for implementing support for a new platform. The *objects.mk*, *Kconfig*, *configs/defconfig* and *platform.c* template files provides enough comments to facilitate the implementation.

Chapter 21

QEMU RISC-V Virt Machine Platform

The **QEMU RISC-V Virt Machine** is a virtual platform created for RISC-V software development and testing. It is also referred to as *QEMU RISC-V VirtIO machine* because it uses VirtIO devices for network, storage, and other types of IO.
To build the platform-specific library and firmware images, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The *QEMU RISC-V Virt Machine* platform does not have any platform-specific options.

Execution on QEMU RISC-V 64-bit

No Payload Case
Build:
```
make PLATFORM=generic
```
Run:
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.bin
```
U-Boot Payload
Note: the command line examples here assume that U-Boot was compiled using the `qemu-riscv64_smode↩_defconfig` configuration.
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<uboot_build_directory>/u-boot.bin
```
Run:
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf
```
or
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <uboot_build_directory>/u-boot.bin
```
Linux Kernel Payload
Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/defconfig*. The kernel must be a flattened image (a file called `Image`) rather than an ELF (`vmlinux`).
Example of building a Linux kernel:
```
make ARCH=riscv CROSS_COMPILE=riscv64-linux- defconfig
make ARCH=riscv CROSS_COMPILE=riscv64-linux- Image
```
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```
Run:
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf \
    -drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \
    -device virtio-blk-device,drive=hd0 \
    -append "root=/dev/vda rw console=ttyS0"
```
or
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <linux_build_directory>/arch/riscv/boot/Image \
    -drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \
```

```
    -device virtio-blk-device,drive=hd0 \
    -append "root=/dev/vda rw console=ttyS0"
```

Execution on QEMU RISC-V 32-bit

No Payload Case

Build:

```
make PLATFORM=generic PLATFORM_RISCV_XLEN=32
```

Run:

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.bin
```

U-Boot Payload

Note: the command line examples here assume that U-Boot was compiled using the `qemu-riscv32_smode↩_defconfig` configuration.

Build:

```
make PLATFORM=generic PLATFORM_RISCV_XLEN=32 FW_PAYLOAD_PATH=<uboot_build_directory>/u-boot.bin
```

Run:

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf
```

or

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <uboot_build_directory>/u-boot.bin
```

Linux Kernel Payload

Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/rv32_defconfig*.

Build:

```
make PLATFORM=generic PLATFORM_RISCV_XLEN=32 FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```

Run:

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf \
    -drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \
    -device virtio-blk-device,drive=hd0 \
    -append "root=/dev/vda rw console=ttyS0"
```

or

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <linux_build_directory>/arch/riscv/boot/Image \
    -drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \
    -device virtio-blk-device,drive=hd0 \
    -append "root=/dev/vda rw console=ttyS0"
```

Debugging with GDB

In a first console start OpenSBI with QEMU:

```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.bin \
    -gdb tcp::1234 \
    -S
```

Parameter *-gdb tcp::1234* specifies 1234 as the debug port. Parameter *-S* lets QEMU wait at the first instruction.

In a second console start GDB:

```
gdb build/platform/generic/firmware/fw_payload.elf \
    -ex 'target remote localhost:1234'
```

Chapter 22

Renesas RZ/Five SoC (R9A07G043F) Platform

The RZ/Five microprocessor includes a single RISC-V CPU Core (Andes AX45MP) 1.0 GHz, 16-bit DDR3L/DDR4 interface. Supported interfaces include:

- Memory controller for DDR4-1600 / DDR3L-1333 with 16 bits
- System RAM (RAM of 128 Kbytes (ECC))
- SPI Multi I/O Bus Controller 1ch
- SD Card Host Interface/Multimedia Card Interface (SD/MMC) 2ch
- Serial Sound Interface (SSI) 4ch
- Sampling Rate Converter (SRC) 1ch
- USB2.0 host/function interface 2ch (ch0: Host-Function ch1: Host only)
- Gigabit Ethernet Interface (GbE) 2ch
- Controller Area Network Interface (CAN) 2ch (CAN-FD ISO 11898-1 (CD2014) compliant)
- Multi-function Timer Pulse Unit 3 (MTU3a) 9 ch (16 bits × 8 channels, 32 bits × 1 channel)
- Port Output Enable 3 (POE3)
- Watchdog Timer (WDT) 1ch
- General Timer (GTM) 3ch (32bits)
- I2C Bus Interface (I2C) 4ch
- Serial Communication Interface with FIFO (SCIFA) 5ch
- Serial Communication Interface (SCI) 2ch
- Renesas Serial Peripheral Interface (RSPI) 3ch
- A/D Converter (ADC) 2ch making it ideal for applications such as entry-class social infrastructure gateway control and industrial gateway control. More details can be found at below link [0].

[0] https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz-mpus/rzfiv
To build platform specific library and firmwares, provide the *PLATFORM=generic* parameter to the top level make command.

Platform Options

The Renesas RZ/Five platform does not have any platform-specific options.

Building Renesas RZ/Five Platform

```
make PLATFORM=generic
```

DTS Example: (RZ/Five AX45MP)

```
compatible = "renesas,r9a07g043f01", "renesas,r9a07g043";

cpus {
    #address-cells = <1>;
    #size-cells = <0>;
    timebase-frequency = <12000000>;

    cpu0: cpu@0 {
        compatible = "andestech,ax45mp", "riscv";
        device_type = "cpu";
        reg = <0x0>;
        status = "okay";
        riscv,isa = "rv64imafdc";
        mmu-type = "riscv,sv39";
        i-cache-size = <0x8000>;
        i-cache-line-size = <0x40>;
        d-cache-size = <0x8000>;
        d-cache-line-size = <0x40>;
        clocks = <&cpg CPG_CORE R9A07G043_CLK_I>;

        cpu0_intc: interrupt-controller {
            #interrupt-cells = <1>;
            compatible = "riscv,cpu-intc";
            interrupt-controller;
        };
    };
};

soc {
    compatible = "simple-bus";
    #address-cells = <1>;
    #size-cells = <0>;
    ranges;

    scif0: serial@1004b800 {
        compatible = "renesas,scif-r9a07g043",
                 "renesas,scif-r9a07g044";
        reg = <0 0x1004b800 0 0x400>;
        interrupts = <412 IRQ_TYPE_LEVEL_HIGH>,
                 <414 IRQ_TYPE_LEVEL_HIGH>,
                 <415 IRQ_TYPE_LEVEL_HIGH>,
                 <413 IRQ_TYPE_LEVEL_HIGH>,
                 <416 IRQ_TYPE_LEVEL_HIGH>,
                 <416 IRQ_TYPE_LEVEL_HIGH>;
        interrupt-names = "eri", "rxi", "txi",
                  "bri", "dri", "tei";
        clocks = <&cpg CPG_MOD R9A07G043_SCIF0_CLK_PCK>;
        clock-names = "fck";
        power-domains = <&cpg>;
        resets = <&cpg R9A07G043_SCIF0_RST_SYSTEM_N>;
        status = "disabled";
    };

    cpg: clock-controller@11010000 {
        compatible = "renesas,r9a07g043-cpg";
        reg = <0 0x11010000 0 0x10000>;
        clocks = <&extal_clk>;
        clock-names = "extal";
        #clock-cells = <2>;
        #reset-cells = <1>;
        #power-domain-cells = <0>;
    };

    sysc: system-controller@11020000 {
        compatible = "renesas,r9a07g043-sysc";
        reg = <0 0x11020000 0 0x10000>;
        status = "disabled";
    };

    pinctrl: pinctrl@11030000 {
        compatible = "renesas,r9a07g043-pinctrl";
        reg = <0 0x11030000 0 0x10000>;
        gpio-controller;
        #gpio-cells = <2>;
        #interrupt-cells = <2>;
        interrupt-controller;
        gpio-ranges = <&pinctrl 0 0 152>;
        clocks = <&cpg CPG_MOD R9A07G043_GPIO_HCLK>;
```

```
        power-domains = <&cpg>;
        resets = <&cpg R9A07G043_GPIO_RSTN>,
             <&cpg R9A07G043_GPIO_PORT_RESETN>,
             <&cpg R9A07G043_GPIO_SPARE_RESETN>;
    };

    plmt0: plmt0@110c0000 {
        compatible = "andestech,plmt0", "riscv,plmt0";
        reg = <0x0 0x110c0000 0x0 0x10000>;
        interrupts-extended = <&cpu0_intc 7>;
    };

    plic: interrupt-controller@12c00000 {
        compatible = "renesas,r9a07g043-plic", "andestech,nceplic100";
        #interrupt-cells = <2>;
        #address-cells = <0>;
        riscv,ndev = <511>;
        interrupt-controller;
        reg = <0x0 0x12c00000 0x0 0x400000>;
        clocks = <&cpg CPG_MOD R9A07G043_NCEPLIC_ACLK>;
        power-domains = <&cpg>;
        resets = <&cpg R9A07G043_NCEPLIC_ARESETN>;
        interrupts-extended = <&cpu0_intc 11 &cpu0_intc 9>;
    };

    plicsw: interrupt-controller@13000000 {
        compatible = "andestech,plicsw";
        reg = <0x0 0x13000000 0x0 0x400000>;
        interrupts-extended = <&cpu0_intc 3>;
        interrupt-controller;
        #address-cells = <2>;
        #interrupt-cells = <2>;
    };
};
```

Chapter 23

Shakti C-class SoC Platform

C-Class is a member of the SHAKTI family of processors from Indian Institute of Technology - Madras (IIT-M).
It is an extremely configurable and commercial-grade 5-stage in-order core supporting the standard RV64GCSUN ISA extensions.
For more details, refer:

- https://gitlab.com/shaktiproject/cores/c-class/blob/master/README.md
- https://c-class.readthedocs.io/en/latest
- https://shakti.org.in

Platform Options

The *Shakti C-class SoC* platform does not have any platform-specific options.

Building Shakti C-class Platform

Linux Kernel Payload

```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image FW_FDT_PATH=<shakti.dtb
      path>
```

Test Payload

```
make PLATFORM=generic FW_FDT_PATH=<shakti.dtb path>
```

Chapter 24

SiFive FU540 SoC Platform

The FU540-C000 is the world's first 4+1 64-bit RISC-V SoC from SiFive. The HiFive Unleashed development platform is based on FU540-C000 and capable of running Linux.
With QEMU v4.2 or above release, the 'sifive_u' machine can be used to test OpenSBI image built for the real hardware as well.
To build platform specific library and firmwares, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The *SiFive FU540 SoC* platform does not have any platform-specific options.

Building SiFive Fu540 Platform

In order to boot SMP Linux in U-Boot, Linux v5.1 (or higher) and latest U-Boot v2020.01 (or higher) should be used.
Linux Kernel Payload
The HiFive Unleashed device tree(DT) is merged in Linux v5.2 release. This DT (device tree) is not backward compatible with the DT passed from FSBL.
To use Linux v5.2 (or higher), the pre-built DTB (DT binary) from Linux v5.2 (or higher) should be used to build SiFive FU540 OpenSBI binaries by using the compile time option *FW_FDT_PATH*.

```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
or
(For Linux v5.2 or higher)
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
      FW_FDT_PATH=<hifive-unleashed-a00.dtb path from Linux kernel>
```

U-Boot Payload
The command-line example here assumes that U-Boot was compiled using the sifive_fu540_defconfig configuration and with U-Boot v2020.01, and up to v2021.04. sifive_unleashed_defconfig shall be used with v2021.07 or above.

```
make PLATFORM=generic FW_PAYLOAD_PATH=<u-boot_build_dir>/u-boot-dtb.bin
```

For U-Boot v2020.07-rc4 or later releases, SPL support was added in U-Boot. Please refer to the detailed U-Boot booting guide available at `U-Boot`.

Flashing the OpenSBI firmware binary to storage media:

The first stage boot loader (`FSBL`) expects the storage media to have a GPT partition table. It tries to look for a partition with following GUID to load the next stage boot loader (OpenSBI in this case).

```
2E54B353-1271-4842-806F-E436D6AF6985
```

That's why the generated firmware binary in above steps should be copied to the partition of the sdcard with above GUID.

```
dd if=build/platform/generic/firmware/fw_payload.bin of=/dev/disk2s1 bs=1024
```

In my case, it is the first partition is **disk2s1** that has been formatted with the above specified GUID.
In case of a brand new sdcard, it should be formatted with below partition tables as described here.

```
sgdisk --clear                                                                  \
       --new=1:2048:67583  --change-name=1:bootloader --typecode=1:2E54B353-1271-4842-806F-E436D6AF6985   \
       --new=2:264192:     --change-name=2:root       --typecode=2:0FC63DAF-8483-4772-8E79-3D69D8477DE4 \
       ${DISK}
```

Booting SiFive Fu540 Platform

Linux Kernel Payload
As Linux kernel image is embedded in the OpenSBI firmware binary, HiFive Unleashed will directly boot into Linux directly after powered on.
U-Boot Payload
As U-Boot image is used as payload, HiFive Unleashed will boot into a U-Boot prompt. U-Boot tftp boot method can be used to load kernel image in U-Boot prompt. Here are the steps do a tftpboot.

1. Set the ip address of the board.
   ```
   setenv ipaddr <ipaddr of the board>
   ```
2. Set the tftpboot server IP.
   ```
   setenv serverip <ipaddr of the tftp server>
   ```
3. Set the network gateway address.
   ```
   setenv gatewayip <ipaddress of the network gateway>
   ```
4. Load the Linux kernel image from the tftp server.
   ```
   tftpboot ${kernel_addr_r} <Image path in tftpboot directory>
   ```
5. Load the ramdisk image from the tftp server. This is only required if ramdisk is loaded from tftp server. This step is optional, if rootfs is already part of the kernel or loaded from an external storage by kernel.
   ```
   tftpboot ${ramdisk_addr_r} <ramdisk path in tftpboot directory>
   ```
6. Load the pre-compiled device tree via tftpboot.
   ```
   tftpboot ${fdt_addr_r} <hifive-unleashed-a00.dtb path in tftpboot directory>
   ```
7. Set the boot command-line arguments.
   ```
   setenv bootargs "root=<root partition> rw console=ttySIF0 earlycon=sbi"
   ```
   (Note: root partition should point to ∗∗ /dev/ram ∗∗ - If a ramdisk is used ∗∗ root=/dev/mmcblk0pX ∗∗ - If a rootfs is already on some other partition of sdcard)
8. Now boot into Linux.
   ```
   booti ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r}
   or
   (If ramdisk is not loaded from network)
   booti ${kernel_addr_r} - ${fdt_addr_r}
   ```

U-Boot & Linux Kernel as a single payload
At U-Boot prompt execute the following boot command to boot Linux.
```
booti ${kernel_addr_r} - ${fdt_addr_r}
```

QEMU Specific Instructions

If you want to test OpenSBI with QEMU 'sifive_u' machine, please follow the same instructions above, with the exception of not passing FW_FDT_PATH.
This is because QEMU generates a device tree blob on the fly based on the command line parameters, and it's compatible with the one used in the upstream Linux kernel.
When U-Boot v2021.07 (or higher) is used as the payload, as the SiFive FU540 DTB for the real hardware is embedded in U-Boot binary itself, due to the same reason above, we need to switch the U-Boot sifive_unleashed↩_defconfig configuration from **CONFIG_OF_SEPARATE** to **CONFIG_OF_PRIOR_STAGE** so that U-Boot uses the DTB generated by QEMU, and u-boot.bin should be used as the payload image, like:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<u-boot_build_dir>/u-boot.bin
```
U-Boot v2020.07 release added SPL support to SiFive HiFive Unleashed board, hence a build error will be seen after you switch to **CONFIG_OF_PRIOR_STAGE**.
```
./tools/mkimage: Can't open arch/riscv/dts/hifive-unleashed-a00.dtb: No such file or directory
./tools/mkimage: failed to build FIT
Makefile:1402: recipe for target 'u-boot.img' failed
make: *** [u-boot.img] Error 1
```
The above errors can be safely ignored as we don't run U-Boot SPL under QEMU.
Run:
```
qemu-system-riscv64 -M sifive_u -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.bin
```
or
```
qemu-system-riscv64 -M sifive_u -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <uboot_build_dir>/u-boot.bin
```

While the real hardware operates at the 64-bit mode, it's possible for QEMU to test the 32-bit OpenSBI firmware. This can be helpful for testing 32-bit SiFive specific drivers.

Chapter 25

Spike Simulator Platform

The **Spike** is a RISC-V ISA simulator which implements a functional model of one or more RISC-V harts. The **Spike** compatible virtual platform is also available on QEMU. In fact, we can use same OpenSBI firmware binaries on **Spike** simulator and QEMU Spike machine.
For more details, refer `Spike on GitHub`
To build the platform-specific library and firmware images, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The *Spike* platform does not have any platform-specific options.

Execution on Spike Simulator

No Payload Case
Build:
```
make PLATFORM=generic
```
Run:
```
spike build/platform/generic/firmware/fw_payload.elf
```
Linux Kernel Payload
Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/defconfig*.
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```
Run:
```
spike -m256 \
    --initrd <path_to_cpio_ramdisk> \
    --bootargs 'root=/dev/ram rw console=hvc0 earlycon=sbi' \
    build/platform/generic/firmware/fw_payload.elf
```
or
```
spike -m256 \
    --kernel <linux_build_directory>/arch/riscv/boot/Image \
    --initrd <path_to_cpio_ramdisk> \
    --bootargs 'root=/dev/ram rw console=hvc0 earlycon=sbi' \
    build/platform/generic/firmware/fw_jump.elf
```

Execution on QEMU RISC-V 64-bit

No Payload Case
Build:
```
make PLATFORM=generic
```
Run:
```
qemu-system-riscv64 -M spike -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf
```
Linux Kernel Payload
Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/defconfig*.
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```
Run:

```
qemu-system-riscv64 -M spike -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf \
    -initrd <path_to_cpio_ramdisk> \
    -append "root=/dev/ram rw console=hvc0 earlycon=sbi"
```

or

```
qemu-system-riscv64 -M spike -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.elf \
    -kernel <linux_build_directory>/arch/riscv/boot/Image \
    -initrd <path_to_cpio_ramdisk> \
    -append "root=/dev/ram rw console=hvc0 earlycon=sbi"
```

Chapter 26

T-HEAD C9xx Series Processors

The C9xx series processors are high-performance RISC-V architecture multi-core processors with AI vector acceleration engine.
For more details, refer `T-HEAD.CN`
To build the platform-specific library and firmware images, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The T-HEAD C9xx does not have any platform-specific compile options because it uses generic platform.
```
CROSS_COMPILE=riscv64-linux-gnu- PLATFORM=generic make
```
Here is the simplest boot flow for a fpga prototype:
(Jtag gdbinit) -> (zsb) -> (opensbi) -> (linux)
For more details, refer: `zero stage boot`

Chapter 27

Data Structure Index

27.1 Data Structures

Here are the data structures with brief descriptions:

__aligned . . . 77
aclint_mswi_data . . . 77
aclint_mtimer_data . . . 78
aplic_data . . . 79
aplic_delegate_data . . . 80
aplic_msicfg_data . . . 81
atomic_t . . . 81
dw_i2c_adapter . . . 81
fdt_driver . . . 82
fdt_general_fixup . . . 82
fdt_gpio . . . 83
fdt_header . . . 83
fdt_mailbox . . . 84
fdt_match . . . 85
fdt_node_header . . . 85
fdt_phandle_args . . . 86
fdt_pmu_hw_event_counter_map . . . 86
fdt_pmu_hw_event_select_map . . . 86
fdt_pmu_raw_event_counter_map . . . 87
fdt_property . . . 87
fdt_reserve_entry . . . 88
fw_dynamic_info . . . 88
gpio_chip . . . 89
gpio_pin . . . 91
i2c_adapter . . . 91
imsic_data . . . 92
imsic_regs . . . 93
mbox_chan . . . 93
mbox_controller . . . 94
mbox_xfer . . . 95
platform_uart_data . . . 97
plic_data . . . 97
plicsw_data . . . 98
plmt_data . . . 99
regmap . . . 99
rpmi_base_get_attributes_resp . . . 100
rpmi_base_get_platform_info_resp . . . 101
rpmi_clock_get_attributes_req . . . 101
rpmi_clock_get_attributes_resp . . . 102
rpmi_clock_get_config_req . . . 102

rpmi_clock_get_config_resp . . . 103
rpmi_clock_get_num_clocks_resp . . . 103
rpmi_clock_get_rate_req . . . 103
rpmi_clock_get_rate_resp . . . 104
rpmi_clock_get_supported_rates_req . . . 104
rpmi_clock_get_supported_rates_resp . . . 104
rpmi_clock_set_config_req . . . 105
rpmi_clock_set_config_resp . . . 105
rpmi_clock_set_rate_req . . . 106
rpmi_clock_set_rate_resp . . . 106
rpmi_cppc_get_fastchan_offset_req . . . 107
rpmi_cppc_get_fastchan_offset_resp . . . 107
rpmi_cppc_get_fastchan_region_resp . . . 107
rpmi_cppc_hart_list_req . . . 109
rpmi_cppc_hart_list_resp . . . 109
rpmi_cppc_probe_req . . . 109
rpmi_cppc_probe_resp . . . 110
rpmi_cppc_read_reg_req . . . 110
rpmi_cppc_read_reg_resp . . . 111
rpmi_cppc_write_reg_req . . . 111
rpmi_cppc_write_reg_resp . . . 112
rpmi_enable_notification_req . . . 112
rpmi_enable_notification_resp . . . 112
rpmi_hsm_get_hart_list_req . . . 113
rpmi_hsm_get_hart_list_resp . . . 113
rpmi_hsm_get_hart_status_req . . . 113
rpmi_hsm_get_hart_status_resp . . . 114
rpmi_hsm_get_susp_info_req . . . 114
rpmi_hsm_get_susp_info_resp . . . 114
rpmi_hsm_get_susp_types_req . . . 115
rpmi_hsm_get_susp_types_resp . . . 115
rpmi_hsm_hart_start_req . . . 116
rpmi_hsm_hart_start_resp . . . 116
rpmi_hsm_hart_stop_req . . . 117
rpmi_hsm_hart_stop_resp . . . 117
rpmi_hsm_hart_susp_req . . . 117
rpmi_hsm_hart_susp_resp . . . 118
rpmi_message . . . 118
rpmi_message_args . . . 119
rpmi_message_header . . . 119
rpmi_sysrst_get_reset_attributes_resp . . . 120
rpmi_syssusp_get_attr_req . . . 121
rpmi_syssusp_get_attr_resp . . . 121
rpmi_syssusp_suspend_req . . . 121
rpmi_syssusp_suspend_resp . . . 122
sbi_console_device . . . 122
sbi_cppc_device . . . 123
sbi_cpu_idle_state . . . 124
sbi_dbtr_data_msg . . . 124
sbi_dbtr_hart_triggers_state . . . 125
sbi_dbtr_id_msg . . . 126
sbi_dbtr_shmem . . . 126
sbi_dbtr_shmem_entry . . . 126
sbi_dbtr_trigger . . . 127
sbi_dlist . . . 128
sbi_domain . . . 128
sbi_domain_data . . . 130
sbi_domain_data_priv . . . 131

sbi_domain_memregion . . . 131
sbi_ecall_extension . . . 131
sbi_ecall_return . . . 132
sbi_fifo . . . 133
sbi_hart_ext_data . . . 133
sbi_hart_features . . . 134
sbi_hartmask . . . 135
sbi_hsm_device . . . 135
sbi_ipi_device . . . 136
sbi_ipi_event_ops . . . 137
sbi_irqchip_device . . . 138
sbi_ldst_data . . . 138
sbi_mpxy_channel . . . 139
sbi_mpxy_channel_attrs . . . 140
sbi_mpxy_msi_info . . . 141
sbi_platform . . . 142
sbi_platform_operations . . . 143
sbi_pmu_device . . . 146
sbi_pmu_event_info . . . 147
sbi_scratch . . . 148
sbi_sse_cb_ops . . . 149
sbi_system_reset_device . . . 150
sbi_system_suspend_device . . . 151
sbi_timer_device . . . 152
sbi_tlb_info . . . 153
sbi_trap_context . . . 153
sbi_trap_info . . . 154
sbi_trap_regs . . . 155
smu_data . . . 158
uu . . . 159

Chapter 28

File Index

28.1 File List

Here is a list of all files with brief descriptions:

include/sbi/fw_dynamic.h . . . 162
include/sbi/riscv_asm.h . . . 166
include/sbi/riscv_atomic.h . . . 178
include/sbi/riscv_barrier.h . . . 185
include/sbi/riscv_dbtr.h . . . 187
include/sbi/riscv_elf.h . . . 197
include/sbi/riscv_encoding.h . . . 197
include/sbi/riscv_fp.h . . . 291
include/sbi/riscv_io.h . . . 294
include/sbi/riscv_locks.h . . . 299
include/sbi/sbi_bitmap.h . . . 304
include/sbi/sbi_bitops.h . . . 312
include/sbi/sbi_byteorder.h . . . 323
include/sbi/sbi_console.h . . . 326
include/sbi/sbi_const.h . . . 334
include/sbi/sbi_cppc.h . . . 336
include/sbi/sbi_csr_detect.h . . . 339
include/sbi/sbi_dbtr.h . . . 342
include/sbi/sbi_domain.h . . . 351
include/sbi/sbi_domain_context.h . . . 370
include/sbi/sbi_domain_data.h . . . 373
include/sbi/sbi_ecall.h . . . 377
include/sbi/sbi_ecall_interface.h . . . 385
include/sbi/sbi_emulate_csr.h . . . 412
include/sbi/sbi_error.h . . . 415
include/sbi/sbi_fifo.h . . . 418
include/sbi/sbi_fwft.h . . . 425
include/sbi/sbi_hart.h . . . 427
include/sbi/sbi_hartmask.h . . . 441
include/sbi/sbi_heap.h . . . 452
include/sbi/sbi_hfence.h . . . 466
include/sbi/sbi_hsm.h . . . 468
include/sbi/sbi_illegal_insn.h . . . 479
include/sbi/sbi_init.h . . . 481
include/sbi/sbi_ipi.h . . . 485
include/sbi/sbi_irqchip.h . . . 493
include/sbi/sbi_list.h . . . 497
include/sbi/sbi_math.h . . . 509
include/sbi/sbi_mpxy.h . . . 511
include/sbi/sbi_platform.h . . . 520

include/sbi/sbi_pmu.h . . . 543
include/sbi/sbi_scratch.h . . . 555
include/sbi/sbi_sse.h . . . 565
include/sbi/sbi_string.h . . . 576
include/sbi/sbi_system.h . . . 582
include/sbi/sbi_timer.h . . . 589
include/sbi/sbi_tlb.h . . . 596
include/sbi/sbi_trap.h . . . 601
include/sbi/sbi_trap_ldst.h . . . 612
include/sbi/sbi_types.h . . . 618
include/sbi/sbi_unit_test.h . . . 624
include/sbi/sbi_unpriv.h . . . 625
include/sbi/sbi_version.h . . . 627
include/sbi_utils/cppc/fdt_cppc.h . . . 628
include/sbi_utils/fdt/fdt_domain.h . . . 630
include/sbi_utils/fdt/fdt_driver.h . . . 632
include/sbi_utils/fdt/fdt_fixup.h . . . 636
include/sbi_utils/fdt/fdt_helper.h . . . 643
include/sbi_utils/fdt/fdt_pmu.h . . . 663
include/sbi_utils/gpio/fdt_gpio.h . . . 665
include/sbi_utils/gpio/gpio.h . . . 667
include/sbi_utils/hsm/fdt_hsm.h . . . 672
include/sbi_utils/i2c/dw_i2c.h . . . 674
include/sbi_utils/i2c/fdt_i2c.h . . . 676
include/sbi_utils/i2c/i2c.h . . . 678
include/sbi_utils/ipi/aclint_mswi.h . . . 683
include/sbi_utils/ipi/andes_plicsw.h . . . 685
include/sbi_utils/ipi/fdt_ipi.h . . . 687
include/sbi_utils/irqchip/aplic.h . . . 689
include/sbi_utils/irqchip/fdt_irqchip.h . . . 692
include/sbi_utils/irqchip/imsic.h . . . 693
include/sbi_utils/irqchip/plic.h . . . 695
include/sbi_utils/mailbox/fdt_mailbox.h . . . 699
include/sbi_utils/mailbox/mailbox.h . . . 701
include/sbi_utils/mailbox/rpmi_mailbox.h . . . 710
include/sbi_utils/mailbox/rpmi_msgprot.h . . . 714
include/sbi_utils/mpxy/fdt_mpxy.h . . . 732
include/sbi_utils/regmap/fdt_regmap.h . . . 733
include/sbi_utils/regmap/regmap.h . . . 736
include/sbi_utils/reset/fdt_reset.h . . . 741
include/sbi_utils/serial/cadence-uart.h . . . 743
include/sbi_utils/serial/fdt_serial.h . . . 744
include/sbi_utils/serial/gaisler-uart.h . . . 746
include/sbi_utils/serial/litex-uart.h . . . 748
include/sbi_utils/serial/renesas-scif.h . . . 750
include/sbi_utils/serial/semihosting.h . . . 751
include/sbi_utils/serial/shakti-uart.h . . . 753
include/sbi_utils/serial/sifive-uart.h . . . 754
include/sbi_utils/serial/uart8250.h . . . 756
include/sbi_utils/serial/xlnx_uartlite.h . . . 758
include/sbi_utils/suspend/fdt_suspend.h . . . 759
include/sbi_utils/sys/atcsmu.h . . . 761
include/sbi_utils/sys/htif.h . . . 766
include/sbi_utils/timer/aclint_mtimer.h . . . 768
include/sbi_utils/timer/andes_plmt.h . . . 772
include/sbi_utils/timer/fdt_timer.h . . . 774
lib/sbi/riscv_asm.c . . . 775
lib/sbi/riscv_atomic.c . . . 785

lib/sbi/riscv_locks.c . . . 791
lib/sbi/sbi_bitmap.c . . . 795
lib/sbi/sbi_bitops.c . . . 797
lib/sbi/sbi_console.c . . . 799
lib/sbi/sbi_cppc.c . . . 821
lib/sbi/sbi_dbtr.c . . . 825
lib/sbi/sbi_domain.c . . . 838
lib/sbi/sbi_domain_context.c . . . 856
lib/sbi/sbi_domain_data.c . . . 862
lib/sbi/sbi_double_trap.c . . . 866
lib/sbi/sbi_ecall.c . . . 868
lib/sbi/sbi_ecall_base.c . . . 872
lib/sbi/sbi_ecall_cppc.c . . . 874
lib/sbi/sbi_ecall_dbcn.c . . . 876
lib/sbi/sbi_ecall_dbtr.c . . . 877
lib/sbi/sbi_ecall_fwft.c . . . 879
lib/sbi/sbi_ecall_hsm.c . . . 880
lib/sbi/sbi_ecall_ipi.c . . . 883
lib/sbi/sbi_ecall_legacy.c . . . 884
lib/sbi/sbi_ecall_mpxy.c . . . 886
lib/sbi/sbi_ecall_pmu.c . . . 888
lib/sbi/sbi_ecall_rfence.c . . . 890
lib/sbi/sbi_ecall_srst.c . . . 891
lib/sbi/sbi_ecall_sse.c . . . 894
lib/sbi/sbi_ecall_susp.c . . . 896
lib/sbi/sbi_ecall_time.c . . . 898
lib/sbi/sbi_ecall_vendor.c . . . 899
lib/sbi/sbi_emulate_csr.c . . . 902
lib/sbi/sbi_fifo.c . . . 905
lib/sbi/sbi_fwft.c . . . 914
lib/sbi/sbi_hart.c . . . 924
lib/sbi/sbi_heap.c . . . 945
lib/sbi/sbi_hsm.c . . . 953
lib/sbi/sbi_illegal_insn.c . . . 967
lib/sbi/sbi_init.c . . . 969
lib/sbi/sbi_ipi.c . . . 981
lib/sbi/sbi_irqchip.c . . . 991
lib/sbi/sbi_math.c . . . 994
lib/sbi/sbi_mpxy.c . . . 994
lib/sbi/sbi_platform.c . . . 1007
lib/sbi/sbi_pmu.c . . . 1009
lib/sbi/sbi_scratch.c . . . 1028
lib/sbi/sbi_sse.c . . . 1032
lib/sbi/sbi_string.c . . . 1064
lib/sbi/sbi_system.c . . . 1068
lib/sbi/sbi_timer.c . . . 1075
lib/sbi/sbi_tlb.c . . . 1082
lib/sbi/sbi_trap.c . . . 1094
lib/sbi/sbi_trap_ldst.c . . . 1099
lib/sbi/sbi_trap_v_ldst.c . . . 1106
lib/sbi/sbi_unpriv.c . . . 1108
lib/sbi/tests/riscv_atomic_test.c . . . 1110
lib/sbi/tests/riscv_locks_test.c . . . 1115
lib/sbi/tests/sbi_bitmap_test.c . . . 1118
lib/sbi/tests/sbi_console_test.c . . . 1120
lib/sbi/tests/sbi_math_test.c . . . 1125
lib/sbi/tests/sbi_unit_test.c . . . 1126
lib/utils/cppc/fdt_cppc.c . . . 1128

lib/utils/cppc/fdt_cppc_rpmi.c . . . 1129
lib/utils/fdt/fdt_domain.c . . . 1133
lib/utils/fdt/fdt_driver.c . . . 1140
lib/utils/fdt/fdt_fixup.c . . . 1145
lib/utils/fdt/fdt_helper.c . . . 1154
lib/utils/fdt/fdt_pmu.c . . . 1175
lib/utils/gpio/fdt_gpio.c . . . 1177
lib/utils/gpio/fdt_gpio_designware.c . . . 1180
lib/utils/gpio/fdt_gpio_sifive.c . . . 1182
lib/utils/gpio/fdt_gpio_starfive.c . . . 1185
lib/utils/gpio/gpio.c . . . 1187
lib/utils/hsm/fdt_hsm.c . . . 1191
lib/utils/hsm/fdt_hsm_rpmi.c . . . 1192
lib/utils/i2c/dw_i2c.c . . . 1198
lib/utils/i2c/fdt_i2c.c . . . 1206
lib/utils/i2c/fdt_i2c_dw.c . . . 1208
lib/utils/i2c/fdt_i2c_sifive.c . . . 1210
lib/utils/i2c/i2c.c . . . 1217
lib/utils/ipi/aclint_mswi.c . . . 1220
lib/utils/ipi/andes_plicsw.c . . . 1222
lib/utils/ipi/fdt_ipi.c . . . 1224
lib/utils/ipi/fdt_ipi_mswi.c . . . 1225
lib/utils/ipi/fdt_ipi_plicsw.c . . . 1227
lib/utils/irqchip/aplic.c . . . 1228
lib/utils/irqchip/fdt_irqchip.c . . . 1236
lib/utils/irqchip/fdt_irqchip_aplic.c . . . 1237
lib/utils/irqchip/fdt_irqchip_imsic.c . . . 1238
lib/utils/irqchip/fdt_irqchip_plic.c . . . 1240
lib/utils/irqchip/imsic.c . . . 1242
lib/utils/irqchip/plic.c . . . 1249
lib/utils/libfdt/fdt.c . . . 1256
lib/utils/libfdt/fdt.h . . . 1267
lib/utils/libfdt/fdt_addresses.c . . . 1269
lib/utils/libfdt/fdt_check.c . . . 1273
lib/utils/libfdt/fdt_empty_tree.c . . . 1275
lib/utils/libfdt/fdt_overlay.c . . . 1276
lib/utils/libfdt/fdt_ro.c . . . 1286
lib/utils/libfdt/fdt_rw.c . . . 1322
lib/utils/libfdt/fdt_strerror.c . . . 1337
lib/utils/libfdt/fdt_sw.c . . . 1338
lib/utils/libfdt/fdt_wip.c . . . 1348
lib/utils/libfdt/libfdt.h . . . 1352
lib/utils/libfdt/libfdt_env.h . . . 1428
lib/utils/libfdt/libfdt_internal.h . . . 1434
lib/utils/libquad/divdi3.c . . . 1445
lib/utils/libquad/moddi3.c . . . 1449
lib/utils/libquad/qdivrem.c . . . 1450
lib/utils/libquad/quad.h . . . 1453
lib/utils/libquad/udivdi3.c . . . 1458
lib/utils/libquad/umoddi3.c . . . 1459
lib/utils/libquad/include/limits.h . . . 1447
lib/utils/libquad/include/sys/cdefs.h . . . 1447
lib/utils/libquad/include/sys/types.h . . . 1448
lib/utils/mailbox/fdt_mailbox.c . . . 1461
lib/utils/mailbox/fdt_mailbox_rpmi_shmem.c . . . 1463
lib/utils/mailbox/mailbox.c . . . 1474
lib/utils/mailbox/rpmi_mailbox.c . . . 1479
lib/utils/mpxy/fdt_mpxy.c . . . 1482

lib/utils/mpxy/fdt_mpxy_rpmi_mbox.c . . . 1484
lib/utils/regmap/fdt_regmap.c . . . 1491
lib/utils/regmap/fdt_regmap_syscon.c . . . 1493
lib/utils/regmap/regmap.c . . . 1500
lib/utils/reset/fdt_reset.c . . . 1505
lib/utils/reset/fdt_reset_atcwdt200.c . . . 1506
lib/utils/reset/fdt_reset_gpio.c . . . 1510
lib/utils/reset/fdt_reset_htif.c . . . 1515
lib/utils/reset/fdt_reset_rpmi.c . . . 1516
lib/utils/reset/fdt_reset_sg2042_hwmon_mcu.c . . . 1519
lib/utils/reset/fdt_reset_sunxi_wdt.c . . . 1522
lib/utils/reset/fdt_reset_syscon.c . . . 1525
lib/utils/serial/cadence-uart.c . . . 1529
lib/utils/serial/fdt_serial.c . . . 1534
lib/utils/serial/fdt_serial_cadence.c . . . 1535
lib/utils/serial/fdt_serial_gaisler.c . . . 1537
lib/utils/serial/fdt_serial_htif.c . . . 1538
lib/utils/serial/fdt_serial_litex.c . . . 1540
lib/utils/serial/fdt_serial_renesas_scif.c . . . 1541
lib/utils/serial/fdt_serial_shakti.c . . . 1543
lib/utils/serial/fdt_serial_sifive.c . . . 1544
lib/utils/serial/fdt_serial_uart8250.c . . . 1546
lib/utils/serial/fdt_serial_xlnx_uartlite.c . . . 1547
lib/utils/serial/gaisler-uart.c . . . 1549
lib/utils/serial/litex-uart.c . . . 1553
lib/utils/serial/renesas_scif.c . . . 1556
lib/utils/serial/semihosting.c . . . 1561
lib/utils/serial/shakti-uart.c . . . 1567
lib/utils/serial/sifive-uart.c . . . 1569
lib/utils/serial/uart8250.c . . . 1574
lib/utils/serial/xlnx-uartlite.c . . . 1579
lib/utils/suspend/fdt_suspend.c . . . 1582
lib/utils/suspend/fdt_suspend_rpmi.c . . . 1584
lib/utils/sys/atcsmu.c . . . 1587
lib/utils/sys/htif.c . . . 1588
lib/utils/timer/aclint_mtimer.c . . . 1597
lib/utils/timer/andes_plmt.c . . . 1601
lib/utils/timer/fdt_timer.c . . . 1603
lib/utils/timer/fdt_timer_mtimer.c . . . 1605
lib/utils/timer/fdt_timer_plmt.c . . . 1607

Chapter 29

Data Structure Documentation

29.1 __aligned Struct Reference

```
#include <riscv_locks.h>
```

Data Fields

- u16 next
- u16 owner

29.1.1 Field Documentation

29.1.1.1 next

```
u16 __aligned::next
```

29.1.1.2 owner

```
u16 __aligned::owner
```

The documentation for this struct was generated from the following file:

- include/sbi/riscv_locks.h

29.2 aclint_mswi_data Struct Reference

```
#include <aclint_mswi.h>
```

Data Fields

- unsigned long addr
- unsigned long size
- u32 first_hartid
- u32 hart_count

29.2.1 Field Documentation

29.2.1.1 addr

```
unsigned long aclint_mswi_data::addr
```

29.2.1.2 first_hartid

```
u32 aclint_mswi_data::first_hartid
```

29.2.1.3 hart_count

```
u32 aclint_mswi_data::hart_count
```

29.2.1.4 size

```
unsigned long aclint_mswi_data::size
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/ipi/aclint_mswi.h

29.3 aclint_mtimer_data Struct Reference

```
#include <aclint_mtimer.h>
```

Data Fields

- unsigned long mtime_freq
- unsigned long mtime_addr
- unsigned long mtime_size
- unsigned long mtimecmp_addr
- unsigned long mtimecmp_size
- u32 first_hartid
- u32 hart_count
- bool has_64bit_mmio
- bool has_shared_mtime
- struct aclint_mtimer_data ∗ time_delta_reference
- unsigned long time_delta_computed
- u64(∗ time_rd)(volatile u64 ∗addr)
- void(∗ time_wr)(bool timecmp, u64 value, volatile u64 ∗addr)

29.3.1 Field Documentation

29.3.1.1 first_hartid

```
u32 aclint_mtimer_data::first_hartid
```

29.3.1.2 hart_count

```
u32 aclint_mtimer_data::hart_count
```

29.3.1.3 has_64bit_mmio

```
bool aclint_mtimer_data::has_64bit_mmio
```

29.3.1.4 has_shared_mtime

```
bool aclint_mtimer_data::has_shared_mtime
```

29.3.1.5 mtime_addr

```
unsigned long aclint_mtimer_data::mtime_addr
```

29.3.1.6 mtime_freq

```
unsigned long aclint_mtimer_data::mtime_freq
```

29.3.1.7 mtime_size

```
unsigned long aclint_mtimer_data::mtime_size
```

29.3.1.8 mtimecmp_addr

```
unsigned long aclint_mtimer_data::mtimecmp_addr
```

29.3.1.9 mtimecmp_size

```
unsigned long aclint_mtimer_data::mtimecmp_size
```

29.3.1.10 time_delta_computed

```
unsigned long aclint_mtimer_data::time_delta_computed
```

29.3.1.11 time_delta_reference

```
struct aclint_mtimer_data* aclint_mtimer_data::time_delta_reference
```

29.3.1.12 time_rd

```
u64(* aclint_mtimer_data::time_rd) (volatile u64 *addr)
```

29.3.1.13 time_wr

```
void(* aclint_mtimer_data::time_wr) (bool timecmp, u64 value, volatile u64 *addr)
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/timer/aclint_mtimer.h

29.4 aplic_data Struct Reference

```
#include <aplic.h>
```

Data Fields

- struct sbi_irqchip_device irqchip
- unsigned long addr
- unsigned long size
- unsigned long num_idc
- unsigned long num_source
- bool targets_mmode
- bool has_msicfg_mmode
- struct aplic_msicfg_data msicfg_mmode
- bool has_msicfg_smode
- struct aplic_msicfg_data msicfg_smode
- struct aplic_delegate_data delegate [APLIC_MAX_DELEGATE]

29.4.1 Field Documentation

29.4.1.1 addr

```
unsigned long aplic_data::addr
```

29.4.1.2 delegate

```
struct aplic_delegate_data aplic_data::delegate[APLIC_MAX_DELEGATE]
```

29.4.1.3 has_msicfg_mmode

```
bool aplic_data::has_msicfg_mmode
```

29.4.1.4 has_msicfg_smode

```
bool aplic_data::has_msicfg_smode
```

29.4.1.5 irqchip

```
struct sbi_irqchip_device aplic_data::irqchip
```

29.4.1.6 msicfg_mmode

```
struct aplic_msicfg_data aplic_data::msicfg_mmode
```

29.4.1.7 msicfg_smode

```
struct aplic_msicfg_data aplic_data::msicfg_smode
```

29.4.1.8 num_idc

```
unsigned long aplic_data::num_idc
```

29.4.1.9 num_source

```
unsigned long aplic_data::num_source
```

29.4.1.10 size

```
unsigned long aplic_data::size
```

29.4.1.11 targets_mmode

```
bool aplic_data::targets_mmode
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/aplic.h

29.5 aplic_delegate_data Struct Reference

```
#include <aplic.h>
```

Data Fields

- u32 first_irq
- u32 last_irq
- u32 child_index

29.5.1 Field Documentation

29.5.1.1 child_index

```
u32 aplic_delegate_data::child_index
```

29.5.1.2 first_irq

```
u32 aplic_delegate_data::first_irq
```

29.5.1.3 last_irq

```
u32 aplic_delegate_data::last_irq
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/aplic.h

29.6 aplic_msicfg_data Struct Reference

```
#include <aplic.h>
```

Data Fields

- unsigned long lhxs
- unsigned long lhxw
- unsigned long hhxs
- unsigned long hhxw
- unsigned long base_addr

29.6.1 Field Documentation

29.6.1.1 base_addr

```
unsigned long aplic_msicfg_data::base_addr
```

29.6.1.2 hhxs

```
unsigned long aplic_msicfg_data::hhxs
```

29.6.1.3 hhxw

```
unsigned long aplic_msicfg_data::hhxw
```

29.6.1.4 lhxs

```
unsigned long aplic_msicfg_data::lhxs
```

29.6.1.5 lhxw

```
unsigned long aplic_msicfg_data::lhxw
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/aplic.h

29.7 atomic_t Struct Reference

```
#include <riscv_atomic.h>
```

Data Fields

- volatile long counter

29.7.1 Field Documentation

29.7.1.1 counter

```
volatile long atomic_t::counter
```

The documentation for this struct was generated from the following file:

- include/sbi/riscv_atomic.h

29.8 dw_i2c_adapter Struct Reference

```
#include <dw_i2c.h>
```

Data Fields

- unsigned long addr
- struct i2c_adapter adapter

29.8.1 Field Documentation

29.8.1.1 adapter

```
struct i2c_adapter dw_i2c_adapter::adapter
```

29.8.1.2 addr

```
unsigned long dw_i2c_adapter::addr
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/i2c/dw_i2c.h

29.9 fdt_driver Struct Reference

```
#include <fdt_driver.h>
```

Data Fields

- const struct fdt_match ∗ match_table
- int(∗ init)(const void ∗fdt, int nodeoff, const struct fdt_match ∗match)
- bool experimental

29.9.1 Field Documentation

29.9.1.1 experimental

```
bool fdt_driver::experimental
```

29.9.1.2 init

```
int(* fdt_driver::init) (const void *fdt, int nodeoff, const struct fdt_match *match)
```

29.9.1.3 match_table

```
const struct fdt_match* fdt_driver::match_table
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_driver.h

29.10 fdt_general_fixup Struct Reference

```
#include <fdt_fixup.h>
```

Data Fields

- struct sbi_dlist head
- const char ∗ name
- void(∗ do_fixup)(struct fdt_general_fixup ∗f, void ∗fdt)

29.10.1 Detailed Description

Representation of a general fixup

29.10.2 Field Documentation

29.10.2.1 do_fixup

```
void(* fdt_general_fixup::do_fixup) (struct fdt_general_fixup *f, void *fdt)
```

29.10.2.2 head

```
struct sbi_dlist fdt_general_fixup::head
```

29.10.2.3 name

```
const char* fdt_general_fixup::name
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_fixup.h

29.11 fdt_gpio Struct Reference

```
#include <fdt_gpio.h>
```

Data Fields

- struct fdt_driver driver
- int(* xlate)(struct gpio_chip *chip, const struct fdt_phandle_args *pargs, struct gpio_pin *out_pin)

29.11.1 Detailed Description

FDT based GPIO driver

29.11.2 Field Documentation

29.11.2.1 driver

```
struct fdt_driver fdt_gpio::driver
```

29.11.2.2 xlate

```
int(* fdt_gpio::xlate) (struct gpio_chip *chip, const struct fdt_phandle_args *pargs, struct
gpio_pin *out_pin)
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/gpio/fdt_gpio.h

29.12 fdt_header Struct Reference

```
#include <fdt.h>
```

Data Fields

- fdt32_t magic
- fdt32_t totalsize
- fdt32_t off_dt_struct
- fdt32_t off_dt_strings
- fdt32_t off_mem_rsvmap
- fdt32_t version
- fdt32_t last_comp_version
- fdt32_t boot_cpuid_phys
- fdt32_t size_dt_strings
- fdt32_t size_dt_struct

29.12.1 Field Documentation

29.12.1.1 boot_cpuid_phys

`fdt32_t fdt_header::boot_cpuid_phys`

29.12.1.2 last_comp_version

`fdt32_t fdt_header::last_comp_version`

29.12.1.3 magic

`fdt32_t fdt_header::magic`

29.12.1.4 off_dt_strings

`fdt32_t fdt_header::off_dt_strings`

29.12.1.5 off_dt_struct

`fdt32_t fdt_header::off_dt_struct`

29.12.1.6 off_mem_rsvmap

`fdt32_t fdt_header::off_mem_rsvmap`

29.12.1.7 size_dt_strings

`fdt32_t fdt_header::size_dt_strings`

29.12.1.8 size_dt_struct

`fdt32_t fdt_header::size_dt_struct`

29.12.1.9 totalsize

`fdt32_t fdt_header::totalsize`

29.12.1.10 version

`fdt32_t fdt_header::version`

The documentation for this struct was generated from the following file:

- lib/utils/libfdt/fdt.h

29.13 fdt_mailbox Struct Reference

```
#include <fdt_mailbox.h>
```

Data Fields

- struct fdt_driver driver
- int(∗ xlate)(struct mbox_controller ∗mbox, const struct fdt_phandle_args ∗pargs, u32 ∗out_chan_args)

29.13.1 Detailed Description

FDT based mailbox driver

29.13.2 Field Documentation

29.13.2.1 driver

```
struct fdt_driver fdt_mailbox::driver
```

29.13.2.2 xlate

```
int(* fdt_mailbox::xlate) (struct mbox_controller *mbox, const struct fdt_phandle_args *pargs,
u32 *out_chan_args)
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/fdt_mailbox.h

29.14 fdt_match Struct Reference

```
#include <fdt_helper.h>
```

Data Fields

- const char * compatible
- const void * data

29.14.1 Field Documentation

29.14.1.1 compatible

```
const char* fdt_match::compatible
```

29.14.1.2 data

```
const void* fdt_match::data
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_helper.h

29.15 fdt_node_header Struct Reference

```
#include <fdt.h>
```

Data Fields

- fdt32_t tag
- char name [0]

29.15.1 Field Documentation

29.15.1.1 name

```
char fdt_node_header::name[0]
```

29.15.1.2 tag

```
fdt32_t fdt_node_header::tag
```

The documentation for this struct was generated from the following file:

- lib/utils/libfdt/fdt.h

29.16 fdt_phandle_args Struct Reference

```
#include <fdt_helper.h>
```

Data Fields

- int node_offset
- int args_count
- u32 args [FDT_MAX_PHANDLE_ARGS]

29.16.1 Field Documentation

29.16.1.1 args

```
u32 fdt_phandle_args::args[FDT_MAX_PHANDLE_ARGS]
```

29.16.1.2 args_count

```
int fdt_phandle_args::args_count
```

29.16.1.3 node_offset

```
int fdt_phandle_args::node_offset
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_helper.h

29.17 fdt_pmu_hw_event_counter_map Struct Reference

```
#include <fdt_pmu.h>
```

Data Fields

- uint32_t eidx_start
- uint32_t eidx_end
- uint32_t ctr_map

29.17.1 Field Documentation

29.17.1.1 ctr_map

```
uint32_t fdt_pmu_hw_event_counter_map::ctr_map
```

29.17.1.2 eidx_end

```
uint32_t fdt_pmu_hw_event_counter_map::eidx_end
```

29.17.1.3 eidx_start

```
uint32_t fdt_pmu_hw_event_counter_map::eidx_start
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_pmu.h

29.18 fdt_pmu_hw_event_select_map Struct Reference

```
#include <fdt_pmu.h>
```

Data Fields

- uint32_t eidx
- uint64_t select

29.18.1 Field Documentation

29.18.1.1 eidx

```
uint32_t fdt_pmu_hw_event_select_map::eidx
```

29.18.1.2 select

```
uint64_t fdt_pmu_hw_event_select_map::select
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_pmu.h

29.19 fdt_pmu_raw_event_counter_map Struct Reference

```
#include <fdt_pmu.h>
```

Data Fields

- uint64_t select
- uint64_t select_mask
- uint32_t ctr_map

29.19.1 Field Documentation

29.19.1.1 ctr_map

```
uint32_t fdt_pmu_raw_event_counter_map::ctr_map
```

29.19.1.2 select

```
uint64_t fdt_pmu_raw_event_counter_map::select
```

29.19.1.3 select_mask

```
uint64_t fdt_pmu_raw_event_counter_map::select_mask
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_pmu.h

29.20 fdt_property Struct Reference

```
#include <fdt.h>
```

Data Fields

- fdt32_t tag
- fdt32_t len
- fdt32_t nameoff
- char data [0]

29.20.1 Field Documentation

29.20.1.1 data

```
char fdt_property::data[0]
```

29.20.1.2 len

`fdt32_t fdt_property::len`

29.20.1.3 nameoff

`fdt32_t fdt_property::nameoff`

29.20.1.4 tag

`fdt32_t fdt_property::tag`

The documentation for this struct was generated from the following file:

- lib/utils/libfdt/fdt.h

29.21 fdt_reserve_entry Struct Reference

`#include <fdt.h>`

Data Fields

- fdt64_t address
- fdt64_t size

29.21.1 Field Documentation

29.21.1.1 address

`fdt64_t fdt_reserve_entry::address`

29.21.1.2 size

`fdt64_t fdt_reserve_entry::size`

The documentation for this struct was generated from the following file:

- lib/utils/libfdt/fdt.h

29.22 fw_dynamic_info Struct Reference

`#include <fw_dynamic.h>`

Data Fields

- unsigned long magic
- unsigned long version
- unsigned long next_addr
- unsigned long next_mode
- unsigned long options
- unsigned long boot_hart

29.22.1 Detailed Description

Representation dynamic info passed by previous booting stage

29.22.2 Field Documentation

29.22.2.1 boot_hart

`unsigned long fw_dynamic_info::boot_hart`

Preferred boot HART id

It is possible that the previous booting stage uses same link address as the FW_DYNAMIC firmware. In this case, the relocation lottery mechanism can potentially overwrite the previous booting stage while other HARTs are still running in the previous booting stage leading to boot-time crash. To avoid this boot-time crash, the previous booting stage can specify last HART that will jump to the FW_DYNAMIC firmware as the preferred boot HART.

To avoid specifying a preferred boot HART, the previous booting stage can set it to -1UL which will force the FW_↩ DYNAMIC firmware to use the relocation lottery mechanism.

29.22.2.2 magic

`unsigned long fw_dynamic_info::magic`

Info magic

29.22.2.3 next_addr

`unsigned long fw_dynamic_info::next_addr`

Next booting stage address

29.22.2.4 next_mode

`unsigned long fw_dynamic_info::next_mode`

Next booting stage mode

29.22.2.5 options

`unsigned long fw_dynamic_info::options`

Options for OpenSBI library

29.22.2.6 version

`unsigned long fw_dynamic_info::version`

Info version

The documentation for this struct was generated from the following file:

- include/sbi/fw_dynamic.h

29.23 gpio_chip Struct Reference

`#include <gpio.h>`

Data Fields

- const void ∗ driver
- unsigned int id
- unsigned int ngpio
- int(∗ get_direction)(struct gpio_pin ∗gp)
- int(∗ direction_input)(struct gpio_pin ∗gp)
- int(∗ direction_output)(struct gpio_pin ∗gp, int value)
- int(∗ get)(struct gpio_pin ∗gp)
- void(∗ set)(struct gpio_pin ∗gp, int value)
- struct sbi_dlist node

29.23.1 Detailed Description

Representation of a GPIO chip

29.23.2 Field Documentation

29.23.2.1 direction_input

`int(* gpio_chip::direction_input) (struct gpio_pin *gp)`
Set input direction of GPIO pin

Returns

0 on success and negative error code on failure

29.23.2.2 direction_output

`int(* gpio_chip::direction_output) (struct gpio_pin *gp, int value)`
Set output direction of GPIO pin with given output value

Returns

0 on success and negative error code on failure

29.23.2.3 driver

`const void* gpio_chip::driver`
Pointer to GPIO driver owning this GPIO chip

29.23.2.4 get

`int(* gpio_chip::get) (struct gpio_pin *gp)`
Get current value of GPIO pin

Returns

0=low, 1=high, or negative error

29.23.2.5 get_direction

`int(* gpio_chip::get_direction) (struct gpio_pin *gp)`
Get current direction of GPIO pin

Returns

0=output, 1=input, or negative error

29.23.2.6 id

`unsigned int gpio_chip::id`
Uniquie ID of the GPIO chip assigned by the driver

29.23.2.7 ngpio

`unsigned int gpio_chip::ngpio`
Number of GPIOs supported by the GPIO chip

29.23.2.8 node

`struct sbi_dlist gpio_chip::node`
List

29.23.2.9 set

`void(* gpio_chip::set) (struct gpio_pin *gp, int value)`
Set output value for GPIO pin
The documentation for this struct was generated from the following file:

- include/sbi_utils/gpio/gpio.h

29.24 gpio_pin Struct Reference

`#include <gpio.h>`

Data Fields

- struct gpio_chip * chip
- unsigned int offset
- unsigned int flags

29.24.1 Detailed Description

Representation of a GPIO pin

29.24.2 Field Documentation

29.24.2.1 chip

`struct gpio_chip* gpio_pin::chip`
Pointer to the GPIO chip

29.24.2.2 flags

`unsigned int gpio_pin::flags`
Additional configuration flags of the GPIO pin desired by GPIO clients.
NOTE: GPIO chip can have custom configuration flags.

29.24.2.3 offset

`unsigned int gpio_pin::offset`
Identification of GPIO pin within GPIO chip
The documentation for this struct was generated from the following file:

- include/sbi_utils/gpio/gpio.h

29.25 i2c_adapter Struct Reference

`#include <i2c.h>`

Data Fields

- int id
- int(* write)(struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- int(* read)(struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- struct sbi_dlist node

29.25.1 Detailed Description

Representation of a I2C adapter

29.25.2 Field Documentation

29.25.2.1 id

`int i2c_adapter::id`
Unique ID of the I2C adapter assigned by the driver

29.25.2.2 node

`struct sbi_dlist i2c_adapter::node`
List

29.25.2.3 read

`int(* i2c_adapter::read) (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)`
Read buffer from given address, register

Returns

0 on success and negative error code on failure

29.25.2.4 write

`int(* i2c_adapter::write) (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)`
Send buffer to given address, register

Returns

0 on success and negative error code on failure

The documentation for this struct was generated from the following file:

- include/sbi_utils/i2c/i2c.h

29.26 imsic_data Struct Reference

`#include <imsic.h>`

Data Fields

- bool targets_mmode
- u32 guest_index_bits
- u32 hart_index_bits
- u32 group_index_bits
- u32 group_index_shift
- unsigned long num_ids
- struct imsic_regs regs [IMSIC_MAX_REGS]

29.26.1 Field Documentation

29.26.1.1 group_index_bits

`u32 imsic_data::group_index_bits`

29.26.1.2 group_index_shift

`u32 imsic_data::group_index_shift`

29.26.1.3 guest_index_bits

```
u32 imsic_data::guest_index_bits
```

29.26.1.4 hart_index_bits

```
u32 imsic_data::hart_index_bits
```

29.26.1.5 num_ids

```
unsigned long imsic_data::num_ids
```

29.26.1.6 regs

```
struct imsic_regs imsic_data::regs[IMSIC_MAX_REGS]
```

29.26.1.7 targets_mmode

```
bool imsic_data::targets_mmode
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/imsic.h

29.27 imsic_regs Struct Reference

```
#include <imsic.h>
```

Data Fields

- unsigned long addr
- unsigned long size

29.27.1 Field Documentation

29.27.1.1 addr

```
unsigned long imsic_regs::addr
```

29.27.1.2 size

```
unsigned long imsic_regs::size
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/imsic.h

29.28 mbox_chan Struct Reference

```
#include <mailbox.h>
```

Data Fields

- struct sbi_dlist node
- struct mbox_controller * mbox
- u32 chan_args [MBOX_CHAN_MAX_ARGS]

29.28.1 Detailed Description

Representation of a mailbox channel

29.28.2 Field Documentation

29.28.2.1 chan_args

```
u32 mbox_chan::chan_args[MBOX_CHAN_MAX_ARGS]
```

29.28.2.2 mbox

```
struct mbox_controller* mbox_chan::mbox
```

Pointer to the mailbox controller

29.28.2.3 node

```
struct sbi_dlist mbox_chan::node
```

List head

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/mailbox.h

29.29 mbox_controller Struct Reference

```
#include <mailbox.h>
```

Data Fields

- struct sbi_dlist node
- atomic_t xfer_next_seq
- struct sbi_dlist chan_list
- unsigned int id
- unsigned int max_xfer_len
- void * driver
- struct mbox_chan *(* request_chan)(struct mbox_controller *mbox, u32 *chan_args)
- void(* free_chan)(struct mbox_controller *mbox, struct mbox_chan *chan)
- int(* xfer)(struct mbox_chan *chan, struct mbox_xfer *xfer)
- int(* get_attribute)(struct mbox_chan *chan, int attr_id, void *out_value)
- int(* set_attribute)(struct mbox_chan *chan, int attr_id, void *new_value)

29.29.1 Detailed Description

Representation of a mailbox controller

29.29.2 Field Documentation

29.29.2.1 chan_list

```
struct sbi_dlist mbox_controller::chan_list
```

29.29.2.2 driver

```
void* mbox_controller::driver
```

Pointer to mailbox driver owning this mailbox controller

29.29.2.3 free_chan

```
void(* mbox_controller::free_chan) (struct mbox_controller *mbox, struct mbox_chan *chan)
```

Free a mailbox channel from the mailbox controller

29.29.2.4 get_attribute

`int(* mbox_controller::get_attribute) (struct mbox_chan *chan, int attr_id, void *out_value)`
Get an attribute of mailbox channel

29.29.2.5 id

`unsigned int mbox_controller::id`
Unique ID of the mailbox controller assigned by the driver

29.29.2.6 max_xfer_len

`unsigned int mbox_controller::max_xfer_len`
Maximum length of transfer supported by the mailbox controller

29.29.2.7 node

`struct sbi_dlist mbox_controller::node`
List head

29.29.2.8 request_chan

`struct mbox_chan *(* mbox_controller::request_chan) (struct mbox_controller *mbox, u32 *chan_↩ args)`
Request a mailbox channel from the mailbox controller

29.29.2.9 set_attribute

`int(* mbox_controller::set_attribute) (struct mbox_chan *chan, int attr_id, void *new_value)`
Set an attribute of mailbox channel

29.29.2.10 xfer

`int(* mbox_controller::xfer) (struct mbox_chan *chan, struct mbox_xfer *xfer)`
Transfer data over mailbox channel

29.29.2.11 xfer_next_seq

`atomic_t mbox_controller::xfer_next_seq`
Next sequence atomic counter
The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/mailbox.h

29.30 mbox_xfer Struct Reference

`#include <mailbox.h>`

Data Fields

- unsigned long flags
- void * args
- long seq
- void * tx
- unsigned long tx_len
- unsigned long tx_timeout
- void * rx
- unsigned long rx_len
- unsigned long rx_timeout

29.30.1 Detailed Description

Representation of a mailbox data transfer
NOTE: If both "tx" and "rx" are non-NULL then Tx is done before Rx.

29.30.2 Field Documentation

29.30.2.1 args

`void* mbox_xfer::args`
Transfer arguments (or parameters)

29.30.2.2 flags

`unsigned long mbox_xfer::flags`
Transfer flags

29.30.2.3 rx

`void* mbox_xfer::rx`
Receive data pointer

29.30.2.4 rx_len

`unsigned long mbox_xfer::rx_len`
Receive data length (valid only if rx != NULL)

29.30.2.5 rx_timeout

`unsigned long mbox_xfer::rx_timeout`
Receive timeout milliseconds (valid only if rx != NULL)
If this field is non-zero along with rx != NULL then the mailbox controller driver will wait specified milliseconds for receive data transfer to complete else the mailbox controller driver will not wait.

29.30.2.6 seq

`long mbox_xfer::seq`
Sequence number
If MBOX_XFER_SEQ is not set in flags then mbox_chan_xfer() will generate a unique sequence number and update this field else mbox_chan_xfer() will blindly use the sequence number specified by this field.

29.30.2.7 tx

`void* mbox_xfer::tx`
Send data pointer

29.30.2.8 tx_len

`unsigned long mbox_xfer::tx_len`
Send data length (valid only if tx != NULL)

29.30.2.9 tx_timeout

`unsigned long mbox_xfer::tx_timeout`
Send timeout milliseconds (valid only if tx != NULL)
If this field is non-zero along with tx != NULL then the mailbox controller driver will wait specified milliseconds for send data transfer to complete else the mailbox controller driver will not wait.
The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/mailbox.h

29.31 platform_uart_data Struct Reference

```
#include <fdt_helper.h>
```

Data Fields

- unsigned long addr
- unsigned long freq
- unsigned long baud
- unsigned long reg_shift
- unsigned long reg_io_width
- unsigned long reg_offset

29.31.1 Field Documentation

29.31.1.1 addr

```
unsigned long platform_uart_data::addr
```

29.31.1.2 baud

```
unsigned long platform_uart_data::baud
```

29.31.1.3 freq

```
unsigned long platform_uart_data::freq
```

29.31.1.4 reg_io_width

```
unsigned long platform_uart_data::reg_io_width
```

29.31.1.5 reg_offset

```
unsigned long platform_uart_data::reg_offset
```

29.31.1.6 reg_shift

```
unsigned long platform_uart_data::reg_shift
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_helper.h

29.32 plic_data Struct Reference

```
#include <plic.h>
```

Data Fields

- struct sbi_irqchip_device irqchip
- unsigned long addr
- unsigned long size
- unsigned long num_src
- unsigned long flags
- void * pm_data
- s16 context_map [ ][2]

29.32.1 Field Documentation

29.32.1.1 addr

```
unsigned long plic_data::addr
```

29.32.1.2 context_map

```
s16 plic_data::context_map[][2]
```

29.32.1.3 flags

```
unsigned long plic_data::flags
```

29.32.1.4 irqchip

```
struct sbi_irqchip_device plic_data::irqchip
```

29.32.1.5 num_src

```
unsigned long plic_data::num_src
```

29.32.1.6 pm_data

```
void* plic_data::pm_data
```

29.32.1.7 size

```
unsigned long plic_data::size
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/plic.h

29.33 plicsw_data Struct Reference

```
#include <andes_plicsw.h>
```

Data Fields

- unsigned long addr
- unsigned long size
- uint32_t hart_count

29.33.1 Field Documentation

29.33.1.1 addr

```
unsigned long plicsw_data::addr
```

29.33.1.2 hart_count

```
uint32_t plicsw_data::hart_count
```

29.33.1.3 size

```
unsigned long plicsw_data::size
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/ipi/andes_plicsw.h

29.34 plmt_data Struct Reference

```
#include <andes_plmt.h>
```

Data Fields

- u32 hart_count
- unsigned long size
- unsigned long timer_freq
- volatile u64 * time_val
- volatile u64 * time_cmp

29.34.1 Field Documentation

29.34.1.1 hart_count

```
u32 plmt_data::hart_count
```

29.34.1.2 size

```
unsigned long plmt_data::size
```

29.34.1.3 time_cmp

```
volatile u64* plmt_data::time_cmp
```

29.34.1.4 time_val

```
volatile u64* plmt_data::time_val
```

29.34.1.5 timer_freq

```
unsigned long plmt_data::timer_freq
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/timer/andes_plmt.h

29.35 regmap Struct Reference

```
#include <regmap.h>
```

Data Fields

- unsigned int id
- int reg_shift
- int reg_stride
- unsigned int reg_base
- unsigned int reg_max
- int(* reg_read)(struct regmap *rmap, unsigned int reg, unsigned int *val)
- int(* reg_write)(struct regmap *rmap, unsigned int reg, unsigned int val)
- int(* reg_update_bits)(struct regmap *rmap, unsigned int reg, unsigned int mask, unsigned int val)
- struct sbi_dlist node

29.35.1 Detailed Description

Representation of a regmap instance

29.35.2 Field Documentation

29.35.2.1 id

`unsigned int regmap::id`
Uniquie ID of the regmap instance assigned by the driver

29.35.2.2 node

`struct sbi_dlist regmap::node`
List

29.35.2.3 reg_base

`unsigned int regmap::reg_base`

29.35.2.4 reg_max

`unsigned int regmap::reg_max`

29.35.2.5 reg_read

`int(* regmap::reg_read) (struct regmap *rmap, unsigned int reg, unsigned int *val)`
Read a regmap register

29.35.2.6 reg_shift

`int regmap::reg_shift`
Configuration of regmap registers

29.35.2.7 reg_stride

`int regmap::reg_stride`

29.35.2.8 reg_update_bits

`int(* regmap::reg_update_bits) (struct regmap *rmap, unsigned int reg, unsigned int mask, unsigned int val)`
Read-modify-write a regmap register

29.35.2.9 reg_write

`int(* regmap::reg_write) (struct regmap *rmap, unsigned int reg, unsigned int val)`
Write a regmap register
The documentation for this struct was generated from the following file:

- include/sbi_utils/regmap/regmap.h

29.36 rpmi_base_get_attributes_resp Struct Reference

`#include <rpmi_msgprot.h>`

Data Fields

- s32 status_code
- u32 f0
- u32 f1
- u32 f2
- u32 f3

29.36.1 Field Documentation

29.36.1.1 f0

```
u32 rpmi_base_get_attributes_resp::f0
```

29.36.1.2 f1

```
u32 rpmi_base_get_attributes_resp::f1
```

29.36.1.3 f2

```
u32 rpmi_base_get_attributes_resp::f2
```

29.36.1.4 f3

```
u32 rpmi_base_get_attributes_resp::f3
```

29.36.1.5 status_code

```
s32 rpmi_base_get_attributes_resp::status_code
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.37 rpmi_base_get_platform_info_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 plat_info_len
- char plat_info [ ]

29.37.1 Field Documentation

29.37.1.1 plat_info

```
char rpmi_base_get_platform_info_resp::plat_info[]
```

29.37.1.2 plat_info_len

```
u32 rpmi_base_get_platform_info_resp::plat_info_len
```

29.37.1.3 status

```
s32 rpmi_base_get_platform_info_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.38 rpmi_clock_get_attributes_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 clock_id

29.38.1 Field Documentation

29.38.1.1 clock_id

```
u32 rpmi_clock_get_attributes_req::clock_id
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.39 rpmi_clock_get_attributes_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 flags
- u32 num_rates
- u32 transition_latency
- u8 name [16]

29.39.1 Field Documentation

29.39.1.1 flags

```
u32 rpmi_clock_get_attributes_resp::flags
```

29.39.1.2 name

```
u8 rpmi_clock_get_attributes_resp::name[16]
```

29.39.1.3 num_rates

```
u32 rpmi_clock_get_attributes_resp::num_rates
```

29.39.1.4 status

```
s32 rpmi_clock_get_attributes_resp::status
```

29.39.1.5 transition_latency

```
u32 rpmi_clock_get_attributes_resp::transition_latency
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.40 rpmi_clock_get_config_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 clock_id

29.40.1 Field Documentation

29.40.1.1 clock_id

```
u32 rpmi_clock_get_config_req::clock_id
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.41 rpmi_clock_get_config_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 config

29.41.1 Field Documentation

29.41.1.1 config

```
u32 rpmi_clock_get_config_resp::config
```

29.41.1.2 status

```
s32 rpmi_clock_get_config_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.42 rpmi_clock_get_num_clocks_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 num_clocks

29.42.1 Field Documentation

29.42.1.1 num_clocks

```
u32 rpmi_clock_get_num_clocks_resp::num_clocks
```

29.42.1.2 status

```
s32 rpmi_clock_get_num_clocks_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.43 rpmi_clock_get_rate_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 clock_id

29.43.1 Field Documentation

29.43.1.1 clock_id

```
u32 rpmi_clock_get_rate_req::clock_id
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.44 rpmi_clock_get_rate_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 clock_rate_low
- u32 clock_rate_high

29.44.1 Field Documentation

29.44.1.1 clock_rate_high

```
u32 rpmi_clock_get_rate_resp::clock_rate_high
```

29.44.1.2 clock_rate_low

```
u32 rpmi_clock_get_rate_resp::clock_rate_low
```

29.44.1.3 status

```
s32 rpmi_clock_get_rate_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.45 rpmi_clock_get_supported_rates_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 clock_id
- u32 clock_rate_index

29.45.1 Field Documentation

29.45.1.1 clock_id

```
u32 rpmi_clock_get_supported_rates_req::clock_id
```

29.45.1.2 clock_rate_index

```
u32 rpmi_clock_get_supported_rates_req::clock_rate_index
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.46 rpmi_clock_get_supported_rates_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 flags
- u32 remaining
- u32 returned
- u32 clock_rate [0]

29.46.1 Field Documentation

29.46.1.1 clock_rate

```
u32 rpmi_clock_get_supported_rates_resp::clock_rate[0]
```

29.46.1.2 flags

```
u32 rpmi_clock_get_supported_rates_resp::flags
```

29.46.1.3 remaining

```
u32 rpmi_clock_get_supported_rates_resp::remaining
```

29.46.1.4 returned

```
u32 rpmi_clock_get_supported_rates_resp::returned
```

29.46.1.5 status

```
s32 rpmi_clock_get_supported_rates_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.47 rpmi_clock_set_config_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 clock_id
- u32 config

29.47.1 Field Documentation

29.47.1.1 clock_id

```
u32 rpmi_clock_set_config_req::clock_id
```

29.47.1.2 config

```
u32 rpmi_clock_set_config_req::config
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.48 rpmi_clock_set_config_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status

29.48.1 Field Documentation

29.48.1.1 status

```
s32 rpmi_clock_set_config_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.49 rpmi_clock_set_rate_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 clock_id
- u32 flags
- u32 clock_rate_low
- u32 clock_rate_high

29.49.1 Field Documentation

29.49.1.1 clock_id

```
u32 rpmi_clock_set_rate_req::clock_id
```

29.49.1.2 clock_rate_high

```
u32 rpmi_clock_set_rate_req::clock_rate_high
```

29.49.1.3 clock_rate_low

```
u32 rpmi_clock_set_rate_req::clock_rate_low
```

29.49.1.4 flags

```
u32 rpmi_clock_set_rate_req::flags
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.50 rpmi_clock_set_rate_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status

29.50.1 Field Documentation

29.50.1.1 status

```
s32 rpmi_clock_set_rate_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.51 rpmi_cppc_get_fastchan_offset_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 hart_id

29.51.1 Field Documentation

29.51.1.1 hart_id

```
u32 rpmi_cppc_get_fastchan_offset_req::hart_id
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.52 rpmi_cppc_get_fastchan_offset_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 fc_perf_request_offset_lo
- u32 fc_perf_request_offset_hi
- u32 fc_perf_feedback_offset_lo
- u32 fc_perf_feedback_offset_hi

29.52.1 Field Documentation

29.52.1.1 fc_perf_feedback_offset_hi

```
u32 rpmi_cppc_get_fastchan_offset_resp::fc_perf_feedback_offset_hi
```

29.52.1.2 fc_perf_feedback_offset_lo

```
u32 rpmi_cppc_get_fastchan_offset_resp::fc_perf_feedback_offset_lo
```

29.52.1.3 fc_perf_request_offset_hi

```
u32 rpmi_cppc_get_fastchan_offset_resp::fc_perf_request_offset_hi
```

29.52.1.4 fc_perf_request_offset_lo

```
u32 rpmi_cppc_get_fastchan_offset_resp::fc_perf_request_offset_lo
```

29.52.1.5 status

```
s32 rpmi_cppc_get_fastchan_offset_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.53 rpmi_cppc_get_fastchan_region_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 flags
- u32 region_addr_lo
- u32 region_addr_hi
- u32 region_size_lo
- u32 region_size_hi
- u32 db_addr_lo
- u32 db_addr_hi
- u32 db_setmask_lo
- u32 db_setmask_hi
- u32 db_preservemask_lo
- u32 db_preservemask_hi

29.53.1 Field Documentation

29.53.1.1 db_addr_hi

```
u32 rpmi_cppc_get_fastchan_region_resp::db_addr_hi
```

29.53.1.2 db_addr_lo

```
u32 rpmi_cppc_get_fastchan_region_resp::db_addr_lo
```

29.53.1.3 db_preservemask_hi

```
u32 rpmi_cppc_get_fastchan_region_resp::db_preservemask_hi
```

29.53.1.4 db_preservemask_lo

```
u32 rpmi_cppc_get_fastchan_region_resp::db_preservemask_lo
```

29.53.1.5 db_setmask_hi

```
u32 rpmi_cppc_get_fastchan_region_resp::db_setmask_hi
```

29.53.1.6 db_setmask_lo

```
u32 rpmi_cppc_get_fastchan_region_resp::db_setmask_lo
```

29.53.1.7 flags

```
u32 rpmi_cppc_get_fastchan_region_resp::flags
```

29.53.1.8 region_addr_hi

```
u32 rpmi_cppc_get_fastchan_region_resp::region_addr_hi
```

29.53.1.9 region_addr_lo

```
u32 rpmi_cppc_get_fastchan_region_resp::region_addr_lo
```

29.53.1.10 region_size_hi

```
u32 rpmi_cppc_get_fastchan_region_resp::region_size_hi
```

29.53.1.11 region_size_lo

```
u32 rpmi_cppc_get_fastchan_region_resp::region_size_lo
```

29.53.1.12 status

```
s32 rpmi_cppc_get_fastchan_region_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.54 rpmi_cppc_hart_list_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 start_index

29.54.1 Field Documentation

29.54.1.1 start_index

```
u32 rpmi_cppc_hart_list_req::start_index
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.55 rpmi_cppc_hart_list_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 remaining
- u32 returned
- u32 hartid [(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) -(sizeof(u32) ∗3))/sizeof(u32)]

29.55.1 Field Documentation

29.55.1.1 hartid

```
u32 rpmi_cppc_hart_list_resp::hartid[(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) -(sizeof(u32)
*3))/sizeof(u32)]
```

29.55.1.2 remaining

```
u32 rpmi_cppc_hart_list_resp::remaining
```

29.55.1.3 returned

```
u32 rpmi_cppc_hart_list_resp::returned
```

29.55.1.4 status

```
s32 rpmi_cppc_hart_list_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.56 rpmi_cppc_probe_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 hart_id
- u32 reg_id

29.56.1 Field Documentation

29.56.1.1 hart_id

```
u32 rpmi_cppc_probe_req::hart_id
```

29.56.1.2 reg_id

```
u32 rpmi_cppc_probe_req::reg_id
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.57 rpmi_cppc_probe_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 reg_len

29.57.1 Field Documentation

29.57.1.1 reg_len

```
u32 rpmi_cppc_probe_resp::reg_len
```

29.57.1.2 status

```
s32 rpmi_cppc_probe_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.58 rpmi_cppc_read_reg_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 hart_id
- u32 reg_id

29.58.1 Field Documentation

29.58.1.1 hart_id

```
u32 rpmi_cppc_read_reg_req::hart_id
```

29.58.1.2 reg_id

```
u32 rpmi_cppc_read_reg_req::reg_id
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.59 rpmi_cppc_read_reg_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 data_lo
- u32 data_hi

29.59.1 Field Documentation

29.59.1.1 data_hi

```
u32 rpmi_cppc_read_reg_resp::data_hi
```

29.59.1.2 data_lo

```
u32 rpmi_cppc_read_reg_resp::data_lo
```

29.59.1.3 status

```
s32 rpmi_cppc_read_reg_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.60 rpmi_cppc_write_reg_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 hart_id
- u32 reg_id
- u32 data_lo
- u32 data_hi

29.60.1 Field Documentation

29.60.1.1 data_hi

```
u32 rpmi_cppc_write_reg_req::data_hi
```

29.60.1.2 data_lo

```
u32 rpmi_cppc_write_reg_req::data_lo
```

29.60.1.3 hart_id

```
u32 rpmi_cppc_write_reg_req::hart_id
```

29.60.1.4 reg_id

```
u32 rpmi_cppc_write_reg_req::reg_id
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.61 rpmi_cppc_write_reg_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status

29.61.1 Field Documentation

29.61.1.1 status

```
s32 rpmi_cppc_write_reg_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.62 rpmi_enable_notification_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 eventid

29.62.1 Detailed Description

RPMI enable notification request

29.62.2 Field Documentation

29.62.2.1 eventid

```
u32 rpmi_enable_notification_req::eventid
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.63 rpmi_enable_notification_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status

29.63.1 Detailed Description

RPMI enable notification response

29.63.2 Field Documentation

29.63.2.1 status

```
s32 rpmi_enable_notification_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.64 rpmi_hsm_get_hart_list_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 start_index

29.64.1 Field Documentation

29.64.1.1 start_index

```
u32 rpmi_hsm_get_hart_list_req::start_index
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.65 rpmi_hsm_get_hart_list_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 remaining
- u32 returned
- u32 hartid [(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) -(sizeof(u32) ∗3))/sizeof(u32)]

29.65.1 Field Documentation

29.65.1.1 hartid

```
u32 rpmi_hsm_get_hart_list_resp::hartid[(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) -(sizeof(u32)
*3))/sizeof(u32)]
```

29.65.1.2 remaining

```
u32 rpmi_hsm_get_hart_list_resp::remaining
```

29.65.1.3 returned

```
u32 rpmi_hsm_get_hart_list_resp::returned
```

29.65.1.4 status

```
s32 rpmi_hsm_get_hart_list_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.66 rpmi_hsm_get_hart_status_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 hartid

29.66.1 Field Documentation

29.66.1.1 hartid

`u32 rpmi_hsm_get_hart_status_req::hartid`

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.67 rpmi_hsm_get_hart_status_resp Struct Reference

`#include <rpmi_msgprot.h>`

Data Fields

- s32 status
- u32 hart_status

29.67.1 Field Documentation

29.67.1.1 hart_status

`u32 rpmi_hsm_get_hart_status_resp::hart_status`

29.67.1.2 status

`s32 rpmi_hsm_get_hart_status_resp::status`

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.68 rpmi_hsm_get_susp_info_req Struct Reference

`#include <rpmi_msgprot.h>`

Data Fields

- u32 suspend_type

29.68.1 Field Documentation

29.68.1.1 suspend_type

`u32 rpmi_hsm_get_susp_info_req::suspend_type`

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.69 rpmi_hsm_get_susp_info_resp Struct Reference

`#include <rpmi_msgprot.h>`

Data Fields

- s32 status
- u32 flags
- u32 entry_latency_us
- u32 exit_latency_us
- u32 wakeup_latency_us
- u32 min_residency_us

29.69.1 Field Documentation

29.69.1.1 entry_latency_us

```
u32 rpmi_hsm_get_susp_info_resp::entry_latency_us
```

29.69.1.2 exit_latency_us

```
u32 rpmi_hsm_get_susp_info_resp::exit_latency_us
```

29.69.1.3 flags

```
u32 rpmi_hsm_get_susp_info_resp::flags
```

29.69.1.4 min_residency_us

```
u32 rpmi_hsm_get_susp_info_resp::min_residency_us
```

29.69.1.5 status

```
s32 rpmi_hsm_get_susp_info_resp::status
```

29.69.1.6 wakeup_latency_us

```
u32 rpmi_hsm_get_susp_info_resp::wakeup_latency_us
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.70 rpmi_hsm_get_susp_types_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 start_index

29.70.1 Field Documentation

29.70.1.1 start_index

```
u32 rpmi_hsm_get_susp_types_req::start_index
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.71 rpmi_hsm_get_susp_types_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 remaining
- u32 returned
- u32 types [(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) -(sizeof(u32) ∗3))/sizeof(u32)]

29.71.1 Field Documentation

29.71.1.1 remaining

```
u32 rpmi_hsm_get_susp_types_resp::remaining
```

29.71.1.2 returned

```
u32 rpmi_hsm_get_susp_types_resp::returned
```

29.71.1.3 status

```
s32 rpmi_hsm_get_susp_types_resp::status
```

29.71.1.4 types

```
u32 rpmi_hsm_get_susp_types_resp::types[(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) -(sizeof(u32)
*3))/sizeof(u32)]
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.72 rpmi_hsm_hart_start_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 hartid
- u32 start_addr_lo
- u32 start_addr_hi

29.72.1 Field Documentation

29.72.1.1 hartid

```
u32 rpmi_hsm_hart_start_req::hartid
```

29.72.1.2 start_addr_hi

```
u32 rpmi_hsm_hart_start_req::start_addr_hi
```

29.72.1.3 start_addr_lo

```
u32 rpmi_hsm_hart_start_req::start_addr_lo
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.73 rpmi_hsm_hart_start_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status

29.73.1 Field Documentation

29.73.1.1 status

`s32 rpmi_hsm_hart_start_resp::status`

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.74 rpmi_hsm_hart_stop_req Struct Reference

`#include <rpmi_msgprot.h>`

Data Fields

- u32 hartid

29.74.1 Field Documentation

29.74.1.1 hartid

`u32 rpmi_hsm_hart_stop_req::hartid`

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.75 rpmi_hsm_hart_stop_resp Struct Reference

`#include <rpmi_msgprot.h>`

Data Fields

- s32 status

29.75.1 Field Documentation

29.75.1.1 status

`s32 rpmi_hsm_hart_stop_resp::status`

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.76 rpmi_hsm_hart_susp_req Struct Reference

`#include <rpmi_msgprot.h>`

Data Fields

- u32 hartid
- u32 suspend_type
- u32 resume_addr_lo
- u32 resume_addr_hi

29.76.1 Field Documentation

29.76.1.1 hartid

`u32 rpmi_hsm_hart_susp_req::hartid`

29.76.1.2 resume_addr_hi

```
u32 rpmi_hsm_hart_susp_req::resume_addr_hi
```

29.76.1.3 resume_addr_lo

```
u32 rpmi_hsm_hart_susp_req::resume_addr_lo
```

29.76.1.4 suspend_type

```
u32 rpmi_hsm_hart_susp_req::suspend_type
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.77 rpmi_hsm_hart_susp_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status

29.77.1 Field Documentation

29.77.1.1 status

```
s32 rpmi_hsm_hart_susp_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.78 rpmi_message Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- struct rpmi_message_header header
- u8 data [0]

29.78.1 Detailed Description

RPMI Message

29.78.2 Field Documentation

29.78.2.1 data

```
u8 rpmi_message::data[0]
```

29.78.2.2 header

```
struct rpmi_message_header rpmi_message::header
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.79 rpmi_message_args Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 flags
- enum rpmi_message_type type
- u8 service_id
- u32 tx_endian_words
- u32 rx_endian_words
- u16 rx_token
- u32 rx_data_len

29.79.1 Detailed Description

RPMI Message Arguments

29.79.2 Field Documentation

29.79.2.1 flags

```
u32 rpmi_message_args::flags
```

29.79.2.2 rx_data_len

```
u32 rpmi_message_args::rx_data_len
```

29.79.2.3 rx_endian_words

```
u32 rpmi_message_args::rx_endian_words
```

29.79.2.4 rx_token

```
u16 rpmi_message_args::rx_token
```

29.79.2.5 service_id

```
u8 rpmi_message_args::service_id
```

29.79.2.6 tx_endian_words

```
u32 rpmi_message_args::tx_endian_words
```

29.79.2.7 type

```
enum rpmi_message_type rpmi_message_args::type
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.80 rpmi_message_header Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- le16_t servicegroup_id
- uint8_t service_id
- uint8_t flags
- le16_t datalen
- le16_t token

29.80.1 Detailed Description

RPMI Message Header

29.80.2 Field Documentation

29.80.2.1 datalen

```
le16_t rpmi_message_header::datalen
```

29.80.2.2 flags

```
uint8_t rpmi_message_header::flags
```

29.80.2.3 service_id

```
uint8_t rpmi_message_header::service_id
```

29.80.2.4 servicegroup_id

```
le16_t rpmi_message_header::servicegroup_id
```

29.80.2.5 token

```
le16_t rpmi_message_header::token
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.81 rpmi_sysrst_get_reset_attributes_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 flags

29.81.1 Detailed Description

Response for system reset attributes

29.81.2 Field Documentation

29.81.2.1 flags

```
u32 rpmi_sysrst_get_reset_attributes_resp::flags
```

29.81.2.2 status

```
s32 rpmi_sysrst_get_reset_attributes_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.82 rpmi_syssusp_get_attr_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 susp_type

29.82.1 Detailed Description

Request for system suspend attributes

29.82.2 Field Documentation

29.82.2.1 susp_type

```
u32 rpmi_syssusp_get_attr_req::susp_type
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.83 rpmi_syssusp_get_attr_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status
- u32 flags

29.83.1 Detailed Description

Response for system suspend attributes

29.83.2 Field Documentation

29.83.2.1 flags

```
u32 rpmi_syssusp_get_attr_resp::flags
```

29.83.2.2 status

```
s32 rpmi_syssusp_get_attr_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.84 rpmi_syssusp_suspend_req Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- u32 hartid
- u32 suspend_type
- u32 resume_addr_lo
- u32 resume_addr_hi

29.84.1 Field Documentation

29.84.1.1 hartid

```
u32 rpmi_syssusp_suspend_req::hartid
```

29.84.1.2 resume_addr_hi

```
u32 rpmi_syssusp_suspend_req::resume_addr_hi
```

29.84.1.3 resume_addr_lo

```
u32 rpmi_syssusp_suspend_req::resume_addr_lo
```

29.84.1.4 suspend_type

```
u32 rpmi_syssusp_suspend_req::suspend_type
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.85 rpmi_syssusp_suspend_resp Struct Reference

```
#include <rpmi_msgprot.h>
```

Data Fields

- s32 status

29.85.1 Field Documentation

29.85.1.1 status

```
s32 rpmi_syssusp_suspend_resp::status
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/mailbox/rpmi_msgprot.h

29.86 sbi_console_device Struct Reference

```
#include <sbi_console.h>
```

Data Fields

- char name [32]
- void(∗ console_putc)(char ch)
- unsigned long(∗ console_puts)(const char ∗str, unsigned long len)
- int(∗ console_getc)(void)

29.86.1 Field Documentation

29.86.1.1 console_getc

`int(* sbi_console_device::console_getc) (void)`
Read a character from the console input

29.86.1.2 console_putc

`void(* sbi_console_device::console_putc) (char ch)`
Write a character to the console output

29.86.1.3 console_puts

`unsigned long(* sbi_console_device::console_puts) (const char *str, unsigned long len)`
Write a character string to the console output

29.86.1.4 name

`char sbi_console_device::name[32]`
Name of the console device
The documentation for this struct was generated from the following file:

- include/sbi/sbi_console.h

29.87 sbi_cppc_device Struct Reference

`#include <sbi_cppc.h>`

Data Fields

- char name [32]
- int(∗ cppc_probe)(unsigned long reg)
- int(∗ cppc_read)(unsigned long reg, uint64_t ∗val)
- int(∗ cppc_write)(unsigned long reg, uint64_t val)

29.87.1 Detailed Description

CPPC device

29.87.2 Field Documentation

29.87.2.1 cppc_probe

`int(* sbi_cppc_device::cppc_probe) (unsigned long reg)`
probe - returns register width if implemented, 0 otherwise

29.87.2.2 cppc_read

`int(* sbi_cppc_device::cppc_read) (unsigned long reg, uint64_t *val)`
read the cppc register

29.87.2.3 cppc_write

`int(* sbi_cppc_device::cppc_write) (unsigned long reg, uint64_t val)`
write to the cppc register

29.87.2.4 name

```
char sbi_cppc_device::name[32]
```

Name of the CPPC device

The documentation for this struct was generated from the following file:

- include/sbi/sbi_cppc.h

29.88 sbi_cpu_idle_state Struct Reference

```
#include <fdt_fixup.h>
```

Data Fields

- const char * name
- uint32_t suspend_param
- bool local_timer_stop
- uint32_t entry_latency_us
- uint32_t exit_latency_us
- uint32_t min_residency_us
- uint32_t wakeup_latency_us

29.88.1 Field Documentation

29.88.1.1 entry_latency_us

```
uint32_t sbi_cpu_idle_state::entry_latency_us
```

29.88.1.2 exit_latency_us

```
uint32_t sbi_cpu_idle_state::exit_latency_us
```

29.88.1.3 local_timer_stop

```
bool sbi_cpu_idle_state::local_timer_stop
```

29.88.1.4 min_residency_us

```
uint32_t sbi_cpu_idle_state::min_residency_us
```

29.88.1.5 name

```
const char* sbi_cpu_idle_state::name
```

29.88.1.6 suspend_param

```
uint32_t sbi_cpu_idle_state::suspend_param
```

29.88.1.7 wakeup_latency_us

```
uint32_t sbi_cpu_idle_state::wakeup_latency_us
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_fixup.h

29.89 sbi_dbtr_data_msg Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- unsigned long tstate
- unsigned long tdata1
- unsigned long tdata2
- unsigned long tdata3

29.89.1 Field Documentation

29.89.1.1 tdata1

```
unsigned long sbi_dbtr_data_msg::tdata1
```

29.89.1.2 tdata2

```
unsigned long sbi_dbtr_data_msg::tdata2
```

29.89.1.3 tdata3

```
unsigned long sbi_dbtr_data_msg::tdata3
```

29.89.1.4 tstate

```
unsigned long sbi_dbtr_data_msg::tstate
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.90 sbi_dbtr_hart_triggers_state Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- struct sbi_dbtr_trigger triggers [RV_MAX_TRIGGERS]
- struct sbi_dbtr_shmem shmem
- u32 total_trigs
- u32 available_trigs
- u32 hartid
- u32 probed

29.90.1 Field Documentation

29.90.1.1 available_trigs

```
u32 sbi_dbtr_hart_triggers_state::available_trigs
```

29.90.1.2 hartid

```
u32 sbi_dbtr_hart_triggers_state::hartid
```

29.90.1.3 probed

```
u32 sbi_dbtr_hart_triggers_state::probed
```

29.90.1.4 shmem

```
struct sbi_dbtr_shmem sbi_dbtr_hart_triggers_state::shmem
```

29.90.1.5 total_trigs

```
u32 sbi_dbtr_hart_triggers_state::total_trigs
```

29.90.1.6 triggers

```
struct sbi_dbtr_trigger sbi_dbtr_hart_triggers_state::triggers[RV_MAX_TRIGGERS]
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.91 sbi_dbtr_id_msg Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- unsigned long idx

29.91.1 Field Documentation

29.91.1.1 idx

```
unsigned long sbi_dbtr_id_msg::idx
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.92 sbi_dbtr_shmem Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- unsigned long phys_lo
- unsigned long phys_hi

29.92.1 Field Documentation

29.92.1.1 phys_hi

```
unsigned long sbi_dbtr_shmem::phys_hi
```

29.92.1.2 phys_lo

```
unsigned long sbi_dbtr_shmem::phys_lo
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.93 sbi_dbtr_shmem_entry Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- struct sbi_dbtr_data_msg data
- struct sbi_dbtr_id_msg id

29.93.1 Detailed Description

SBI shared mem messages layout

29.93.2 Field Documentation

29.93.2.1 data

```
struct sbi_dbtr_data_msg sbi_dbtr_shmem_entry::data
```

29.93.2.2 id

```
struct sbi_dbtr_id_msg sbi_dbtr_shmem_entry::id
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.94 sbi_dbtr_trigger Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- unsigned long index
- unsigned long type_mask
- unsigned long state
- unsigned long tdata1
- unsigned long tdata2
- unsigned long tdata3

29.94.1 Field Documentation

29.94.1.1 index

```
unsigned long sbi_dbtr_trigger::index
```

29.94.1.2 state

```
unsigned long sbi_dbtr_trigger::state
```

29.94.1.3 tdata1

```
unsigned long sbi_dbtr_trigger::tdata1
```

29.94.1.4 tdata2

```
unsigned long sbi_dbtr_trigger::tdata2
```

29.94.1.5 tdata3

```
unsigned long sbi_dbtr_trigger::tdata3
```

29.94.1.6 type_mask

```
unsigned long sbi_dbtr_trigger::type_mask
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.95 sbi_dlist Struct Reference

```
#include <sbi_list.h>
```

Data Fields

- struct sbi_dlist * next
- struct sbi_dlist * prev

29.95.1 Field Documentation

29.95.1.1 next

```
struct sbi_dlist* sbi_dlist::next
```

29.95.1.2 prev

```
struct sbi_dlist * sbi_dlist::prev
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_list.h

29.96 sbi_domain Struct Reference

```
#include <sbi_domain.h>
```

Data Fields

- struct sbi_dlist node
- struct sbi_domain_data_priv data_priv
- u32 index
- struct sbi_hartmask assigned_harts
- spinlock_t assigned_harts_lock
- char name [64]
- const struct sbi_hartmask * possible_harts
- struct sbi_domain_memregion * regions
- u32 boot_hartid
- unsigned long next_arg1
- unsigned long next_addr
- unsigned long next_mode
- bool system_reset_allowed
- bool system_suspend_allowed
- bool fw_region_inited

29.96.1 Detailed Description

Representation of OpenSBI domain

29.96.2 Field Documentation

29.96.2.1 assigned_harts

```
struct sbi_hartmask sbi_domain::assigned_harts
```

HARTs assigned to this domain

29.96.2.2 assigned_harts_lock

```
spinlock_t sbi_domain::assigned_harts_lock
```

Spinlock for accessing assigned_harts

29.96.2.3 boot_hartid

`u32 sbi_domain::boot_hartid`
HART id of the HART booting this domain

29.96.2.4 data_priv

`struct sbi_domain_data_priv sbi_domain::data_priv`
Internal state of per-domain data

29.96.2.5 fw_region_inited

`bool sbi_domain::fw_region_inited`
Identifies whether to include the firmware region

29.96.2.6 index

`u32 sbi_domain::index`
Logical index of this domain

29.96.2.7 name

`char sbi_domain::name[64]`
Name of this domain

29.96.2.8 next_addr

`unsigned long sbi_domain::next_addr`
Address of next booting stage for this domain

29.96.2.9 next_arg1

`unsigned long sbi_domain::next_arg1`
Arg1 (or 'a1' register) of next booting stage for this domain

29.96.2.10 next_mode

`unsigned long sbi_domain::next_mode`
Privilege mode of next booting stage for this domain

29.96.2.11 node

`struct sbi_dlist sbi_domain::node`
Node in linked list of domains

29.96.2.12 possible_harts

`const struct sbi_hartmask* sbi_domain::possible_harts`
Possible HARTs in this domain

29.96.2.13 regions

`struct sbi_domain_memregion* sbi_domain::regions`
Array of memory regions terminated by a region with order zero

29.96.2.14 system_reset_allowed

`bool sbi_domain::system_reset_allowed`
Is domain allowed to reset the system

29.96.2.15 system_suspend_allowed

`bool sbi_domain::system_suspend_allowed`
Is domain allowed to suspend the system
The documentation for this struct was generated from the following file:

- include/sbi/sbi_domain.h

29.97 sbi_domain_data Struct Reference

`#include <sbi_domain_data.h>`

Data Fields

- struct sbi_dlist head
- unsigned long data_idx
- unsigned long data_size
- int(∗ data_setup)(struct sbi_domain ∗dom, struct sbi_domain_data ∗data, void ∗data_ptr)
- void(∗ data_cleanup)(struct sbi_domain ∗dom, struct sbi_domain_data ∗data, void ∗data_ptr)

29.97.1 Detailed Description

Representation of a domain data

29.97.2 Field Documentation

29.97.2.1 data_cleanup

`void(* sbi_domain_data::data_cleanup) (struct sbi_domain *dom, struct sbi_domain_data *data, void *data_ptr)`
Optional callback to cleanup domain data

29.97.2.2 data_idx

`unsigned long sbi_domain_data::data_idx`
Identifier which used to locate per-domain data
Note: initialized by domain framework

29.97.2.3 data_setup

`int(* sbi_domain_data::data_setup) (struct sbi_domain *dom, struct sbi_domain_data *data, void *data_ptr)`
Optional callback to setup domain data

29.97.2.4 data_size

`unsigned long sbi_domain_data::data_size`
Size of per-domain data

29.97.2.5 head

`struct sbi_dlist sbi_domain_data::head`
Head is used for maintaining data list
Note: initialized by domain framework
The documentation for this struct was generated from the following file:

- include/sbi/sbi_domain_data.h

29.98 sbi_domain_data_priv Struct Reference

```
#include <sbi_domain_data.h>
```

Data Fields

- void * idx_to_data_ptr [SBI_DOMAIN_MAX_DATA_PTRS]

29.98.1 Detailed Description

Representation of per-domain data

29.98.2 Field Documentation

29.98.2.1 idx_to_data_ptr

```
void* sbi_domain_data_priv::idx_to_data_ptr[SBI_DOMAIN_MAX_DATA_PTRS]
```

Array of domain data pointers indexed by domain data identifier
The documentation for this struct was generated from the following file:

- include/sbi/sbi_domain_data.h

29.99 sbi_domain_memregion Struct Reference

```
#include <sbi_domain.h>
```

Data Fields

- unsigned long order
- unsigned long base
- unsigned long flags

29.99.1 Detailed Description

Representation of OpenSBI domain memory region

29.99.2 Field Documentation

29.99.2.1 base

```
unsigned long sbi_domain_memregion::base
```

Base address of memory region It must be 2^order aligned address

29.99.2.2 flags

```
unsigned long sbi_domain_memregion::flags
```

29.99.2.3 order

```
unsigned long sbi_domain_memregion::order
```

Size of memory region as power of 2 It has to be minimum 3 and maximum __riscv_xlen
The documentation for this struct was generated from the following file:

- include/sbi/sbi_domain.h

29.100 sbi_ecall_extension Struct Reference

```
#include <sbi_ecall.h>
```

Data Fields

- struct sbi_dlist head
- char name [8]
- unsigned long extid_start
- unsigned long extid_end
- bool experimental
- int(∗ register_extensions)(void)
- int(∗ probe)(unsigned long extid, unsigned long ∗out_val)
- int(∗ handle)(unsigned long extid, unsigned long funcid, struct sbi_trap_regs ∗regs, struct sbi_ecall_return ∗out)

29.100.1 Field Documentation

29.100.1.1 experimental

```
bool sbi_ecall_extension::experimental
```

29.100.1.2 extid_end

```
unsigned long sbi_ecall_extension::extid_end
```

29.100.1.3 extid_start

```
unsigned long sbi_ecall_extension::extid_start
```

29.100.1.4 handle

```
int(* sbi_ecall_extension::handle) (unsigned long extid, unsigned long funcid, struct sbi_trap_regs
*regs, struct sbi_ecall_return *out)
```

29.100.1.5 head

```
struct sbi_dlist sbi_ecall_extension::head
```

29.100.1.6 name

```
char sbi_ecall_extension::name[8]
```

29.100.1.7 probe

```
int(* sbi_ecall_extension::probe) (unsigned long extid, unsigned long *out_val)
```

29.100.1.8 register_extensions

```
int(* sbi_ecall_extension::register_extensions) (void)
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_ecall.h

29.101 sbi_ecall_return Struct Reference

```
#include <sbi_ecall.h>
```

Data Fields

- bool skip_regs_update
- unsigned long value

29.101.1 Field Documentation

29.101.1.1 skip_regs_update

```
bool sbi_ecall_return::skip_regs_update
```

29.101.1.2 value

```
unsigned long sbi_ecall_return::value
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_ecall.h

29.102 sbi_fifo Struct Reference

```
#include <sbi_fifo.h>
```

Data Fields

- void * queue
- spinlock_t qlock
- u16 entry_size
- u16 num_entries
- u16 avail
- u16 tail

29.102.1 Field Documentation

29.102.1.1 avail

```
u16 sbi_fifo::avail
```

29.102.1.2 entry_size

```
u16 sbi_fifo::entry_size
```

29.102.1.3 num_entries

```
u16 sbi_fifo::num_entries
```

29.102.1.4 qlock

```
spinlock_t sbi_fifo::qlock
```

29.102.1.5 queue

```
void* sbi_fifo::queue
```

29.102.1.6 tail

```
u16 sbi_fifo::tail
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_fifo.h

29.103 sbi_hart_ext_data Struct Reference

```
#include <sbi_hart.h>
```

Data Fields

- const unsigned int id
- const char * name

29.103.1 Field Documentation

29.103.1.1 id

```
const unsigned int sbi_hart_ext_data::id
```

29.103.1.2 name

```
const char* sbi_hart_ext_data::name
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_hart.h

29.104 sbi_hart_features Struct Reference

```
#include <sbi_hart.h>
```

Data Fields

- bool detected
- int priv_version
- unsigned long extensions [BITS_TO_LONGS(SBI_HART_EXT_MAX)]
- unsigned int pmp_count
- unsigned int pmp_addr_bits
- unsigned int pmp_log2gran
- unsigned int mhpm_mask
- unsigned int mhpm_bits

29.104.1 Field Documentation

29.104.1.1 detected

```
bool sbi_hart_features::detected
```

29.104.1.2 extensions

```
unsigned long sbi_hart_features::extensions[BITS_TO_LONGS(SBI_HART_EXT_MAX)]
```

29.104.1.3 mhpm_bits

```
unsigned int sbi_hart_features::mhpm_bits
```

29.104.1.4 mhpm_mask

```
unsigned int sbi_hart_features::mhpm_mask
```

29.104.1.5 pmp_addr_bits

```
unsigned int sbi_hart_features::pmp_addr_bits
```

29.104.1.6 pmp_count

```
unsigned int sbi_hart_features::pmp_count
```

29.104.1.7 pmp_log2gran

```
unsigned int sbi_hart_features::pmp_log2gran
```

29.104.1.8 priv_version

```
int sbi_hart_features::priv_version
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_hart.h

29.105 sbi_hartmask Struct Reference

```
#include <sbi_hartmask.h>
```

Public Member Functions

- DECLARE_BITMAP (bits, SBI_HARTMASK_MAX_BITS)

29.105.1 Detailed Description

Representation of hartmask

29.105.2 Member Function Documentation

29.105.2.1 DECLARE_BITMAP()

```
sbi_hartmask::DECLARE_BITMAP (
            bits ,
            SBI_HARTMASK_MAX_BITS  )
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_hartmask.h

29.106 sbi_hsm_device Struct Reference

```
#include <sbi_hsm.h>
```

Data Fields

- char name [32]
- int(∗ hart_start)(u32 hartid, ulong saddr)
- int(∗ hart_stop)(void)
- int(∗ hart_suspend)(u32 suspend_type, ulong mmode_resume_addr)
- void(∗ hart_resume)(void)

29.106.1 Detailed Description

Hart state managment device

29.106.2 Field Documentation

29.106.2.1 hart_resume

```
void(* sbi_hsm_device::hart_resume) (void)
```

Perform platform-specific actions to resume from a suspended state.
This includes restoring any platform state that was lost during non-retentive suspend.

29.106.2.2 hart_start

`int(* sbi_hsm_device::hart_start) (u32 hartid, ulong saddr)`
Start (or power-up) the given hart

29.106.2.3 hart_stop

`int(* sbi_hsm_device::hart_stop) (void)`
Stop (or power-down) the current hart from running.
Return SBI_ENOTSUPP if the hart does not support platform-specific stop actions.
For successful stop, the call won't return.

29.106.2.4 hart_suspend

`int(* sbi_hsm_device::hart_suspend) (u32 suspend_type, ulong mmode_resume_addr)`
Put the current hart in platform specific suspend (or low-power) state.
For successful retentive suspend, the call will return 0 when the hart resumes normal execution.
For successful non-retentive suspend, the hart will resume from the warm boot entry point.
NOTE: mmode_resume_addr(resume address) is optional hence it may or may not be honored by the platform. If its not honored then platform must ensure to resume from the warmboot address.

29.106.2.5 name

`char sbi_hsm_device::name[32]`
Name of the hart state managment device
The documentation for this struct was generated from the following file:

- include/sbi/sbi_hsm.h

29.107 sbi_ipi_device Struct Reference

`#include <sbi_ipi.h>`

Data Fields

- char name [32]
- void(* ipi_send)(u32 hart_index)
- void(* ipi_clear)(void)

29.107.1 Detailed Description

IPI hardware device

29.107.2 Field Documentation

29.107.2.1 ipi_clear

`void(* sbi_ipi_device::ipi_clear) (void)`
Clear IPI for the current hart

29.107.2.2 ipi_send

`void(* sbi_ipi_device::ipi_send) (u32 hart_index)`
Send IPI to a target HART index

29.107.2.3 name

`char sbi_ipi_device::name[32]`
Name of the IPI device
The documentation for this struct was generated from the following file:

- include/sbi/sbi_ipi.h

29.108 sbi_ipi_event_ops Struct Reference

`#include <sbi_ipi.h>`

Data Fields

- char name [32]
- int(∗ update)(struct sbi_scratch ∗scratch, struct sbi_scratch ∗remote_scratch, u32 remote_hartindex, void ∗data)
- void(∗ sync)(struct sbi_scratch ∗scratch)
- void(∗ process)(struct sbi_scratch ∗scratch)

29.108.1 Detailed Description

IPI event operations or callbacks

29.108.2 Field Documentation

29.108.2.1 name

`char sbi_ipi_event_ops::name[32]`
Name of the IPI event operations

29.108.2.2 process

`void(* sbi_ipi_event_ops::process) (struct sbi_scratch *scratch)`
Process callback to handle IPI event Note: This is a mandatory callback and it is called on the remote HART after IPI is triggered.

29.108.2.3 sync

`void(* sbi_ipi_event_ops::sync) (struct sbi_scratch *scratch)`
Sync callback to wait for remote HART Note: This is an optional callback and it is called just after triggering IPI to remote HART.

29.108.2.4 update

`int(* sbi_ipi_event_ops::update) (struct sbi_scratch *scratch, struct sbi_scratch *remote_↩scratch, u32 remote_hartindex, void *data)`
Update callback to save/enqueue data for remote HART Note: This is an optional callback and it is called just before triggering IPI to remote HART.

Returns

< 0, error or failure

SBI_IPI_UPDATE_SUCCESS, success

SBI_IPI_UPDATE_BREAK, break IPI, done on local hart

SBI_IPI_UPDATE_RETRY, need retry

The documentation for this struct was generated from the following file:

- include/sbi/sbi_ipi.h

29.109 sbi_irqchip_device Struct Reference

`#include <sbi_irqchip.h>`

Data Fields

- struct sbi_dlist node
- int(∗ warm_init)(struct sbi_irqchip_device ∗dev)
- int(∗ irq_handle)(void)

29.109.1 Detailed Description

irqchip hardware device

29.109.2 Field Documentation

29.109.2.1 irq_handle

`int(* sbi_irqchip_device::irq_handle) (void)`
Handle an IRQ from this irqchip

29.109.2.2 node

`struct sbi_dlist sbi_irqchip_device::node`
Node in the list of irqchip devices

29.109.2.3 warm_init

`int(* sbi_irqchip_device::warm_init) (struct sbi_irqchip_device *dev)`
Initialize per-hart state for the current hart
The documentation for this struct was generated from the following file:

- include/sbi/sbi_irqchip.h

29.110 sbi_ldst_data Union Reference

`#include <sbi_trap_ldst.h>`

Data Fields

- u64 data_u64
- u32 data_u32
- u8 data_bytes [8]
- ulong data_ulong

29.110.1 Field Documentation

29.110.1.1 data_bytes

`u8 sbi_ldst_data::data_bytes[8]`

29.110.1.2 data_u32

`u32 sbi_ldst_data::data_u32`

29.110.1.3 data_u64

`u64 sbi_ldst_data::data_u64`

29.110.1.4 data_ulong

`ulong sbi_ldst_data::data_ulong`

The documentation for this union was generated from the following file:

- include/sbi/sbi_trap_ldst.h

29.111 sbi_mpxy_channel Struct Reference

`#include <sbi_mpxy.h>`

Data Fields

- struct sbi_dlist head
- u32 channel_id
- struct sbi_mpxy_channel_attrs attrs
- int(∗ read_attributes)(struct sbi_mpxy_channel ∗channel, u32 ∗outmem, u32 base_attr_id, u32 attr_count)
- int(∗ write_attributes)(struct sbi_mpxy_channel ∗channel, u32 ∗inmem, u32 base_attr_id, u32 attr_count)
- int(∗ send_message_with_response)(struct sbi_mpxy_channel ∗channel, u32 msg_id, void ∗msgbuf, u32 msg_len, void ∗respbuf, u32 resp_max_len, unsigned long ∗resp_len)
- int(∗ send_message_without_response)(struct sbi_mpxy_channel ∗channel, u32 msg_id, void ∗msgbuf, u32 msg_len)
- int(∗ get_notification_events)(struct sbi_mpxy_channel ∗channel, void ∗eventsbuf, u32 bufsize, unsigned long ∗events_len)
- void(∗ switch_eventsstate)(u32 enable)

29.111.1 Detailed Description

A Message proxy channel accessible through SBI interface

29.111.2 Field Documentation

29.111.2.1 attrs

`struct sbi_mpxy_channel_attrs sbi_mpxy_channel::attrs`

29.111.2.2 channel_id

`u32 sbi_mpxy_channel::channel_id`

29.111.2.3 get_notification_events

`int(* sbi_mpxy_channel::get_notification_events) (struct sbi_mpxy_channel *channel, void *eventsbuf, u32 bufsize, unsigned long *events_len)`

Get notifications events if supported on a channel NOTE: eventsbuf requires little-endian byte-ordering

29.111.2.4 head

`struct sbi_dlist sbi_mpxy_channel::head`

List head to a set of channels

29.111.2.5 read_attributes

`int(* sbi_mpxy_channel::read_attributes) (struct sbi_mpxy_channel *channel, u32 *outmem, u32 base_attr_id, u32 attr_count)`

Read message protocol attributes NOTE: inmem requires little-endian byte-ordering

29.111.2.6 send_message_with_response

`int(* sbi_mpxy_channel::send_message_with_response) (struct sbi_mpxy_channel *channel, u32 msg_id, void *msgbuf, u32 msg_len, void *respbuf, u32 resp_max_len, unsigned long *resp_len)`
Send a message and wait for response NOTE: msgbuf requires little-endian byte-ordering

29.111.2.7 send_message_without_response

`int(* sbi_mpxy_channel::send_message_without_response) (struct sbi_mpxy_channel *channel, u32 msg_id, void *msgbuf, u32 msg_len)`
Send message without response

29.111.2.8 switch_eventsstate

`void(* sbi_mpxy_channel::switch_eventsstate) (u32 enable)`
Callback to enable the events state reporting in the message protocol implementation

29.111.2.9 write_attributes

`int(* sbi_mpxy_channel::write_attributes) (struct sbi_mpxy_channel *channel, u32 *inmem, u32 base_attr_id, u32 attr_count)`
Write message protocol attributes NOTE: outmem requires little-endian byte-ordering
The documentation for this struct was generated from the following file:

- include/sbi/sbi_mpxy.h

29.112 sbi_mpxy_channel_attrs Struct Reference

`#include <sbi_mpxy.h>`

Data Fields

- u32 msg_proto_id
- u32 msg_proto_version
- u32 msg_data_maxlen
- u32 msg_send_timeout
- u32 msg_completion_timeout
- u32 capability
- u32 sse_event_id
- u32 msi_control
- struct sbi_mpxy_msi_info msi_info
- u32 eventsstate_ctrl

29.112.1 Detailed Description

Channel attributes. NOTE: The sequence of attribute fields are as per the defined sequence in the attribute table in spec(or as per the enum sbi_mpxy_attr_id).

29.112.2 Field Documentation

29.112.2.1 capability

`u32 sbi_mpxy_channel_attrs::capability`

29.112.2.2 eventsstate_ctrl

`u32 sbi_mpxy_channel_attrs::eventsstate_ctrl`

29.112.2.3 msg_completion_timeout

```
u32 sbi_mpxy_channel_attrs::msg_completion_timeout
```

29.112.2.4 msg_data_maxlen

```
u32 sbi_mpxy_channel_attrs::msg_data_maxlen
```

29.112.2.5 msg_proto_id

```
u32 sbi_mpxy_channel_attrs::msg_proto_id
```

29.112.2.6 msg_proto_version

```
u32 sbi_mpxy_channel_attrs::msg_proto_version
```

29.112.2.7 msg_send_timeout

```
u32 sbi_mpxy_channel_attrs::msg_send_timeout
```

29.112.2.8 msi_control

```
u32 sbi_mpxy_channel_attrs::msi_control
```

29.112.2.9 msi_info

```
struct sbi_mpxy_msi_info sbi_mpxy_channel_attrs::msi_info
```

29.112.2.10 sse_event_id

```
u32 sbi_mpxy_channel_attrs::sse_event_id
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_mpxy.h

29.113 sbi_mpxy_msi_info Struct Reference

```
#include <sbi_mpxy.h>
```

Data Fields

- u32 msi_addr_lo
- u32 msi_addr_hi
- u32 msi_data

29.113.1 Field Documentation

29.113.1.1 msi_addr_hi

```
u32 sbi_mpxy_msi_info::msi_addr_hi
```

29.113.1.2 msi_addr_lo

```
u32 sbi_mpxy_msi_info::msi_addr_lo
```

29.113.1.3 msi_data

```
u32 sbi_mpxy_msi_info::msi_data
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_mpxy.h

29.114 sbi_platform Struct Reference

`#include <sbi_platform.h>`

Data Fields

- u32 opensbi_version
- u32 platform_version
- char name [64]
- u64 features
- u32 hart_count
- u32 hart_stack_size
- u32 heap_size
- u32 reserved
- unsigned long platform_ops_addr
- unsigned long firmware_context
- const u32 ∗ hart_index2id

29.114.1 Detailed Description

Representation of a platform

29.114.2 Field Documentation

29.114.2.1 features

`u64 sbi_platform::features`
Supported features

29.114.2.2 firmware_context

`unsigned long sbi_platform::firmware_context`
Pointer to system firmware specific context

29.114.2.3 hart_count

`u32 sbi_platform::hart_count`
Total number of HARTs

29.114.2.4 hart_index2id

`const u32* sbi_platform::hart_index2id`
HART index to HART id table
For used HART index <abc>: hart_index2id[<abc>] = some HART id For unused HART index <abc>: hart_↩index2id[<abc>] = -1U
If hart_index2id == NULL then we assume identity mapping hart_index2id[<abc>] = <abc>
We have only two restrictions:

1. HART index < sbi_platform hart_count
2. HART id < SBI_HARTMASK_MAX_BITS

29.114.2.5 hart_stack_size

`u32 sbi_platform::hart_stack_size`
Per-HART stack size for exception/interrupt handling

29.114.2.6 heap_size

`u32 sbi_platform::heap_size`
Size of heap shared by all HARTs

29.114.2.7 name

`char sbi_platform::name[64]`
Name of the platform

29.114.2.8 opensbi_version

`u32 sbi_platform::opensbi_version`
OpenSBI version this sbi_platform is based on. It's a 32-bit value where upper 16-bits are major number and lower 16-bits are minor number

29.114.2.9 platform_ops_addr

`unsigned long sbi_platform::platform_ops_addr`
Pointer to sbi platform operations

29.114.2.10 platform_version

`u32 sbi_platform::platform_version`
OpenSBI platform version released by vendor. It's a 32-bit value where upper 16-bits are major number and lower 16-bits are minor number

29.114.2.11 reserved

`u32 sbi_platform::reserved`
Reserved for future use
The documentation for this struct was generated from the following file:

- include/sbi/sbi_platform.h

29.115 sbi_platform_operations Struct Reference

`#include <sbi_platform.h>`

Data Fields

- bool(∗ cold_boot_allowed)(u32 hartid)
- int(∗ nascent_init)(void)
- int(∗ early_init)(bool cold_boot)
- int(∗ final_init)(bool cold_boot)
- void(∗ early_exit)(void)
- void(∗ final_exit)(void)
- int(∗ misa_check_extension)(char ext)
- int(∗ misa_get_xlen)(void)
- int(∗ extensions_init)(struct sbi_hart_features ∗hfeatures)
- int(∗ domains_init)(void)
- int(∗ pmu_init)(void)
- uint64_t(∗ pmu_xlate_to_mhpmevent)(uint32_t event_idx, uint64_t data)
- int(∗ irqchip_init)(void)
- int(∗ ipi_init)(void)
- u64(∗ get_tlbr_flush_limit)(void)
- u32(∗ get_tlb_num_entries)(void)
- int(∗ timer_init)(void)

- int(∗ mpxy_init)(void)
- bool(∗ vendor_ext_check)(void)
- int(∗ vendor_ext_provider)(long funcid, struct sbi_trap_regs ∗regs, struct sbi_ecall_return ∗out)
- int(∗ emulate_load)(int rlen, unsigned long addr, union sbi_ldst_data ∗out_val)
- int(∗ emulate_store)(int wlen, unsigned long addr, union sbi_ldst_data in_val)

29.115.1 Detailed Description

Platform functions

29.115.2 Field Documentation

29.115.2.1 cold_boot_allowed

```
bool(* sbi_platform_operations::cold_boot_allowed) (u32 hartid)
```

29.115.2.2 domains_init

```
int(* sbi_platform_operations::domains_init) (void)
```

Initialize (or populate) domains for the platform

29.115.2.3 early_exit

```
void(* sbi_platform_operations::early_exit) (void)
```

Platform early exit

29.115.2.4 early_init

```
int(* sbi_platform_operations::early_init) (bool cold_boot)
```

Platform early initialization

29.115.2.5 emulate_load

```
int(* sbi_platform_operations::emulate_load) (int rlen, unsigned long addr, union sbi_ldst_data
*out_val)
```

platform specific handler to fixup load fault

29.115.2.6 emulate_store

```
int(* sbi_platform_operations::emulate_store) (int wlen, unsigned long addr, union sbi_ldst_data
in_val)
```

platform specific handler to fixup store fault

29.115.2.7 extensions_init

```
int(* sbi_platform_operations::extensions_init) (struct sbi_hart_features *hfeatures)
```

Initialize (or populate) HART extensions for the platform

29.115.2.8 final_exit

```
void(* sbi_platform_operations::final_exit) (void)
```

Platform final exit

29.115.2.9 final_init

```
int(* sbi_platform_operations::final_init) (bool cold_boot)
```

Platform final initialization

29.115.2.10 get_tlb_num_entries

```
u32(* sbi_platform_operations::get_tlb_num_entries) (void)
```

Get tlb fifo num entries

29.115.2.11 get_tlbr_flush_limit

```
u64(* sbi_platform_operations::get_tlbr_flush_limit) (void)
```

Get tlb flush limit value

29.115.2.12 ipi_init

```
int(* sbi_platform_operations::ipi_init) (void)
```

Initialize IPI during cold boot

29.115.2.13 irqchip_init

```
int(* sbi_platform_operations::irqchip_init) (void)
```

Initialize the platform interrupt controller during cold boot

29.115.2.14 misa_check_extension

```
int(* sbi_platform_operations::misa_check_extension) (char ext)
```

For platforms that do not implement misa, non-standard methods are needed to determine cpu extension.

29.115.2.15 misa_get_xlen

```
int(* sbi_platform_operations::misa_get_xlen) (void)
```

For platforms that do not implement misa, non-standard methods are needed to get MXL field of misa.

29.115.2.16 mpxy_init

```
int(* sbi_platform_operations::mpxy_init) (void)
```

Initialize the platform Message Proxy(MPXY) driver

29.115.2.17 nascent_init

```
int(* sbi_platform_operations::nascent_init) (void)
```

29.115.2.18 pmu_init

```
int(* sbi_platform_operations::pmu_init) (void)
```

Initialize hw performance counters

29.115.2.19 pmu_xlate_to_mhpmevent

```
uint64_t(* sbi_platform_operations::pmu_xlate_to_mhpmevent) (uint32_t event_idx, uint64_↩
t data)
```

Get platform specific mhpmevent value

29.115.2.20 timer_init

```
int(* sbi_platform_operations::timer_init) (void)
```

Initialize platform timer during cold boot

29.115.2.21 vendor_ext_check

```
bool(* sbi_platform_operations::vendor_ext_check) (void)
```

Check if SBI vendor extension is implemented or not

29.115.2.22 vendor_ext_provider

`int(* sbi_platform_operations::vendor_ext_provider) (long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)`
platform specific SBI extension implementation provider
The documentation for this struct was generated from the following file:

- include/sbi/sbi_platform.h

29.116 sbi_pmu_device Struct Reference

`#include <sbi_pmu.h>`

Data Fields

- char name [32]
- int(* fw_event_validate_encoding)(uint32_t hartid, uint64_t event_data)
- bool(* fw_counter_match_encoding)(uint32_t hartid, uint32_t counter_index, uint64_t event_data)
- int(* fw_counter_width)(void)
- uint64_t(* fw_counter_read_value)(uint32_t hartid, uint32_t counter_index)
- void(* fw_counter_write_value)(uint32_t hartid, uint32_t counter_index, uint64_t value)
- int(* fw_counter_start)(uint32_t hartid, uint32_t counter_index, uint64_t event_data)
- int(* fw_counter_stop)(uint32_t hartid, uint32_t counter_index)
- void(* hw_counter_enable_irq)(uint32_t counter_index)
- void(* hw_counter_disable_irq)(uint32_t counter_index)
- int(* hw_counter_irq_bit)(void)
- void(* hw_counter_filter_mode)(unsigned long flags, int counter_index)

29.116.1 Field Documentation

29.116.1.1 fw_counter_match_encoding

`bool(* sbi_pmu_device::fw_counter_match_encoding) (uint32_t hartid, uint32_t counter_index, uint64_t event_data)`
Match custom firmware counter with custom firmware event Note: 0 <= counter_index < SBI_PMU_FW_CTR_↩ MAX

29.116.1.2 fw_counter_read_value

`uint64_t(* sbi_pmu_device::fw_counter_read_value) (uint32_t hartid, uint32_t counter_index)`
Read value of custom firmware counter Note: 0 <= counter_index < SBI_PMU_FW_CTR_MAX

29.116.1.3 fw_counter_start

`int(* sbi_pmu_device::fw_counter_start) (uint32_t hartid, uint32_t counter_index, uint64_↩ t event_data)`
Start custom firmware counter Note: 0 <= counter_index < SBI_PMU_FW_CTR_MAX

29.116.1.4 fw_counter_stop

`int(* sbi_pmu_device::fw_counter_stop) (uint32_t hartid, uint32_t counter_index)`
Stop custom firmware counter Note: 0 <= counter_index < SBI_PMU_FW_CTR_MAX

29.116.1.5 fw_counter_width

`int(* sbi_pmu_device::fw_counter_width) (void)`
Fetch the max width of this counter in number of bits.

29.116.1.6 fw_counter_write_value

```
void(* sbi_pmu_device::fw_counter_write_value) (uint32_t hartid, uint32_t counter_index, uint64↩
_t value)
```

Write value to custom firmware counter Note: 0 <= counter_index < SBI_PMU_FW_CTR_MAX

29.116.1.7 fw_event_validate_encoding

```
int(* sbi_pmu_device::fw_event_validate_encoding) (uint32_t hartid, uint64_t event_data)
```

Validate event code of custom firmware event

29.116.1.8 hw_counter_disable_irq

```
void(* sbi_pmu_device::hw_counter_disable_irq) (uint32_t counter_index)
```

Custom disable irq for hardware counter Note: 0 <= counter_index < SBI_PMU_HW_CTR_MAX

29.116.1.9 hw_counter_enable_irq

```
void(* sbi_pmu_device::hw_counter_enable_irq) (uint32_t counter_index)
```

Custom enable irq for hardware counter Note: 0 <= counter_index < SBI_PMU_HW_CTR_MAX

29.116.1.10 hw_counter_filter_mode

```
void(* sbi_pmu_device::hw_counter_filter_mode) (unsigned long flags, int counter_index)
```

Custom function to inhibit counting of events while in specified mode.

29.116.1.11 hw_counter_irq_bit

```
int(* sbi_pmu_device::hw_counter_irq_bit) (void)
```

Custom function returning the machine-specific irq-bit.

29.116.1.12 name

```
char sbi_pmu_device::name[32]
```

Name of the PMU platform device

The documentation for this struct was generated from the following file:

- include/sbi/sbi_pmu.h

29.117 sbi_pmu_event_info Struct Reference

```
#include <sbi_ecall_interface.h>
```

Data Fields

- uint32_t event_idx
- uint32_t output
- uint64_t event_data

29.117.1 Field Documentation

29.117.1.1 event_data

```
uint64_t sbi_pmu_event_info::event_data
```

29.117.1.2 event_idx

```
uint32_t sbi_pmu_event_info::event_idx
```

29.117.1.3 output

`uint32_t sbi_pmu_event_info::output`

The documentation for this struct was generated from the following file:

- include/sbi/sbi_ecall_interface.h

29.118 sbi_scratch Struct Reference

`#include <sbi_scratch.h>`

Data Fields

- unsigned long fw_start
- unsigned long fw_size
- unsigned long fw_rw_offset
- unsigned long fw_heap_offset
- unsigned long fw_heap_size
- unsigned long next_arg1
- unsigned long next_addr
- unsigned long next_mode
- unsigned long warmboot_addr
- unsigned long platform_addr
- unsigned long hartid_to_scratch
- unsigned long trap_context
- unsigned long tmp0
- unsigned long options
- unsigned long hartindex

29.118.1 Detailed Description

Representation of per-HART scratch space

29.118.2 Field Documentation

29.118.2.1 fw_heap_offset

`unsigned long sbi_scratch::fw_heap_offset`

Offset (in bytes) of the heap area

29.118.2.2 fw_heap_size

`unsigned long sbi_scratch::fw_heap_size`

Size (in bytes) of the heap area

29.118.2.3 fw_rw_offset

`unsigned long sbi_scratch::fw_rw_offset`

Offset (in bytes) of the R/W section

29.118.2.4 fw_size

`unsigned long sbi_scratch::fw_size`

Size (in bytes) of firmware linked to OpenSBI library

29.118.2.5 fw_start

`unsigned long sbi_scratch::fw_start`

Start (or base) address of firmware linked to OpenSBI library

29.118.2.6 hartid_to_scratch

`unsigned long sbi_scratch::hartid_to_scratch`
Address of HART ID to sbi_scratch conversion function

29.118.2.7 hartindex

`unsigned long sbi_scratch::hartindex`
Index of the hart

29.118.2.8 next_addr

`unsigned long sbi_scratch::next_addr`
Address of next booting stage for this HART

29.118.2.9 next_arg1

`unsigned long sbi_scratch::next_arg1`
Arg1 (or 'a1' register) of next booting stage for this HART

29.118.2.10 next_mode

`unsigned long sbi_scratch::next_mode`
Privilege mode of next booting stage for this HART

29.118.2.11 options

`unsigned long sbi_scratch::options`
Options for OpenSBI library

29.118.2.12 platform_addr

`unsigned long sbi_scratch::platform_addr`
Address of sbi_platform

29.118.2.13 tmp0

`unsigned long sbi_scratch::tmp0`
Temporary storage

29.118.2.14 trap_context

`unsigned long sbi_scratch::trap_context`
Address of current trap context

29.118.2.15 warmboot_addr

`unsigned long sbi_scratch::warmboot_addr`
Warm boot entry point address for this HART
The documentation for this struct was generated from the following file:

- include/sbi/sbi_scratch.h

29.119 sbi_sse_cb_ops Struct Reference

```
#include <sbi_sse.h>
```

Data Fields

- void(∗ set_hartid_cb)(uint32_t event_id, unsigned long hart_id)
- void(∗ complete_cb)(uint32_t event_id)
- void(∗ register_cb)(uint32_t event_id)
- void(∗ unregister_cb)(uint32_t event_id)
- void(∗ enable_cb)(uint32_t event_id)
- void(∗ disable_cb)(uint32_t event_id)

29.119.1 Field Documentation

29.119.1.1 complete_cb

`void(* sbi_sse_cb_ops::complete_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_COMPLETE is invoked on the event.

29.119.1.2 disable_cb

`void(* sbi_sse_cb_ops::disable_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_DISABLE is invoked on the event.

29.119.1.3 enable_cb

`void(* sbi_sse_cb_ops::enable_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_ENABLE is invoked on the event.

29.119.1.4 register_cb

`void(* sbi_sse_cb_ops::register_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_REGISTER is invoked on the event.

29.119.1.5 set_hartid_cb

`void(* sbi_sse_cb_ops::set_hartid_cb) (uint32_t event_id, unsigned long hart_id)`
Called when hart_id is changed on the event.

29.119.1.6 unregister_cb

`void(* sbi_sse_cb_ops::unregister_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_UNREGISTER is invoked on the event.
The documentation for this struct was generated from the following file:

- include/sbi/sbi_sse.h

29.120 sbi_system_reset_device Struct Reference

`#include <sbi_system.h>`

Data Fields

- char name [32]
- int(∗ system_reset_check)(u32 reset_type, u32 reset_reason)
- void(∗ system_reset)(u32 reset_type, u32 reset_reason)
- struct sbi_dlist node

29.120.1 Detailed Description

System reset hardware device

29.120.2 Field Documentation

29.120.2.1 name

```
char sbi_system_reset_device::name[32]
```

Name of the system reset device

29.120.2.2 node

```
struct sbi_dlist sbi_system_reset_device::node
```

List

29.120.2.3 system_reset

```
void(* sbi_system_reset_device::system_reset) (u32 reset_type, u32 reset_reason)
```

Reset the system

29.120.2.4 system_reset_check

```
int(* sbi_system_reset_device::system_reset_check) (u32 reset_type, u32 reset_reason)
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_system.h

29.121 sbi_system_suspend_device Struct Reference

```
#include <sbi_system.h>
```

Data Fields

- char name [32]
- int(* system_suspend_check)(u32 sleep_type)
- int(* system_suspend)(u32 sleep_type, unsigned long mmode_resume_addr)

29.121.1 Detailed Description

System suspend device

29.121.2 Field Documentation

29.121.2.1 name

```
char sbi_system_suspend_device::name[32]
```

Name of the system suspend device

29.121.2.2 system_suspend

```
int(* sbi_system_suspend_device::system_suspend) (u32 sleep_type, unsigned long mmode_resume_↩
addr)
```

Suspend the system

@sleep_type: The sleep type identifier passed to the SBI call. @mmode_resume_addr: This is the same as sbi_scratch.warmboot_addr. Some platforms may not be able to return from system_suspend(), so they will jump directly to this address instead. Platforms which can return from system_suspend() may ignore this parameter.

29.121.2.3 system_suspend_check

```
int(* sbi_system_suspend_device::system_suspend_check) (u32 sleep_type)
```

Check whether sleep type is supported by the device

Returns 0 when @sleep_type supported, SBI_ERR_INVALID_PARAM when @sleep_type is reserved, or SBI_↩ ERR_NOT_SUPPORTED when @sleep_type is not reserved and is implemented, but the platform doesn't support it due to missing dependencies.
The documentation for this struct was generated from the following file:

- include/sbi/sbi_system.h

29.122 sbi_timer_device Struct Reference

`#include <sbi_timer.h>`

Data Fields

- char name [32]
- unsigned long timer_freq
- u64(∗ timer_value)(void)
- void(∗ timer_event_start)(u64 next_event)
- void(∗ timer_event_stop)(void)
- int(∗ warm_init)(void)

29.122.1 Detailed Description

Timer hardware device

29.122.2 Field Documentation

29.122.2.1 name

`char sbi_timer_device::name[32]`
Name of the timer operations

29.122.2.2 timer_event_start

`void(* sbi_timer_device::timer_event_start) (u64 next_event)`
Start timer event for current HART

29.122.2.3 timer_event_stop

`void(* sbi_timer_device::timer_event_stop) (void)`
Stop timer event for current HART

29.122.2.4 timer_freq

`unsigned long sbi_timer_device::timer_freq`
Frequency of timer in HZ

29.122.2.5 timer_value

`u64(* sbi_timer_device::timer_value) (void)`
Get free-running timer value

29.122.2.6 warm_init

`int(* sbi_timer_device::warm_init) (void)`
Initialize timer device for current HART
The documentation for this struct was generated from the following file:

- include/sbi/sbi_timer.h

29.123 sbi_tlb_info Struct Reference

```
#include <sbi_tlb.h>
```

Data Fields

- unsigned long start
- unsigned long size
- uint16_t asid
- uint16_t vmid
- enum sbi_tlb_type type
- struct sbi_hartmask smask

29.123.1 Field Documentation

29.123.1.1 asid

```
uint16_t sbi_tlb_info::asid
```

29.123.1.2 size

```
unsigned long sbi_tlb_info::size
```

29.123.1.3 smask

```
struct sbi_hartmask sbi_tlb_info::smask
```

29.123.1.4 start

```
unsigned long sbi_tlb_info::start
```

29.123.1.5 type

```
enum sbi_tlb_type sbi_tlb_info::type
```

29.123.1.6 vmid

```
uint16_t sbi_tlb_info::vmid
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_tlb.h

29.124 sbi_trap_context Struct Reference

```
#include <sbi_trap.h>
```

Data Fields

- struct sbi_trap_regs regs
- struct sbi_trap_info trap
- struct sbi_trap_context * prev_context

29.124.1 Detailed Description

Representation of trap context saved on stack

29.124.2 Field Documentation

29.124.2.1 prev_context

`struct sbi_trap_context* sbi_trap_context::prev_context`
Pointer to previous trap context

29.124.2.2 regs

`struct sbi_trap_regs sbi_trap_context::regs`
Register state

29.124.2.3 trap

`struct sbi_trap_info sbi_trap_context::trap`
Trap details
The documentation for this struct was generated from the following file:

- include/sbi/sbi_trap.h

29.125 sbi_trap_info Struct Reference

`#include <sbi_trap.h>`

Data Fields

- unsigned long cause
- unsigned long tval
- unsigned long tval2
- unsigned long tinst
- unsigned long gva

29.125.1 Detailed Description

Representation of trap details

29.125.2 Field Documentation

29.125.2.1 cause

`unsigned long sbi_trap_info::cause`
cause Trap exception cause

29.125.2.2 gva

`unsigned long sbi_trap_info::gva`
gva Guest virtual address in tval flag

29.125.2.3 tinst

`unsigned long sbi_trap_info::tinst`
tinst Trap instruction

29.125.2.4 tval

`unsigned long sbi_trap_info::tval`
tval Trap value

29.125.2.5 tval2

`unsigned long sbi_trap_info::tval2`
tval2 Trap value 2
The documentation for this struct was generated from the following file:

- include/sbi/sbi_trap.h

29.126 sbi_trap_regs Struct Reference

```
#include <sbi_trap.h>
```

Data Fields

- unsigned long zero
- unsigned long ra
- unsigned long sp
- unsigned long gp
- unsigned long tp
- unsigned long t0
- unsigned long t1
- unsigned long t2
- unsigned long s0
- unsigned long s1
- unsigned long a0
- unsigned long a1
- unsigned long a2
- unsigned long a3
- unsigned long a4
- unsigned long a5
- unsigned long a6
- unsigned long a7
- unsigned long s2
- unsigned long s3
- unsigned long s4
- unsigned long s5
- unsigned long s6
- unsigned long s7
- unsigned long s8
- unsigned long s9
- unsigned long s10
- unsigned long s11
- unsigned long t3
- unsigned long t4
- unsigned long t5
- unsigned long t6
- unsigned long mepc
- unsigned long mstatus
- unsigned long mstatusH

29.126.1 Detailed Description

Representation of register state at time of trap/interrupt

29.126.2 Field Documentation

29.126.2.1 a0

`unsigned long sbi_trap_regs::a0`
a0 register state

29.126.2.2 a1

`unsigned long sbi_trap_regs::a1`
a1 register state

29.126.2.3 a2

`unsigned long sbi_trap_regs::a2`
a2 register state

29.126.2.4 a3

`unsigned long sbi_trap_regs::a3`
a3 register state

29.126.2.5 a4

`unsigned long sbi_trap_regs::a4`
a4 register state

29.126.2.6 a5

`unsigned long sbi_trap_regs::a5`
a5 register state

29.126.2.7 a6

`unsigned long sbi_trap_regs::a6`
a6 register state

29.126.2.8 a7

`unsigned long sbi_trap_regs::a7`
a7 register state

29.126.2.9 gp

`unsigned long sbi_trap_regs::gp`
gp register state

29.126.2.10 mepc

`unsigned long sbi_trap_regs::mepc`
mepc register state

29.126.2.11 mstatus

`unsigned long sbi_trap_regs::mstatus`
mstatus register state

29.126.2.12 mstatusH

`unsigned long sbi_trap_regs::mstatusH`
mstatusH register state (only for 32-bit)

29.126.2.13 ra

`unsigned long sbi_trap_regs::ra`
ra register state

29.126.2.14 s0

`unsigned long sbi_trap_regs::s0`
s0 register state

29.126.2.15 s1

`unsigned long sbi_trap_regs::s1`
s1 register state

29.126.2.16 s10

`unsigned long sbi_trap_regs::s10`
s10 register state

29.126.2.17 s11

`unsigned long sbi_trap_regs::s11`
s11 register state

29.126.2.18 s2

`unsigned long sbi_trap_regs::s2`
s2 register state

29.126.2.19 s3

`unsigned long sbi_trap_regs::s3`
s3 register state

29.126.2.20 s4

`unsigned long sbi_trap_regs::s4`
s4 register state

29.126.2.21 s5

`unsigned long sbi_trap_regs::s5`
s5 register state

29.126.2.22 s6

`unsigned long sbi_trap_regs::s6`
s6 register state

29.126.2.23 s7

`unsigned long sbi_trap_regs::s7`
s7 register state

29.126.2.24 s8

`unsigned long sbi_trap_regs::s8`
s8 register state

29.126.2.25 s9

`unsigned long sbi_trap_regs::s9`
s9 register state

29.126.2.26 sp

`unsigned long sbi_trap_regs::sp`
sp register state

29.126.2.27 t0

`unsigned long sbi_trap_regs::t0`
t0 register state

29.126.2.28 t1

`unsigned long sbi_trap_regs::t1`
t1 register state

29.126.2.29 t2

`unsigned long sbi_trap_regs::t2`
t2 register state

29.126.2.30 t3

`unsigned long sbi_trap_regs::t3`
t3 register state

29.126.2.31 t4

`unsigned long sbi_trap_regs::t4`
t4 register state

29.126.2.32 t5

`unsigned long sbi_trap_regs::t5`
t5 register state

29.126.2.33 t6

`unsigned long sbi_trap_regs::t6`
t6 register state

29.126.2.34 tp

`unsigned long sbi_trap_regs::tp`
tp register state

29.126.2.35 zero

`unsigned long sbi_trap_regs::zero`
zero register state
The documentation for this struct was generated from the following file:

- include/sbi/sbi_trap.h

29.127 smu_data Struct Reference

```
#include <atcsmu.h>
```

Data Fields

- unsigned long addr

29.127.1 Field Documentation

29.127.1.1 addr

`unsigned long smu_data::addr`

The documentation for this struct was generated from the following file:

- include/sbi_utils/sys/atcsmu.h

29.128 uu Union Reference

`#include <quad.h>`

Data Fields

- quad_t q
- quad_t uq
- long sl [2]
- u_long ul [2]

29.128.1 Field Documentation

29.128.1.1 q

`quad_t uu::q`

29.128.1.2 sl

`long uu::sl[2]`

29.128.1.3 ul

`u_long uu::ul[2]`

29.128.1.4 uq

`quad_t uu::uq`

The documentation for this union was generated from the following file:

- lib/utils/libquad/quad.h

Chapter 30

File Documentation

30.1 docs/contributing.md File Reference

30.2 docs/domain_support.md File Reference

30.3 docs/firmware/fw.md File Reference

30.4 docs/firmware/fw_dynamic.md File Reference

30.5 docs/firmware/fw_jump.md File Reference

30.6 docs/firmware/fw_payload.md File Reference

30.7 docs/firmware/payload_linux.md File Reference

30.8 docs/firmware/payload_uboot.md File Reference

30.9 docs/library_usage.md File Reference

30.10 docs/opensbi_config.md File Reference

30.11 docs/platform/andes-ae350.md File Reference

30.12 docs/platform/fpga-ariane.md File Reference

30.13 docs/platform/fpga-openpiton.md File Reference

30.14 docs/platform/generic.md File Reference

30.15 docs/platform/nuclei_ux600.md File Reference

30.16 docs/platform/platform.md File Reference

30.17 docs/platform/qemu_virt.md File Reference

30.18 docs/platform/renesas-rzfive.md File Reference

30.19 docs/platform/shakti_cclass.md File Reference

30.20 docs/platform/sifive_fu540.md File Reference

30.21 docs/platform/spike.md File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fw_dynamic.h:

Data Structures

- struct fw_dynamic_info

Macros

- #define FW_DYNAMIC_INFO_MAGIC_OFFSET (0 ∗ __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_VERSION_OFFSET (1 ∗ __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET (2 ∗ __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_NEXT_MODE_OFFSET (3 ∗ __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_OPTIONS_OFFSET (4 ∗ __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_BOOT_HART_OFFSET (5 ∗ __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_MAGIC_VALUE 0x4942534f
- #define FW_DYNAMIC_INFO_VERSION_2 0x2
- #define FW_DYNAMIC_INFO_VERSION_MAX FW_DYNAMIC_INFO_VERSION_2
- #define FW_DYNAMIC_INFO_NEXT_MODE_U 0x0
- #define FW_DYNAMIC_INFO_NEXT_MODE_S 0x1
- #define FW_DYNAMIC_INFO_NEXT_MODE_M 0x3

Variables

- struct fw_dynamic_info __packed

30.26.1 Macro Definition Documentation

30.26.1.1 FW_DYNAMIC_INFO_BOOT_HART_OFFSET

```
#define FW_DYNAMIC_INFO_BOOT_HART_OFFSET (5 * __SIZEOF_LONG__)
```

Offset of boot_hart member in fw_dynamic_info (version >= 2)

30.26.1.2 FW_DYNAMIC_INFO_MAGIC_OFFSET

```
#define FW_DYNAMIC_INFO_MAGIC_OFFSET (0 * __SIZEOF_LONG__)
```

Offset of magic member in fw_dynamic_info

30.26.1.3 FW_DYNAMIC_INFO_MAGIC_VALUE

```
#define FW_DYNAMIC_INFO_MAGIC_VALUE 0x4942534f
```

Expected value of info magic ('OSBI' ascii string in hex)

30.26.1.4 FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET

```
#define FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET (2 * __SIZEOF_LONG__)
```

Offset of next_addr member in fw_dynamic_info (version >= 1)

30.26.1.5 FW_DYNAMIC_INFO_NEXT_MODE_M

```
#define FW_DYNAMIC_INFO_NEXT_MODE_M 0x3
```

30.26.1.6 FW_DYNAMIC_INFO_NEXT_MODE_OFFSET

```
#define FW_DYNAMIC_INFO_NEXT_MODE_OFFSET (3 * __SIZEOF_LONG__)
```

Offset of next_mode member in fw_dynamic_info (version >= 1)

30.26.1.7 FW_DYNAMIC_INFO_NEXT_MODE_S

```
#define FW_DYNAMIC_INFO_NEXT_MODE_S 0x1
```

30.26.1.8 FW_DYNAMIC_INFO_NEXT_MODE_U

```
#define FW_DYNAMIC_INFO_NEXT_MODE_U 0x0
```

Possible next mode values

30.26.1.9 FW_DYNAMIC_INFO_OPTIONS_OFFSET

```
#define FW_DYNAMIC_INFO_OPTIONS_OFFSET (4 * __SIZEOF_LONG__)
```

Offset of options member in fw_dynamic_info (version >= 1)

30.26.1.10 FW_DYNAMIC_INFO_VERSION_2

```
#define FW_DYNAMIC_INFO_VERSION_2 0x2
```

Maximum supported info version

30.26.1.11 FW_DYNAMIC_INFO_VERSION_MAX

```
#define FW_DYNAMIC_INFO_VERSION_MAX FW_DYNAMIC_INFO_VERSION_2
```

30.26.1.12 FW_DYNAMIC_INFO_VERSION_OFFSET

```
#define FW_DYNAMIC_INFO_VERSION_OFFSET (1 * __SIZEOF_LONG__)
```

Offset of version member in fw_dynamic_info

30.26.2 Variable Documentation

30.26.2.1 __packed

```
struct fw_dynamic_info __packed
```

30.27 fw_dynamic.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __FW_DYNAMIC_H__
#define __FW_DYNAMIC_H__

#include <sbi/riscv_asm.h>

/* clang-format off */

#define FW_DYNAMIC_INFO_MAGIC_OFFSET        (0 * __SIZEOF_LONG__)
#define FW_DYNAMIC_INFO_VERSION_OFFSET      (1 * __SIZEOF_LONG__)
#define FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET    (2 * __SIZEOF_LONG__)
#define FW_DYNAMIC_INFO_NEXT_MODE_OFFSET    (3 * __SIZEOF_LONG__)
#define FW_DYNAMIC_INFO_OPTIONS_OFFSET      (4 * __SIZEOF_LONG__)
#define FW_DYNAMIC_INFO_BOOT_HART_OFFSET    (5 * __SIZEOF_LONG__)

#define FW_DYNAMIC_INFO_MAGIC_VALUE     0x4942534f

#define FW_DYNAMIC_INFO_VERSION_2       0x2
#define FW_DYNAMIC_INFO_VERSION_MAX     FW_DYNAMIC_INFO_VERSION_2

#define FW_DYNAMIC_INFO_NEXT_MODE_U     0x0
#define FW_DYNAMIC_INFO_NEXT_MODE_S     0x1
#define FW_DYNAMIC_INFO_NEXT_MODE_M     0x3

/* clang-format on */

#ifndef __ASSEMBLER__

#include <sbi/sbi_types.h>

struct fw_dynamic_info {
    unsigned long magic;
    unsigned long version;
    unsigned long next_addr;
    unsigned long next_mode;
    unsigned long options;
    unsigned long boot_hart;
} __packed;

_Static_assert(
    offsetof(struct fw_dynamic_info, magic)
        == FW_DYNAMIC_INFO_MAGIC_OFFSET,
    "struct fw_dynamic_info definition has changed, please redefine "
    "FW_DYNAMIC_INFO_MAGIC_OFFSET");
_Static_assert(
    offsetof(struct fw_dynamic_info, version)
        == FW_DYNAMIC_INFO_VERSION_OFFSET,
    "struct fw_dynamic_info definition has changed, please redefine "
    "FW_DYNAMIC_INFO_VERSION_OFFSET");
_Static_assert(
    offsetof(struct fw_dynamic_info, next_addr)
        == FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET,
    "struct fw_dynamic_info definition has changed, please redefine "
    "FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET");
_Static_assert(
    offsetof(struct fw_dynamic_info, next_mode)
        == FW_DYNAMIC_INFO_NEXT_MODE_OFFSET,
    "struct fw_dynamic_info definition has changed, please redefine "
    "FW_DYNAMIC_INFO_NEXT_MODE_OFFSET");
_Static_assert(
    offsetof(struct fw_dynamic_info, options)
        == FW_DYNAMIC_INFO_OPTIONS_OFFSET,
    "struct fw_dynamic_info definition has changed, please redefine "
    "FW_DYNAMIC_INFO_OPTIONS_OFFSET");
_Static_assert(
    offsetof(struct fw_dynamic_info, boot_hart)
        == FW_DYNAMIC_INFO_BOOT_HART_OFFSET,
    "struct fw_dynamic_info definition has changed, please redefine "
    "FW_DYNAMIC_INFO_BOOT_HART_OFFSET");

#endif

#endif
```

30.28 include/sbi/riscv_asm.h File Reference

`#include <sbi/riscv_encoding.h>`
Include dependency graph for riscv_asm.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define __ASM_STR(x) #x
- #define PAGE_SHIFT (12)
- #define PAGE_SIZE (_AC(1, UL) << PAGE_SHIFT)
- #define PAGE_MASK (~(PAGE_SIZE - 1))
- #define REG_L __REG_SEL(ld, lw)
- #define REG_S __REG_SEL(sd, sw)
- #define SZREG __REG_SEL(8, 4)
- #define LGREG __REG_SEL(3, 2)
- #define csr_swap(csr, val)
- #define csr_read(csr)
- #define csr_read_relaxed(csr)
- #define csr_write(csr, val)
- #define csr_read_set(csr, val)
- #define csr_set(csr, val)
- #define csr_read_clear(csr, val)
- #define csr_clear(csr, val)
- #define wfi()
- #define ebreak()
- #define current_hartid() ((unsigned int)csr_read_relaxed(CSR_MHARTID))
- #define misa_extension(c)

Functions

- unsigned long csr_read_num (int csr_num)
- void csr_write_num (int csr_num, unsigned long val)
- int misa_extension_imp (char ext)
- int misa_xlen (void)
- void misa_string (int xlen, char *out, unsigned int out_sz)
- int pmp_disable (unsigned int n)
- int is_pmp_entry_mapped (unsigned long entry)
- int pmp_set (unsigned int n, unsigned long prot, unsigned long addr, unsigned long log2len)
- int pmp_get (unsigned int n, unsigned long *prot_out, unsigned long *addr_out, unsigned long *log2len)

30.28.1 Macro Definition Documentation

30.28.1.1 __ASM_STR

```
#define __ASM_STR(
            x ) #x
```

30.28.1.2 csr_clear

```
#define csr_clear(
            csr,
            val )
```

Value:

```
({                                                          \
    unsigned long __v = (unsigned long)(val);               \
    __asm__ __volatile__("csrc " __ASM_STR(csr) ", %0" \
                 :                                          \
                 : "rK"(__v)                                \
                 : "memory");                               \
})
```

30.28.1.3 csr_read

```
#define csr_read(
            csr )
```

Value:

```
({                                                          \
    register unsigned long __v;                             \
    __asm__ __volatile__("csrr %0, " __ASM_STR(csr) \
                 : "=r"(__v)                                \
                 :                                          \
                 : "memory");                               \
    __v;                                                    \
})
```

30.28.1.4 csr_read_clear

```
#define csr_read_clear(
            csr,
            val )
```

Value:

```
({                                                                  \
    unsigned long __v = (unsigned long)(val);                       \
    __asm__ __volatile__("csrrc %0, " __ASM_STR(csr) ", %1" \
                 : "=r"(__v)                                        \
                 : "rK"(__v)                                        \
                 : "memory");                                       \
    __v;                                                            \
})
```

30.28.1.5 csr_read_relaxed

```
#define csr_read_relaxed(
            csr )
```

Value:

```
({                                                                  \
    register unsigned long __v;                                     \
    __asm__ ("csrr %0, " __ASM_STR(csr) : "=r"(__v)); \
    __v;                                                            \
})
```

30.28.1.6 csr_read_set

```
#define csr_read_set(
            csr,
            val )
```

Value:

```
({                                                                  \
    unsigned long __v = (unsigned long)(val);                       \
    __asm__ __volatile__("csrrs %0, " __ASM_STR(csr) ", %1" \
                : "=r"(__v)                                         \
                : "rK"(__v)                                         \
                : "memory");                                        \
    __v;                                                            \
})
```

30.28.1.7 csr_set

```
#define csr_set(
            csr,
            val )
```

Value:

```
({                                                                  \
    unsigned long __v = (unsigned long)(val);                       \
    __asm__ __volatile__("csrs " __ASM_STR(csr) ", %0" \
                :                                                   \
                : "rK"(__v)                                         \
                : "memory");                                        \
})
```

30.28.1.8 csr_swap

```
#define csr_swap(
            csr,
            val )
```

Value:

```
({                                                                  \
    unsigned long __v = (unsigned long)(val);                       \
    __asm__ __volatile__("csrrw %0, " __ASM_STR(csr) ", %1" \
                : "=r"(__v)                                         \
                : "rK"(__v)                                         \
                : "memory");                                        \
    __v;                                                            \
})
```

30.28.1.9 csr_write

```
#define csr_write(
            csr,
            val )
```

Value:

```
({                                                                  \
    unsigned long __v = (unsigned long)(val);                       \
    __asm__ __volatile__("csrw " __ASM_STR(csr) ", %0" \
                :                                                   \
                : "rK"(__v)                                         \
                : "memory");                                        \
})
```

30.28.1.10 current_hartid

```
#define current_hartid( ) ((unsigned int)csr_read_relaxed(CSR_MHARTID))
```

30.28.1.11 ebreak

```
#define ebreak( )
```

Value:

```
do {                                                    \
    __asm__ __volatile__("ebreak" ::: "memory"); \
} while (0)
```

30.28.1.12 LGREG

```
#define LGREG __REG_SEL(3, 2)
```

30.28.1.13 misa_extension

```
#define misa_extension(
            c )
```

Value:

```
({\
    _Static_assert(((c >= 'A') && (c <= 'Z')),\
        "The parameter of misa_extension must be [A-Z]");\
    misa_extension_imp(c);\
})
```

30.28.1.14 PAGE_MASK

```
#define PAGE_MASK (~(PAGE_SIZE - 1))
```

30.28.1.15 PAGE_SHIFT

```
#define PAGE_SHIFT (12)
```

30.28.1.16 PAGE_SIZE

```
#define PAGE_SIZE (_AC(1, UL) << PAGE_SHIFT)
```

30.28.1.17 REG_L

```
#define REG_L __REG_SEL(ld, lw)
```

30.28.1.18 REG_S

```
#define REG_S __REG_SEL(sd, sw)
```

30.28.1.19 SZREG

```
#define SZREG __REG_SEL(8, 4)
```

30.28.1.20 wfi

```
#define wfi( )
```

Value:

```
do {                                                    \
    __asm__ __volatile__("wfi" ::: "memory"); \
} while (0)
```

30.28.2 Function Documentation

30.28.2.1 csr_read_num()

```
unsigned long csr_read_num (
            int csr_num )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.28.2.2 csr_write_num()

```
void csr_write_num (
            int csr_num,
            unsigned long val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.28.2.3 is_pmp_entry_mapped()

```
int is_pmp_entry_mapped (
            unsigned long entry )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.28.2.4 misa_extension_imp()

```
int misa_extension_imp (
            char ext )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.28.2.5 misa_string()

```
void misa_string (
            int xlen,
            char * out,
            unsigned int out_sz )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.28.2.6 misa_xlen()

```
int misa_xlen (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.28.2.7 pmp_disable()

```
int pmp_disable (
            unsigned int n )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.28.2.8 pmp_get()

```
int pmp_get (
            unsigned int n,
            unsigned long * prot_out,
            unsigned long * addr_out,
            unsigned long * log2len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.28.2.9 pmp_set()

```
int pmp_set (
            unsigned int n,
            unsigned long prot,
            unsigned long addr,
            unsigned long log2len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.29 riscv_asm.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __RISCV_ASM_H__
#define __RISCV_ASM_H__

#include <sbi/riscv_encoding.h>

/* clang-format off */

#ifdef __ASSEMBLER__
#define __ASM_STR(x)    x
#else
#define __ASM_STR(x)    #x
#endif

#if __riscv_xlen == 64
#define __REG_SEL(a, b) __ASM_STR(a)
#elif __riscv_xlen == 32
#define __REG_SEL(a, b) __ASM_STR(b)
#else
#error "Unexpected __riscv_xlen"
#endif

#define PAGE_SHIFT  (12)
#define PAGE_SIZE   (_AC(1, UL) << PAGE_SHIFT)
#define PAGE_MASK   (~(PAGE_SIZE - 1))

#define REG_L       __REG_SEL(ld, lw)
#define REG_S       __REG_SEL(sd, sw)
#define SZREG       __REG_SEL(8, 4)
#define LGREG       __REG_SEL(3, 2)

#if __SIZEOF_POINTER__ == 8
#ifdef __ASSEMBLER__
#define RISCV_PTR       .dword
#define RISCV_SZPTR     8
#define RISCV_LGPTR     3
#else
#define RISCV_PTR       ".dword"
#define RISCV_SZPTR     "8"
#define RISCV_LGPTR     "3"
#endif
#elif __SIZEOF_POINTER__ == 4
#ifdef __ASSEMBLER__
#define RISCV_PTR       .word
#define RISCV_SZPTR     4
#define RISCV_LGPTR     2
#else
#define RISCV_PTR       ".word"
#define RISCV_SZPTR     "4"
#define RISCV_LGPTR     "2"
#endif
#else
#error "Unexpected __SIZEOF_POINTER__"
#endif

#if (__SIZEOF_INT__ == 4)
#define RISCV_INT       __ASM_STR(.word)
#define RISCV_SZINT     __ASM_STR(4)
#define RISCV_LGINT     __ASM_STR(2)
#else
#error "Unexpected __SIZEOF_INT__"
#endif

#if (__SIZEOF_SHORT__ == 2)
#define RISCV_SHORT     __ASM_STR(.half)
#define RISCV_SZSHORT       __ASM_STR(2)
#define RISCV_LGSHORT       __ASM_STR(1)
#else
#error "Unexpected __SIZEOF_SHORT__"
#endif

/* clang-format on */

#ifndef __ASSEMBLER__

```

```
#define csr_swap(csr, val)                                                  \
    ({                                                                      \
        unsigned long __v = (unsigned long)(val);                           \
        __asm__ __volatile__("csrrw %0, " __ASM_STR(csr) ", %1" \
                    : "=r"(__v)                                             \
                    : "rK"(__v)                                             \
                    : "memory");                                            \
        __v;                                                                \
    })

#define csr_read(csr)                                                       \
    ({                                                                      \
        register unsigned long __v;                                         \
        __asm__ __volatile__("csrr %0, " __ASM_STR(csr) \
                    : "=r"(__v)                                             \
                    :                                                       \
                    : "memory");                                            \
        __v;                                                                \
    })

/* Variant of csr_read() that allows the compiler to cache the value. */
#define csr_read_relaxed(csr)                                               \
    ({                                                                      \
        register unsigned long __v;                                         \
        __asm__ ("csrr %0, " __ASM_STR(csr) : "=r"(__v)); \
        __v;                                                                \
    })

#define csr_write(csr, val)                                                 \
    ({                                                                      \
        unsigned long __v = (unsigned long)(val);                           \
        __asm__ __volatile__("csrw " __ASM_STR(csr) ", %0" \
                    :                                                       \
                    : "rK"(__v)                                             \
                    : "memory");                                            \
    })

#define csr_read_set(csr, val)                                              \
    ({                                                                      \
        unsigned long __v = (unsigned long)(val);                           \
        __asm__ __volatile__("csrrs %0, " __ASM_STR(csr) ", %1" \
                    : "=r"(__v)                                             \
                    : "rK"(__v)                                             \
                    : "memory");                                            \
        __v;                                                                \
    })

#define csr_set(csr, val)                                                   \
    ({                                                                      \
        unsigned long __v = (unsigned long)(val);                           \
        __asm__ __volatile__("csrs " __ASM_STR(csr) ", %0" \
                    :                                                       \
                    : "rK"(__v)                                             \
                    : "memory");                                            \
    })

#define csr_read_clear(csr, val)                                            \
    ({                                                                      \
        unsigned long __v = (unsigned long)(val);                           \
        __asm__ __volatile__("csrrc %0, " __ASM_STR(csr) ", %1" \
                    : "=r"(__v)                                             \
                    : "rK"(__v)                                             \
                    : "memory");                                            \
        __v;                                                                \
    })

#define csr_clear(csr, val)                                                 \
    ({                                                                      \
        unsigned long __v = (unsigned long)(val);                           \
        __asm__ __volatile__("csrc " __ASM_STR(csr) ", %0" \
                    :                                                       \
                    : "rK"(__v)                                             \
                    : "memory");                                            \
    })

unsigned long csr_read_num(int csr_num);

void csr_write_num(int csr_num, unsigned long val);

#define wfi()                                                               \
    do {                                                                    \
        __asm__ __volatile__("wfi" ::: "memory"); \
    } while (0)

#define ebreak()                                                            \
    do {                                                                    \
        __asm__ __volatile__("ebreak" ::: "memory"); \
```

```
    } while (0)

/* Get current HART id */
#define current_hartid()    ((unsigned int)csr_read_relaxed(CSR_MHARTID))

/* determine CPU extension, return non-zero support */
int misa_extension_imp(char ext);

#define misa_extension(c)\
({\
    _Static_assert(((c >= 'A') && (c <= 'Z')),\
        "The parameter of misa_extension must be [A-Z]");\
    misa_extension_imp(c);\
})

/* Get MXL field of misa, return -1 on error */
int misa_xlen(void);

/* Get RISC-V ISA string representation */
void misa_string(int xlen, char *out, unsigned int out_sz);

/* Disable pmp entry at a given index */
int pmp_disable(unsigned int n);

/* Check if the matching field is set */
int is_pmp_entry_mapped(unsigned long entry);

int pmp_set(unsigned int n, unsigned long prot, unsigned long addr,
        unsigned long log2len);

int pmp_get(unsigned int n, unsigned long *prot_out, unsigned long *addr_out,
        unsigned long *log2len);

#endif /* !__ASSEMBLER__ */

#endif
```

30.30 include/sbi/riscv_atomic.h File Reference

This graph shows which files directly or indirectly include this file:

Data Structures

- struct atomic_t

Macros

- #define ATOMIC_INIT(_lptr, val) (_lptr)->counter = (val)
- #define ATOMIC_INITIALIZER(val)

Functions

- long atomic_read (atomic_t *atom)
- void atomic_write (atomic_t *atom, long value)
- long atomic_add_return (atomic_t *atom, long value)
- long atomic_sub_return (atomic_t *atom, long value)
- long atomic_cmpxchg (atomic_t *atom, long oldval, long newval)
- long atomic_xchg (atomic_t *atom, long newval)
- unsigned int atomic_raw_xchg_uint (volatile unsigned int *ptr, unsigned int newval)
- unsigned long atomic_raw_xchg_ulong (volatile unsigned long *ptr, unsigned long newval)
- int atomic_set_bit (int nr, atomic_t *atom)
- int atomic_clear_bit (int nr, atomic_t *atom)

- int atomic_raw_set_bit (int nr, volatile unsigned long *addr)
- int atomic_raw_clear_bit (int nr, volatile unsigned long *addr)

30.30.1 Macro Definition Documentation

30.30.1.1 ATOMIC_INIT

```
#define ATOMIC_INIT(
            _lptr,
            val ) (_lptr)->counter = (val)
```

30.30.1.2 ATOMIC_INITIALIZER

```
#define ATOMIC_INITIALIZER(
            val )
```

Value:

```
{                          \
    .counter = (val), \
}
```

30.30.2 Function Documentation

30.30.2.1 atomic_add_return()

```
long atomic_add_return (
            atomic_t * atom,
            long value )
```

Here is the caller graph for this function:

30.30.2.2 atomic_clear_bit()

```
int atomic_clear_bit (
            int nr,
            atomic_t * atom )
```

Clear a bit in an atomic variable and return the value of bit before modify. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:

Here is the caller graph for this function:

30.30.2.3 atomic_cmpxchg()

```
long atomic_cmpxchg (
            atomic_t * atom,
            long oldval,
            long newval )
```

Here is the caller graph for this function:

30.30.2.4 atomic_raw_clear_bit()

```
int atomic_raw_clear_bit (
            int nr,
            volatile unsigned long * addr )
```

Clear a bit in any address and return the value of bit before modify. @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:

30.30.2.5 atomic_raw_set_bit()

```
int atomic_raw_set_bit (
            int nr,
            volatile unsigned long * addr )
```

Set a bit in any address and return the value of bit before modify. @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:

30.30.2.6 atomic_raw_xchg_uint()

```
unsigned int atomic_raw_xchg_uint (
            volatile unsigned int * ptr,
            unsigned int newval )
```

30.30.2.7 atomic_raw_xchg_ulong()

```
unsigned long atomic_raw_xchg_ulong (
            volatile unsigned long * ptr,
            unsigned long newval )
```

Here is the caller graph for this function:

30.30.2.8 atomic_read()

```
long atomic_read (
            atomic_t * atom )
```

Here is the caller graph for this function:

30.30.2.9 atomic_set_bit()

```
int atomic_set_bit (
            int nr,
            atomic_t * atom )
```

Set a bit in an atomic variable and return the value of bit before modify. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:

Here is the caller graph for this function:

30.30.2.10 atomic_sub_return()

```
long atomic_sub_return (
            atomic_t * atom,
            long value )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.30.2.11 atomic_write()

```
void atomic_write (
            atomic_t * atom,
            long value )
```

Here is the caller graph for this function:

30.30.2.12 atomic_xchg()

```
long atomic_xchg (
            atomic_t * atom,
            long newval )
```

Here is the caller graph for this function:

30.31 riscv_atomic.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __RISCV_ATOMIC_H__
#define __RISCV_ATOMIC_H__

typedef struct {
    volatile long counter;
} atomic_t;

#define ATOMIC_INIT(_lptr, val) (_lptr)->counter = (val)

#define ATOMIC_INITIALIZER(val)   \
    {                             \
        .counter = (val), \
    }

long atomic_read(atomic_t *atom);

void atomic_write(atomic_t *atom, long value);

long atomic_add_return(atomic_t *atom, long value);

long atomic_sub_return(atomic_t *atom, long value);

long atomic_cmpxchg(atomic_t *atom, long oldval, long newval);

long atomic_xchg(atomic_t *atom, long newval);

unsigned int atomic_raw_xchg_uint(volatile unsigned int *ptr,
                  unsigned int newval);

unsigned long atomic_raw_xchg_ulong(volatile unsigned long *ptr,
                    unsigned long newval);
int atomic_set_bit(int nr, atomic_t *atom);

int atomic_clear_bit(int nr, atomic_t *atom);

int atomic_raw_set_bit(int nr, volatile unsigned long *addr);

int atomic_raw_clear_bit(int nr, volatile unsigned long *addr);

#endif
```

30.32 include/sbi/riscv_barrier.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define RISCV_ACQUIRE_BARRIER "\tfence r , rw\n"
- #define RISCV_RELEASE_BARRIER "\tfence rw, w\n"
- #define RISCV_FENCE(p, s) __asm__ __volatile__ ("fence " #p "," #s : : : "memory")
- #define RISCV_FENCE_I __asm__ __volatile__ ("fence.i" : : : "memory")
- #define mb() RISCV_FENCE(iorw,iorw)
- #define rmb() RISCV_FENCE(ir,ir)
- #define wmb() RISCV_FENCE(ow,ow)
- #define smp_mb() RISCV_FENCE(rw,rw)
- #define smp_rmb() RISCV_FENCE(r,r)
- #define smp_wmb() RISCV_FENCE(w,w)
- #define cpu_relax()
- #define __smp_store_release(p, v)
- #define __smp_load_acquire(p)

30.32.1 Macro Definition Documentation

30.32.1.1 __smp_load_acquire

```
#define __smp_load_acquire(
            p )
```

Value:

```
({                                  \
    typeof(*p) ___p1 = *(p); \
    RISCV_FENCE(r, rw);      \
    ___p1;                   \
})
```

30.32.1.2 __smp_store_release

```
#define __smp_store_release(
            p,
            v )
```

Value:

```
do {                        \
    RISCV_FENCE(rw, w); \
    *(p) = (v);         \
} while (0)
```

30.32.1.3 cpu_relax

```
#define cpu_relax( )
```

Value:

```
do {                                \
    unsigned long __t;                  \
    __asm__ __volatile__ ("div %0, %0, zero" : "=r" (__t)); \
} while (0)
```

30.32.1.4 mb

```
#define mb( ) RISCV_FENCE(iorw,iorw)
```

30.32.1.5 RISCV_ACQUIRE_BARRIER

```
#define RISCV_ACQUIRE_BARRIER "\tfence r , rw\n"
```

30.32.1.6 RISCV_FENCE

```
#define RISCV_FENCE(
            p,
            s )  __asm__ __volatile__ ("fence " #p "," #s : : : "memory")
```

30.32.1.7 RISCV_FENCE_I

```
#define RISCV_FENCE_I  __asm__ __volatile__ ("fence.i" : : : "memory")
```

30.32.1.8 RISCV_RELEASE_BARRIER

```
#define RISCV_RELEASE_BARRIER "\tfence rw, w\n"
```

30.32.1.9 rmb

```
#define rmb( ) RISCV_FENCE(ir,ir)
```

30.32.1.10 smp_mb

```
#define smp_mb( ) RISCV_FENCE(rw,rw)
```

30.32.1.11 smp_rmb

```
#define smp_rmb( ) RISCV_FENCE(r,r)
```

30.32.1.12 smp_wmb

```
#define smp_wmb( ) RISCV_FENCE(w,w)
```

30.32.1.13 wmb

```
#define wmb( ) RISCV_FENCE(ow,ow)
```

30.33 riscv_barrier.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __RISCV_BARRIER_H__
#define __RISCV_BARRIER_H__

/* clang-format off */

#define RISCV_ACQUIRE_BARRIER       "\tfence r , rw\n"
#define RISCV_RELEASE_BARRIER       "\tfence rw,  w\n"

#define RISCV_FENCE(p, s) \
    __asm__ __volatile__ ("fence " #p "," #s : : : "memory")

#define RISCV_FENCE_I \
    __asm__ __volatile__ ("fence.i" : : : "memory")

/* Read & Write Memory barrier */
#define mb()            RISCV_FENCE(iorw,iorw)

/* Read Memory barrier */
#define rmb()           RISCV_FENCE(ir,ir)

/* Write Memory barrier */
#define wmb()           RISCV_FENCE(ow,ow)

/* SMP Read & Write Memory barrier */
```

```
#define smp_mb()        RISCV_FENCE(rw,rw)

/* SMP Read Memory barrier */
#define smp_rmb()       RISCV_FENCE(r,r)

/* SMP Write Memory barrier */
#define smp_wmb()       RISCV_FENCE(w,w)

/* CPU relax for busy loop */
#define cpu_relax()                                 \
do {                                        \
    unsigned long __t;                      \
    __asm__ __volatile__ ("div %0, %0, zero" : "=r" (__t)); \
} while (0)

/* clang-format on */

#define __smp_store_release(p, v)    \
    do {                             \
        RISCV_FENCE(rw, w); \
        *(p) = (v);         \
    } while (0)

#define __smp_load_acquire(p)            \
    ({                                   \
        typeof(*p) ___p1 = *(p); \
        RISCV_FENCE(r, rw);      \
        ___p1;                   \
    })

#endif
```

30.34 include/sbi/riscv_dbtr.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define RV_MAX_TRIGGERS 32
- #define RV_DBTR_BIT(_prefix, _name) RV_DBTR_##_prefix##_##_name##_BIT
- #define RV_DBTR_BIT_MASK(_prefix, _name) RV_DBTR_##_prefix##_name##_BIT_MASK
- #define RV_DBTR_DECLARE_BIT(_prefix, _name, _val) RV_DBTR_BIT(_prefix, _name) = _val
- #define RV_DBTR_DECLARE_BIT_MASK(_prefix, _name, _width)
- #define CLEAR_DBTR_BIT(_target, _prefix, _bit_name) __clear_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)
- #define SET_DBTR_BIT(_target, _prefix, _bit_name) __set_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)
- #define RV_DBTR_SET_TDATA1_TYPE(_t1, _type)
- #define RV_DBTR_SET_MC_TYPE(_t1, _type)
- #define RV_DBTR_SET_MC6_TYPE(_t1, _type)
- #define RV_DBTR_SET_MC_EXEC(_t1) SET_DBTR_BIT(_t1, MC, EXEC)

- #define RV_DBTR_SET_MC_LOAD(_t1) SET_DBTR_BIT(_t1, MC, LOAD)
- #define RV_DBTR_SET_MC_STORE(_t1) SET_DBTR_BIT(_t1, MC, STORE)
- #define RV_DBTR_SET_MC_SIZELO(_t1, _val)
- #define RV_DBTR_SET_MC_SIZEHI(_t1, _val)
- #define RV_DBTR_SET_MC6_EXEC(_t1) SET_DBTR_BIT(_t1, MC6, EXEC)
- #define RV_DBTR_SET_MC6_LOAD(_t1) SET_DBTR_BIT(_t1, MC6, LOAD)
- #define RV_DBTR_SET_MC6_STORE(_t1) SET_DBTR_BIT(_t1, MC6, STORE)
- #define RV_DBTR_SET_MC6_SIZE(_t1, _val)

Typedefs

- typedef unsigned long riscv_dbtr_tdata1_mcontrol_t
- typedef unsigned long riscv_dbtr_tdata1_mcontrol6_t
- typedef unsigned long riscv_dbtr_tdata1_t

Enumerations

- enum {
  RISCV_DBTR_TRIG_NONE = 0 , RISCV_DBTR_TRIG_LEGACY , RISCV_DBTR_TRIG_MCONTROL , RISCV_DBTR_TRIG_ICOUNT ,
  RISCV_DBTR_TRIG_ITRIGGER , RISCV_DBTR_TRIG_ETRIGGER , RISCV_DBTR_TRIG_MCONTROL6
  }
- enum { RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) }
- enum { RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) }
- enum {
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) }
- enum {
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) }
- enum {
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) }

- enum {
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) }

30.34.1 Macro Definition Documentation

30.34.1.1 CLEAR_DBTR_BIT

```
#define CLEAR_DBTR_BIT(
            _target,
            _prefix,
            _bit_name )  __clear_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)
```

30.34.1.2 RV_DBTR_BIT

```
#define RV_DBTR_BIT(
            _prefix,
            _name )  RV_DBTR_##_prefix##_##_name##_BIT
```

30.34.1.3 RV_DBTR_BIT_MASK

```
#define RV_DBTR_BIT_MASK(
            _prefix,
            _name )  RV_DBTR_##_prefix##_name##_BIT_MASK
```

30.34.1.4 RV_DBTR_DECLARE_BIT

```
#define RV_DBTR_DECLARE_BIT(
            _prefix,
            _name,
            _val )  RV_DBTR_BIT(_prefix, _name) = _val
```

30.34.1.5 RV_DBTR_DECLARE_BIT_MASK

```
#define RV_DBTR_DECLARE_BIT_MASK(
            _prefix,
            _name,
            _width )
```

Value:

```
RV_DBTR_BIT_MASK(_prefix, _name) =                \
    (((1UL « _width) - 1) « RV_DBTR_BIT(_prefix, _name))
```

30.34.1.6 RV_DBTR_SET_MC6_EXEC

```
#define RV_DBTR_SET_MC6_EXEC(
            _t1 )  SET_DBTR_BIT(_t1, MC6, EXEC)
```

30.34.1.7 RV_DBTR_SET_MC6_LOAD

```
#define RV_DBTR_SET_MC6_LOAD(
            _t1 )  SET_DBTR_BIT(_t1, MC6, LOAD)
```

30.34.1.8 RV_DBTR_SET_MC6_SIZE

```
#define RV_DBTR_SET_MC6_SIZE(
            _t1,
            _val )
```

Value:

```
do {                                    \
    _t1 &= ~RV_DBTR_BIT_MASK(MC6, SIZE);        \
    _t1 |= ((_val « RV_DBTR_BIT(MC6, SIZE)) \
        & RV_DBTR_BIT_MASK(MC6, SIZE));     \
} while(0);
```

30.34.1.9 RV_DBTR_SET_MC6_STORE

```
#define RV_DBTR_SET_MC6_STORE(
            _t1 )   SET_DBTR_BIT(_t1, MC6, STORE)
```

30.34.1.10 RV_DBTR_SET_MC6_TYPE

```
#define RV_DBTR_SET_MC6_TYPE(
            _t1,
            _type )
```

Value:

```
do {                                    \
    _t1 &= ~RV_DBTR_BIT_MASK(MC6, TYPE);        \
    _t1 |= (((unsigned long)_type           \
         « RV_DBTR_BIT(MC6, TYPE))      \
        & RV_DBTR_BIT_MASK(MC6, TYPE));     \
}while (0);
```

30.34.1.11 RV_DBTR_SET_MC_EXEC

```
#define RV_DBTR_SET_MC_EXEC(
            _t1 )   SET_DBTR_BIT(_t1, MC, EXEC)
```

30.34.1.12 RV_DBTR_SET_MC_LOAD

```
#define RV_DBTR_SET_MC_LOAD(
            _t1 )   SET_DBTR_BIT(_t1, MC, LOAD)
```

30.34.1.13 RV_DBTR_SET_MC_SIZEHI

```
#define RV_DBTR_SET_MC_SIZEHI(
            _t1,
            _val )
```

Value:

```
do {                                    \
    _t1 &= ~RV_DBTR_BIT_MASK(MC, SIZEHI);       \
    _t1 |= ((_val « RV_DBTR_BIT(MC, SIZEHI))    \
        & RV_DBTR_BIT_MASK(MC, SIZEHI));    \
} while(0);
```

30.34.1.14 RV_DBTR_SET_MC_SIZELO

```
#define RV_DBTR_SET_MC_SIZELO(
            _t1,
            _val )
```

Value:

```
do {                                    \
    _t1 &= ~RV_DBTR_BIT_MASK(MC, SIZELO);       \
    _t1 |= ((_val « RV_DBTR_BIT(MC, SIZELO))    \
        & RV_DBTR_BIT_MASK(MC, SIZELO));    \
} while(0);
```

30.34.1.15 RV_DBTR_SET_MC_STORE

```
#define RV_DBTR_SET_MC_STORE(
            _t1 )  SET_DBTR_BIT(_t1, MC, STORE)
```

30.34.1.16 RV_DBTR_SET_MC_TYPE

```
#define RV_DBTR_SET_MC_TYPE(
            _t1,
            _type )
```

Value:

```
    do {                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC, TYPE);      \
        _t1 |= (((unsigned long)_type            \
             « RV_DBTR_BIT(MC, TYPE))        \
            & RV_DBTR_BIT_MASK(MC, TYPE));       \
    }while (0);
```

30.34.1.17 RV_DBTR_SET_TDATA1_TYPE

```
#define RV_DBTR_SET_TDATA1_TYPE(
            _t1,
            _type )
```

Value:

```
    do {                                \
        _t1 &= ~RV_DBTR_BIT_MASK(TDATA1, TYPE);          \
        _t1 |= (((unsigned long)_type                 \
             « RV_DBTR_BIT(TDATA1, TYPE))             \
            & RV_DBTR_BIT_MASK(TDATA1, TYPE));        \
    }while (0);
```

30.34.1.18 RV_MAX_TRIGGERS

```
#define RV_MAX_TRIGGERS 32
```

30.34.1.19 SET_DBTR_BIT

```
#define SET_DBTR_BIT(
            _target,
            _prefix,
            _bit_name )  __set_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)
```

30.34.2 Typedef Documentation

30.34.2.1 riscv_dbtr_tdata1_mcontrol6_t

```
typedef unsigned long riscv_dbtr_tdata1_mcontrol6_t
```

30.34.2.2 riscv_dbtr_tdata1_mcontrol_t

```
typedef unsigned long riscv_dbtr_tdata1_mcontrol_t
```

30.34.2.3 riscv_dbtr_tdata1_t

```
typedef unsigned long riscv_dbtr_tdata1_t
```

30.34.3 Enumeration Type Documentation

30.34.3.1 anonymous enum

```
anonymous enum
```

Enumerator

RISCV_DBTR_TRIG_NONE	

Enumerator

RISCV_DBTR_TRIG_LEGACY	
RISCV_DBTR_TRIG_MCONTROL	
RISCV_DBTR_TRIG_ICOUNT	
RISCV_DBTR_TRIG_ITRIGGER	
RISCV_DBTR_TRIG_ETRIGGER	
RISCV_DBTR_TRIG_MCONTROL6	

30.34.3.2 anonymous enum

`anonymous enum`

Enumerator

RV_DBTR_DECLARE_BIT	

30.34.3.3 anonymous enum

`anonymous enum`

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

30.34.3.4 anonymous enum

`anonymous enum`

Enumerator

RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	

30.34.3.5 anonymous enum

`anonymous enum`

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

30.34.3.6 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	

30.34.3.7 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

30.35 riscv_dbtr.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Ventana Micro System, Inc.
 *
 * Authors:
 *   Himanshu Chauhan <hchauhan@ventanamicro.com>
 */

#ifndef __RISCV_DBTR_H__
#define __RISCV_DBTR_H__

#define RV_MAX_TRIGGERS 32

enum {
    RISCV_DBTR_TRIG_NONE = 0,
    RISCV_DBTR_TRIG_LEGACY,
    RISCV_DBTR_TRIG_MCONTROL,
    RISCV_DBTR_TRIG_ICOUNT,
    RISCV_DBTR_TRIG_ITRIGGER,
    RISCV_DBTR_TRIG_ETRIGGER,
    RISCV_DBTR_TRIG_MCONTROL6,
};

#define RV_DBTR_BIT(_prefix, _name)     \
    RV_DBTR_##_prefix##_##_name##_BIT

#define RV_DBTR_BIT_MASK(_prefix, _name)    \
    RV_DBTR_##_prefix##_name##_BIT_MASK

#define RV_DBTR_DECLARE_BIT(_prefix, _name, _val)   \
    RV_DBTR_BIT(_prefix, _name) = _val

#define RV_DBTR_DECLARE_BIT_MASK(_prefix, _name, _width)        \
    RV_DBTR_BIT_MASK(_prefix, _name) =              \
        (((1UL << _width) - 1) << RV_DBTR_BIT(_prefix, _name))

#define CLEAR_DBTR_BIT(_target, _prefix, _bit_name)     \
    __clear_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)

#define SET_DBTR_BIT(_target, _prefix, _bit_name)       \
    __set_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)

/* Trigger Data 1 */
enum {
    RV_DBTR_DECLARE_BIT(TDATA1, DATA,   0),
#if __riscv_xlen == 64
    RV_DBTR_DECLARE_BIT(TDATA1, DMODE,  59),
    RV_DBTR_DECLARE_BIT(TDATA1, TYPE,   60),
#elif __riscv_xlen == 32
    RV_DBTR_DECLARE_BIT(TDATA1, DMODE,  27),
    RV_DBTR_DECLARE_BIT(TDATA1, TYPE,   28),
#else
    #error "Unknown __riscv_xlen"
```

```
#endif
};

enum {
#if __riscv_xlen == 64
    RV_DBTR_DECLARE_BIT_MASK(TDATA1, DATA,  59),
#elif __riscv_xlen == 32
    RV_DBTR_DECLARE_BIT_MASK(TDATA1, DATA,  27),
#else
    #error "Unknown __riscv_xlen"
#endif
    RV_DBTR_DECLARE_BIT_MASK(TDATA1, DMODE, 1),
    RV_DBTR_DECLARE_BIT_MASK(TDATA1, TYPE,  4),
};

/* MC - Match Control Type Register */
enum {
    RV_DBTR_DECLARE_BIT(MC, LOAD,    0),
    RV_DBTR_DECLARE_BIT(MC, STORE,   1),
    RV_DBTR_DECLARE_BIT(MC, EXEC,    2),
    RV_DBTR_DECLARE_BIT(MC, U,       3),
    RV_DBTR_DECLARE_BIT(MC, S,       4),
    RV_DBTR_DECLARE_BIT(MC, RES2,    5),
    RV_DBTR_DECLARE_BIT(MC, M,       6),
    RV_DBTR_DECLARE_BIT(MC, MATCH,   7),
    RV_DBTR_DECLARE_BIT(MC, CHAIN,   11),
    RV_DBTR_DECLARE_BIT(MC, ACTION,  12),
    RV_DBTR_DECLARE_BIT(MC, SIZELO,  16),
    RV_DBTR_DECLARE_BIT(MC, TIMING,  18),
    RV_DBTR_DECLARE_BIT(MC, SELECT,  19),
    RV_DBTR_DECLARE_BIT(MC, HIT,     20),
#if __riscv_xlen >= 64
    RV_DBTR_DECLARE_BIT(MC, SIZEHI,  21),
#endif
#if __riscv_xlen == 64
    RV_DBTR_DECLARE_BIT(MC, MASKMAX, 53),
    RV_DBTR_DECLARE_BIT(MC, DMODE,   59),
    RV_DBTR_DECLARE_BIT(MC, TYPE,    60),
#elif __riscv_xlen == 32
    RV_DBTR_DECLARE_BIT(MC, MASKMAX, 21),
    RV_DBTR_DECLARE_BIT(MC, DMODE,   27),
    RV_DBTR_DECLARE_BIT(MC, TYPE,    28),
#else
    #error "Unknown __riscv_xlen"
#endif
};

enum {
    RV_DBTR_DECLARE_BIT_MASK(MC, LOAD,    1),
    RV_DBTR_DECLARE_BIT_MASK(MC, STORE,   1),
    RV_DBTR_DECLARE_BIT_MASK(MC, EXEC,    1),
    RV_DBTR_DECLARE_BIT_MASK(MC, U,       1),
    RV_DBTR_DECLARE_BIT_MASK(MC, S,       1),
    RV_DBTR_DECLARE_BIT_MASK(MC, RES2,    1),
    RV_DBTR_DECLARE_BIT_MASK(MC, M,       1),
    RV_DBTR_DECLARE_BIT_MASK(MC, MATCH,   4),
    RV_DBTR_DECLARE_BIT_MASK(MC, CHAIN,   1),
    RV_DBTR_DECLARE_BIT_MASK(MC, ACTION,  4),
    RV_DBTR_DECLARE_BIT_MASK(MC, SIZELO,  2),
    RV_DBTR_DECLARE_BIT_MASK(MC, TIMING,  1),
    RV_DBTR_DECLARE_BIT_MASK(MC, SELECT,  1),
    RV_DBTR_DECLARE_BIT_MASK(MC, HIT,     1),
#if __riscv_xlen >= 64
    RV_DBTR_DECLARE_BIT_MASK(MC, SIZEHI,  2),
#endif
    RV_DBTR_DECLARE_BIT_MASK(MC, MASKMAX, 6),
    RV_DBTR_DECLARE_BIT_MASK(MC, DMODE,   1),
    RV_DBTR_DECLARE_BIT_MASK(MC, TYPE,    4),
};

/* MC6 - Match Control 6 Type Register */
enum {
    RV_DBTR_DECLARE_BIT(MC6, LOAD,   0),
    RV_DBTR_DECLARE_BIT(MC6, STORE,  1),
    RV_DBTR_DECLARE_BIT(MC6, EXEC,   2),
    RV_DBTR_DECLARE_BIT(MC6, U,      3),
    RV_DBTR_DECLARE_BIT(MC6, S,      4),
    RV_DBTR_DECLARE_BIT(MC6, RES2,   5),
    RV_DBTR_DECLARE_BIT(MC6, M,      6),
    RV_DBTR_DECLARE_BIT(MC6, MATCH,  7),
    RV_DBTR_DECLARE_BIT(MC6, CHAIN,  11),
    RV_DBTR_DECLARE_BIT(MC6, ACTION, 12),
    RV_DBTR_DECLARE_BIT(MC6, SIZE,   16),
    RV_DBTR_DECLARE_BIT(MC6, TIMING, 20),
    RV_DBTR_DECLARE_BIT(MC6, SELECT, 21),
    RV_DBTR_DECLARE_BIT(MC6, HIT,    22),
    RV_DBTR_DECLARE_BIT(MC6, VU,     23),
```

```
    RV_DBTR_DECLARE_BIT(MC6, VS,     24),
#if __riscv_xlen == 64
    RV_DBTR_DECLARE_BIT(MC6, DMODE,  59),
    RV_DBTR_DECLARE_BIT(MC6, TYPE,   60),
#elif __riscv_xlen == 32
    RV_DBTR_DECLARE_BIT(MC6, DMODE,  27),
    RV_DBTR_DECLARE_BIT(MC6, TYPE,   28),
#else
    #error "Unknown __riscv_xlen"
#endif
};

enum {
    RV_DBTR_DECLARE_BIT_MASK(MC6, LOAD,   1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, STORE,  1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, EXEC,   1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, U,      1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, S,      1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, RES2,   1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, M,      1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, MATCH,  4),
    RV_DBTR_DECLARE_BIT_MASK(MC6, CHAIN,  1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, ACTION, 4),
    RV_DBTR_DECLARE_BIT_MASK(MC6, SIZE,   4),
    RV_DBTR_DECLARE_BIT_MASK(MC6, TIMING, 1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, SELECT, 1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, HIT,    1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, VU,     1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, VS,     1),
#if __riscv_xlen == 64
    RV_DBTR_DECLARE_BIT_MASK(MC6, DMODE,  1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, TYPE,   4),
#elif __riscv_xlen == 32
    RV_DBTR_DECLARE_BIT_MASK(MC6, DMODE,  1),
    RV_DBTR_DECLARE_BIT_MASK(MC6, TYPE,   4),
#else
    #error "Unknown __riscv_xlen"
#endif
};

#define RV_DBTR_SET_TDATA1_TYPE(_t1, _type)             \
    do {                                                \
        _t1 &= ~RV_DBTR_BIT_MASK(TDATA1, TYPE);         \
        _t1 |= (((unsigned long)_type                   \
             « RV_DBTR_BIT(TDATA1, TYPE))               \
            & RV_DBTR_BIT_MASK(TDATA1, TYPE));          \
    }while (0);

#define RV_DBTR_SET_MC_TYPE(_t1, _type)                 \
    do {                                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC, TYPE);             \
        _t1 |= (((unsigned long)_type                   \
             « RV_DBTR_BIT(MC, TYPE))                   \
            & RV_DBTR_BIT_MASK(MC, TYPE));              \
    }while (0);

#define RV_DBTR_SET_MC6_TYPE(_t1, _type)                \
    do {                                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC6, TYPE);            \
        _t1 |= (((unsigned long)_type                   \
             « RV_DBTR_BIT(MC6, TYPE))                  \
            & RV_DBTR_BIT_MASK(MC6, TYPE));             \
    }while (0);

#define RV_DBTR_SET_MC_EXEC(_t1)                        \
    SET_DBTR_BIT(_t1, MC, EXEC)

#define RV_DBTR_SET_MC_LOAD(_t1)                        \
    SET_DBTR_BIT(_t1, MC, LOAD)

#define RV_DBTR_SET_MC_STORE(_t1)                       \
    SET_DBTR_BIT(_t1, MC, STORE)

#define RV_DBTR_SET_MC_SIZELO(_t1, _val)                \
    do {                                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC, SIZELO);           \
        _t1 |= ((_val « RV_DBTR_BIT(MC, SIZELO))        \
            & RV_DBTR_BIT_MASK(MC, SIZELO));            \
    } while(0);

#define RV_DBTR_SET_MC_SIZEHI(_t1, _val)                \
    do {                                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC, SIZEHI);           \
        _t1 |= ((_val « RV_DBTR_BIT(MC, SIZEHI))        \
            & RV_DBTR_BIT_MASK(MC, SIZEHI));            \
    } while(0);

```

```
#define RV_DBTR_SET_MC6_EXEC(_t1)        \
    SET_DBTR_BIT(_t1, MC6, EXEC)

#define RV_DBTR_SET_MC6_LOAD(_t1)        \
    SET_DBTR_BIT(_t1, MC6, LOAD)

#define RV_DBTR_SET_MC6_STORE(_t1)       \
    SET_DBTR_BIT(_t1, MC6, STORE)

#define RV_DBTR_SET_MC6_SIZE(_t1, _val)              \
    do {                                             \
        _t1 &= ~RV_DBTR_BIT_MASK(MC6, SIZE);         \
        _t1 |= ((_val « RV_DBTR_BIT(MC6, SIZE))     \
            & RV_DBTR_BIT_MASK(MC6, SIZE));      \
    } while(0);

typedef unsigned long riscv_dbtr_tdata1_mcontrol_t;
typedef unsigned long riscv_dbtr_tdata1_mcontrol6_t;
typedef unsigned long riscv_dbtr_tdata1_t;

#endif /* __RISCV_DBTR_H__ */
```

30.36 include/sbi/riscv_elf.h File Reference

Macros

- #define R_RISCV_RELATIVE 3

30.36.1 Macro Definition Documentation

30.36.1.1 R_RISCV_RELATIVE

```
#define R_RISCV_RELATIVE 3
```

30.37 riscv_elf.h

Go to the documentation of this file.

```
#ifndef __RISCV_ELF_H__
#define __RISCV_ELF_H__

#define R_RISCV_RELATIVE    3

#endif
```

30.38 include/sbi/riscv_encoding.h File Reference

`#include <sbi/sbi_const.h>`

Include dependency graph for riscv_encoding.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define MSTATUS_SIE _UL(0x00000002)
- #define MSTATUS_MIE _UL(0x00000008)
- #define MSTATUS_SPIE_SHIFT 5
- #define MSTATUS_SPIE (_UL(1) << MSTATUS_SPIE_SHIFT)
- #define MSTATUS_UBE _UL(0x00000040)
- #define MSTATUS_MPIE _UL(0x00000080)
- #define MSTATUS_SPP_SHIFT 8
- #define MSTATUS_SPP (_UL(1) << MSTATUS_SPP_SHIFT)
- #define MSTATUS_MPP_SHIFT 11
- #define MSTATUS_MPP (_UL(3) << MSTATUS_MPP_SHIFT)
- #define MSTATUS_FS _UL(0x00006000)
- #define MSTATUS_XS _UL(0x00018000)
- #define MSTATUS_VS _UL(0x00000600)
- #define MSTATUS_MPRV _UL(0x00020000)
- #define MSTATUS_SUM _UL(0x00040000)
- #define MSTATUS_MXR _UL(0x00080000)
- #define MSTATUS_TVM _UL(0x00100000)
- #define MSTATUS_TW _UL(0x00200000)
- #define MSTATUS_TSR _UL(0x00400000)
- #define MSTATUS_SPELP _UL(0x00800000)
- #define MSTATUS32_SD _UL(0x80000000)
- #define MSTATUSH_SBE _UL(0x00000010)
- #define MSTATUSH_MBE _UL(0x00000020)
- #define MSTATUSH_GVA _UL(0x00000040)
- #define MSTATUSH_GVA_SHIFT 6
- #define MSTATUSH_MPV _UL(0x00000080)
- #define MSTATUSH_MPELP _UL(0x00000200)
- #define MSTATUSH_MDT _UL(0x00000400)
- #define MSTATUS32_SD _UL(0x80000000)
- #define MSTATUS64_SD _ULL(0x8000000000000000)
- #define SSTATUS_SIE MSTATUS_SIE
- #define SSTATUS_SPIE_SHIFT MSTATUS_SPIE_SHIFT
- #define SSTATUS_SPIE MSTATUS_SPIE
- #define SSTATUS_SPP_SHIFT MSTATUS_SPP_SHIFT
- #define SSTATUS_SPP MSTATUS_SPP
- #define SSTATUS_FS MSTATUS_FS
- #define SSTATUS_XS MSTATUS_XS
- #define SSTATUS_VS MSTATUS_VS
- #define SSTATUS_SUM MSTATUS_SUM
- #define SSTATUS_MXR MSTATUS_MXR
- #define SSTATUS32_SD MSTATUS32_SD
- #define SSTATUS64_UXL MSTATUS_UXL
- #define SSTATUS64_SD MSTATUS64_SD
- #define HSTATUS_VTSR _UL(0x00400000)
- #define HSTATUS_VTW _UL(0x00200000)
- #define HSTATUS_VTVM _UL(0x00100000)

- #define HSTATUS_VGEIN _UL(0x0003f000)
- #define HSTATUS_VGEIN_SHIFT 12
- #define HSTATUS_HU _UL(0x00000200)
- #define HSTATUS_SPVP _UL(0x00000100)
- #define HSTATUS_SPV _UL(0x00000080)
- #define HSTATUS_GVA _UL(0x00000040)
- #define HSTATUS_VSBE _UL(0x00000020)
- #define MCAUSE_IRQ_MASK (_UL(1) << (__riscv_xlen - 1))
- #define IRQ_S_SOFT 1
- #define IRQ_VS_SOFT 2
- #define IRQ_M_SOFT 3
- #define IRQ_S_TIMER 5
- #define IRQ_VS_TIMER 6
- #define IRQ_M_TIMER 7
- #define IRQ_S_EXT 9
- #define IRQ_VS_EXT 10
- #define IRQ_M_EXT 11
- #define IRQ_S_GEXT 12
- #define IRQ_PMU_OVF 13
- #define MIP_SSIP (_UL(1) << IRQ_S_SOFT)
- #define MIP_VSSIP (_UL(1) << IRQ_VS_SOFT)
- #define MIP_MSIP (_UL(1) << IRQ_M_SOFT)
- #define MIP_STIP (_UL(1) << IRQ_S_TIMER)
- #define MIP_VSTIP (_UL(1) << IRQ_VS_TIMER)
- #define MIP_MTIP (_UL(1) << IRQ_M_TIMER)
- #define MIP_SEIP (_UL(1) << IRQ_S_EXT)
- #define MIP_VSEIP (_UL(1) << IRQ_VS_EXT)
- #define MIP_MEIP (_UL(1) << IRQ_M_EXT)
- #define MIP_SGEIP (_UL(1) << IRQ_S_GEXT)
- #define MIP_LCOFIP (_UL(1) << IRQ_PMU_OVF)
- #define SIP_SSIP MIP_SSIP
- #define SIP_STIP MIP_STIP
- #define PRV_U _UL(0)
- #define PRV_S _UL(1)
- #define PRV_M _UL(3)
- #define SATP32_MODE _UL(0x80000000)
- #define SATP32_ASID _UL(0x7FC00000)
- #define SATP32_PPN _UL(0x003FFFFF)
- #define SATP64_MODE _ULL(0xF000000000000000)
- #define SATP64_ASID _ULL(0x0FFFF00000000000)
- #define SATP64_PPN _ULL(0x00000FFFFFFFFFFF)
- #define SATP_MODE_OFF _UL(0)
- #define SATP_MODE_SV32 _UL(1)
- #define SATP_MODE_SV39 _UL(8)
- #define SATP_MODE_SV48 _UL(9)
- #define SATP_MODE_SV57 _UL(10)
- #define SATP_MODE_SV64 _UL(11)
- #define HGATP_MODE_OFF _UL(0)
- #define HGATP_MODE_SV32X4 _UL(1)
- #define HGATP_MODE_SV39X4 _UL(8)
- #define HGATP_MODE_SV48X4 _UL(9)
- #define HGATP32_MODE_SHIFT 31
- #define HGATP32_VMID_SHIFT 22
- #define HGATP32_VMID_MASK _UL(0x1FC00000)
- #define HGATP32_PPN _UL(0x003FFFFF)

- #define HGATP64_MODE_SHIFT 60
- #define HGATP64_VMID_SHIFT 44
- #define HGATP64_VMID_MASK _ULL(0x03FFF00000000000)
- #define HGATP64_PPN _ULL(0x00000FFFFFFFFFFF)
- #define PMP_R _UL(0x01)
- #define PMP_W _UL(0x02)
- #define PMP_X _UL(0x04)
- #define PMP_A _UL(0x18)
- #define PMP_A_TOR _UL(0x08)
- #define PMP_A_NA4 _UL(0x10)
- #define PMP_A_NAPOT _UL(0x18)
- #define PMP_L _UL(0x80)
- #define PMP_SHIFT 2
- #define PMP_COUNT 64
- #define PMP_ADDR_MASK _UL(0xFFFFFFFF)
- #define MSTATUS_SD MSTATUS32_SD
- #define SSTATUS_SD SSTATUS32_SD
- #define SATP_MODE SATP32_MODE
- #define HGATP_PPN HGATP32_PPN
- #define HGATP_VMID_SHIFT HGATP32_VMID_SHIFT
- #define HGATP_VMID_MASK HGATP32_VMID_MASK
- #define HGATP_MODE_SHIFT HGATP32_MODE_SHIFT
- #define TOPI_IID_SHIFT 16
- #define TOPI_IID_MASK 0xfff
- #define TOPI_IPRIO_MASK 0xff
- #define MHPMEVENTH_OF (_ULL(1) << 31)
- #define MHPMEVENTH_MINH (_ULL(1) << 30)
- #define MHPMEVENTH_SINH (_ULL(1) << 29)
- #define MHPMEVENTH_UINH (_ULL(1) << 28)
- #define MHPMEVENTH_VSINH (_ULL(1) << 27)
- #define MHPMEVENTH_VUINH (_ULL(1) << 26)
- #define MHPMEVENT_OF (MHPMEVENTH_OF << 32)
- #define MHPMEVENT_MINH (MHPMEVENTH_MINH << 32)
- #define MHPMEVENT_SINH (MHPMEVENTH_SINH << 32)
- #define MHPMEVENT_UINH (MHPMEVENTH_UINH << 32)
- #define MHPMEVENT_VSINH (MHPMEVENTH_VSINH << 32)
- #define MHPMEVENT_VUINH (MHPMEVENTH_VUINH << 32)
- #define MHPMEVENT_SSCOF_MASK _ULL(0xFF00000000000000)
- #define ENVCFG_STCE (_ULL(1) << 63)
- #define ENVCFG_PBMTE (_ULL(1) << 62)
- #define ENVCFG_ADUE_SHIFT 61
- #define ENVCFG_ADUE (_ULL(1) << ENVCFG_ADUE_SHIFT)
- #define ENVCFG_CDE (_ULL(1) << 60)
- #define ENVCFG_DTE_SHIFT 59
- #define ENVCFG_DTE (_ULL(1) << ENVCFG_DTE_SHIFT)
- #define ENVCFG_PMM (_ULL(0x3) << 32)
- #define ENVCFG_PMM_PMLEN_0 (_ULL(0x0) << 32)
- #define ENVCFG_PMM_PMLEN_7 (_ULL(0x2) << 32)
- #define ENVCFG_PMM_PMLEN_16 (_ULL(0x3) << 32)
- #define ENVCFG_CBZE (_UL(1) << 7)
- #define ENVCFG_CBCFE (_UL(1) << 6)
- #define ENVCFG_CBIE_SHIFT 4
- #define ENVCFG_CBIE (_UL(0x3) << ENVCFG_CBIE_SHIFT)
- #define ENVCFG_CBIE_ILL _UL(0x0)
- #define ENVCFG_CBIE_FLUSH _UL(0x1)

- #define ENVCFG_CBIE_INV _UL(0x3)
- #define ENVCFG_SSE_SHIFT 3
- #define ENVCFG_SSE (_UL(1) << ENVCFG_SSE_SHIFT)
- #define ENVCFG_LPE_SHIFT 2
- #define ENVCFG_LPE (_UL(1) << ENVCFG_LPE_SHIFT)
- #define ENVCFG_FIOM _UL(0x1)
- #define CSR_USTATUS 0x000
- #define CSR_UIE 0x004
- #define CSR_UTVEC 0x005
- #define CSR_SSP 0x011
- #define CSR_USCRATCH 0x040
- #define CSR_UEPC 0x041
- #define CSR_UCAUSE 0x042
- #define CSR_UTVAL 0x043
- #define CSR_UIP 0x044
- #define CSR_FFLAGS 0x001
- #define CSR_FRM 0x002
- #define CSR_FCSR 0x003
- #define CSR_CYCLE 0xc00
- #define CSR_TIME 0xc01
- #define CSR_INSTRET 0xc02
- #define CSR_HPMCOUNTER3 0xc03
- #define CSR_HPMCOUNTER4 0xc04
- #define CSR_HPMCOUNTER5 0xc05
- #define CSR_HPMCOUNTER6 0xc06
- #define CSR_HPMCOUNTER7 0xc07
- #define CSR_HPMCOUNTER8 0xc08
- #define CSR_HPMCOUNTER9 0xc09
- #define CSR_HPMCOUNTER10 0xc0a
- #define CSR_HPMCOUNTER11 0xc0b
- #define CSR_HPMCOUNTER12 0xc0c
- #define CSR_HPMCOUNTER13 0xc0d
- #define CSR_HPMCOUNTER14 0xc0e
- #define CSR_HPMCOUNTER15 0xc0f
- #define CSR_HPMCOUNTER16 0xc10
- #define CSR_HPMCOUNTER17 0xc11
- #define CSR_HPMCOUNTER18 0xc12
- #define CSR_HPMCOUNTER19 0xc13
- #define CSR_HPMCOUNTER20 0xc14
- #define CSR_HPMCOUNTER21 0xc15
- #define CSR_HPMCOUNTER22 0xc16
- #define CSR_HPMCOUNTER23 0xc17
- #define CSR_HPMCOUNTER24 0xc18
- #define CSR_HPMCOUNTER25 0xc19
- #define CSR_HPMCOUNTER26 0xc1a
- #define CSR_HPMCOUNTER27 0xc1b
- #define CSR_HPMCOUNTER28 0xc1c
- #define CSR_HPMCOUNTER29 0xc1d
- #define CSR_HPMCOUNTER30 0xc1e
- #define CSR_HPMCOUNTER31 0xc1f
- #define CSR_CYCLEH 0xc80
- #define CSR_TIMEH 0xc81
- #define CSR_INSTRETH 0xc82
- #define CSR_HPMCOUNTER3H 0xc83
- #define CSR_HPMCOUNTER4H 0xc84

- #define CSR_HPMCOUNTER5H 0xc85
- #define CSR_HPMCOUNTER6H 0xc86
- #define CSR_HPMCOUNTER7H 0xc87
- #define CSR_HPMCOUNTER8H 0xc88
- #define CSR_HPMCOUNTER9H 0xc89
- #define CSR_HPMCOUNTER10H 0xc8a
- #define CSR_HPMCOUNTER11H 0xc8b
- #define CSR_HPMCOUNTER12H 0xc8c
- #define CSR_HPMCOUNTER13H 0xc8d
- #define CSR_HPMCOUNTER14H 0xc8e
- #define CSR_HPMCOUNTER15H 0xc8f
- #define CSR_HPMCOUNTER16H 0xc90
- #define CSR_HPMCOUNTER17H 0xc91
- #define CSR_HPMCOUNTER18H 0xc92
- #define CSR_HPMCOUNTER19H 0xc93
- #define CSR_HPMCOUNTER20H 0xc94
- #define CSR_HPMCOUNTER21H 0xc95
- #define CSR_HPMCOUNTER22H 0xc96
- #define CSR_HPMCOUNTER23H 0xc97
- #define CSR_HPMCOUNTER24H 0xc98
- #define CSR_HPMCOUNTER25H 0xc99
- #define CSR_HPMCOUNTER26H 0xc9a
- #define CSR_HPMCOUNTER27H 0xc9b
- #define CSR_HPMCOUNTER28H 0xc9c
- #define CSR_HPMCOUNTER29H 0xc9d
- #define CSR_HPMCOUNTER30H 0xc9e
- #define CSR_HPMCOUNTER31H 0xc9f
- #define CSR_SSTATUS 0x100
- #define CSR_SIE 0x104
- #define CSR_STVEC 0x105
- #define CSR_SCOUNTEREN 0x106
- #define CSR_SENVCFG 0x10a
- #define CSR_SCOUNTINHIBIT 0x120
- #define CSR_SSCRATCH 0x140
- #define CSR_SEPC 0x141
- #define CSR_SCAUSE 0x142
- #define CSR_STVAL 0x143
- #define CSR_SIP 0x144
- #define CSR_STIMECMP 0x14D
- #define CSR_STIMECMPH 0x15D
- #define CSR_SATP 0x180
- #define CSR_SISELECT 0x150
- #define CSR_SIREG 0x151
- #define CSR_SIREG2 0x152
- #define CSR_SIREG3 0x153
- #define CSR_SIREG4 0x155
- #define CSR_SIREG5 0x156
- #define CSR_SIREG6 0x157
- #define CSR_STOPEI 0x15c
- #define CSR_STOPI 0xdb0
- #define CSR_SIEH 0x114
- #define CSR_SIPH 0x154
- #define CSR_SSTATEEN0 0x10C
- #define CSR_SSTATEEN1 0x10D
- #define CSR_SSTATEEN2 0x10E

- #define CSR_SSTATEEN3 0x10F
- #define CSR_HSTATUS 0x600
- #define CSR_HEDELEG 0x602
- #define CSR_HIDELEG 0x603
- #define CSR_HIE 0x604
- #define CSR_HCOUNTEREN 0x606
- #define CSR_HGEIE 0x607
- #define CSR_HENVCFG 0x60a
- #define CSR_HENVCFGH 0x61a
- #define CSR_HTVAL 0x643
- #define CSR_HIP 0x644
- #define CSR_HVIP 0x645
- #define CSR_HTINST 0x64a
- #define CSR_HGEIP 0xe12
- #define CSR_HGATP 0x680
- #define CSR_HTIMEDELTA 0x605
- #define CSR_HTIMEDELTAH 0x615
- #define CSR_VSSTATUS 0x200
- #define CSR_VSIE 0x204
- #define CSR_VSTVEC 0x205
- #define CSR_VSSCRATCH 0x240
- #define CSR_VSEPC 0x241
- #define CSR_VSCAUSE 0x242
- #define CSR_VSTVAL 0x243
- #define CSR_VSIP 0x244
- #define CSR_VSATP 0x280
- #define CSR_HVIEN 0x608
- #define CSR_HVICTL 0x609
- #define CSR_HVIPRIO1 0x646
- #define CSR_HVIPRIO2 0x647
- #define CSR_VSISELECT 0x250
- #define CSR_VSIREG 0x251
- #define CSR_VSIREG2 0x252
- #define CSR_VSIREG3 0x253
- #define CSR_VSIREG4 0x255
- #define CSR_VSIREG5 0x256
- #define CSR_VSIREG6 0x257
- #define CSR_VSTOPEI 0x25c
- #define CSR_VSTOPI 0xeb0
- #define CSR_HIDELEGH 0x613
- #define CSR_HVIENH 0x618
- #define CSR_HVIPH 0x655
- #define CSR_HVIPRIO1H 0x656
- #define CSR_HVIPRIO2H 0x657
- #define CSR_VSIEH 0x214
- #define CSR_VSIPH 0x254
- #define CSR_HSTATEEN0 0x60C
- #define CSR_HSTATEEN0H 0x61C
- #define CSR_HSTATEEN1 0x60D
- #define CSR_HSTATEEN1H 0x61D
- #define CSR_HSTATEEN2 0x60E
- #define CSR_HSTATEEN2H 0x61E
- #define CSR_HSTATEEN3 0x60F
- #define CSR_HSTATEEN3H 0x61F
- #define CSR_MVENDORID 0xf11

- #define CSR_MARCHID 0xf12
- #define CSR_MIMPID 0xf13
- #define CSR_MHARTID 0xf14
- #define CSR_MCONFIGPTR 0xf15
- #define CSR_MSTATUS 0x300
- #define CSR_MISA 0x301
- #define CSR_MEDELEG 0x302
- #define CSR_MIDELEG 0x303
- #define CSR_MIE 0x304
- #define CSR_MTVEC 0x305
- #define CSR_MCOUNTEREN 0x306
- #define CSR_MSTATUSH 0x310
- #define CSR_MENVCFG 0x30a
- #define CSR_MENVCFGH 0x31a
- #define CSR_MSCRATCH 0x340
- #define CSR_MEPC 0x341
- #define CSR_MCAUSE 0x342
- #define CSR_MTVAL 0x343
- #define CSR_MIP 0x344
- #define CSR_MTINST 0x34a
- #define CSR_MTVAL2 0x34b
- #define CSR_PMPCFG0 0x3a0
- #define CSR_PMPCFG1 0x3a1
- #define CSR_PMPCFG2 0x3a2
- #define CSR_PMPCFG3 0x3a3
- #define CSR_PMPCFG4 0x3a4
- #define CSR_PMPCFG5 0x3a5
- #define CSR_PMPCFG6 0x3a6
- #define CSR_PMPCFG7 0x3a7
- #define CSR_PMPCFG8 0x3a8
- #define CSR_PMPCFG9 0x3a9
- #define CSR_PMPCFG10 0x3aa
- #define CSR_PMPCFG11 0x3ab
- #define CSR_PMPCFG12 0x3ac
- #define CSR_PMPCFG13 0x3ad
- #define CSR_PMPCFG14 0x3ae
- #define CSR_PMPCFG15 0x3af
- #define CSR_PMPADDR0 0x3b0
- #define CSR_PMPADDR1 0x3b1
- #define CSR_PMPADDR2 0x3b2
- #define CSR_PMPADDR3 0x3b3
- #define CSR_PMPADDR4 0x3b4
- #define CSR_PMPADDR5 0x3b5
- #define CSR_PMPADDR6 0x3b6
- #define CSR_PMPADDR7 0x3b7
- #define CSR_PMPADDR8 0x3b8
- #define CSR_PMPADDR9 0x3b9
- #define CSR_PMPADDR10 0x3ba
- #define CSR_PMPADDR11 0x3bb
- #define CSR_PMPADDR12 0x3bc
- #define CSR_PMPADDR13 0x3bd
- #define CSR_PMPADDR14 0x3be
- #define CSR_PMPADDR15 0x3bf
- #define CSR_PMPADDR16 0x3c0
- #define CSR_PMPADDR17 0x3c1

- #define CSR_PMPADDR18 0x3c2
- #define CSR_PMPADDR19 0x3c3
- #define CSR_PMPADDR20 0x3c4
- #define CSR_PMPADDR21 0x3c5
- #define CSR_PMPADDR22 0x3c6
- #define CSR_PMPADDR23 0x3c7
- #define CSR_PMPADDR24 0x3c8
- #define CSR_PMPADDR25 0x3c9
- #define CSR_PMPADDR26 0x3ca
- #define CSR_PMPADDR27 0x3cb
- #define CSR_PMPADDR28 0x3cc
- #define CSR_PMPADDR29 0x3cd
- #define CSR_PMPADDR30 0x3ce
- #define CSR_PMPADDR31 0x3cf
- #define CSR_PMPADDR32 0x3d0
- #define CSR_PMPADDR33 0x3d1
- #define CSR_PMPADDR34 0x3d2
- #define CSR_PMPADDR35 0x3d3
- #define CSR_PMPADDR36 0x3d4
- #define CSR_PMPADDR37 0x3d5
- #define CSR_PMPADDR38 0x3d6
- #define CSR_PMPADDR39 0x3d7
- #define CSR_PMPADDR40 0x3d8
- #define CSR_PMPADDR41 0x3d9
- #define CSR_PMPADDR42 0x3da
- #define CSR_PMPADDR43 0x3db
- #define CSR_PMPADDR44 0x3dc
- #define CSR_PMPADDR45 0x3dd
- #define CSR_PMPADDR46 0x3de
- #define CSR_PMPADDR47 0x3df
- #define CSR_PMPADDR48 0x3e0
- #define CSR_PMPADDR49 0x3e1
- #define CSR_PMPADDR50 0x3e2
- #define CSR_PMPADDR51 0x3e3
- #define CSR_PMPADDR52 0x3e4
- #define CSR_PMPADDR53 0x3e5
- #define CSR_PMPADDR54 0x3e6
- #define CSR_PMPADDR55 0x3e7
- #define CSR_PMPADDR56 0x3e8
- #define CSR_PMPADDR57 0x3e9
- #define CSR_PMPADDR58 0x3ea
- #define CSR_PMPADDR59 0x3eb
- #define CSR_PMPADDR60 0x3ec
- #define CSR_PMPADDR61 0x3ed
- #define CSR_PMPADDR62 0x3ee
- #define CSR_PMPADDR63 0x3ef
- #define CSR_MCYCLE 0xb00
- #define CSR_MINSTRET 0xb02
- #define CSR_MHPMCOUNTER3 0xb03
- #define CSR_MHPMCOUNTER4 0xb04
- #define CSR_MHPMCOUNTER5 0xb05
- #define CSR_MHPMCOUNTER6 0xb06
- #define CSR_MHPMCOUNTER7 0xb07
- #define CSR_MHPMCOUNTER8 0xb08
- #define CSR_MHPMCOUNTER9 0xb09

- #define CSR_MHPMCOUNTER10 0xb0a
- #define CSR_MHPMCOUNTER11 0xb0b
- #define CSR_MHPMCOUNTER12 0xb0c
- #define CSR_MHPMCOUNTER13 0xb0d
- #define CSR_MHPMCOUNTER14 0xb0e
- #define CSR_MHPMCOUNTER15 0xb0f
- #define CSR_MHPMCOUNTER16 0xb10
- #define CSR_MHPMCOUNTER17 0xb11
- #define CSR_MHPMCOUNTER18 0xb12
- #define CSR_MHPMCOUNTER19 0xb13
- #define CSR_MHPMCOUNTER20 0xb14
- #define CSR_MHPMCOUNTER21 0xb15
- #define CSR_MHPMCOUNTER22 0xb16
- #define CSR_MHPMCOUNTER23 0xb17
- #define CSR_MHPMCOUNTER24 0xb18
- #define CSR_MHPMCOUNTER25 0xb19
- #define CSR_MHPMCOUNTER26 0xb1a
- #define CSR_MHPMCOUNTER27 0xb1b
- #define CSR_MHPMCOUNTER28 0xb1c
- #define CSR_MHPMCOUNTER29 0xb1d
- #define CSR_MHPMCOUNTER30 0xb1e
- #define CSR_MHPMCOUNTER31 0xb1f
- #define CSR_MCYCLEH 0xb80
- #define CSR_MINSTRETH 0xb82
- #define CSR_MHPMCOUNTER3H 0xb83
- #define CSR_MHPMCOUNTER4H 0xb84
- #define CSR_MHPMCOUNTER5H 0xb85
- #define CSR_MHPMCOUNTER6H 0xb86
- #define CSR_MHPMCOUNTER7H 0xb87
- #define CSR_MHPMCOUNTER8H 0xb88
- #define CSR_MHPMCOUNTER9H 0xb89
- #define CSR_MHPMCOUNTER10H 0xb8a
- #define CSR_MHPMCOUNTER11H 0xb8b
- #define CSR_MHPMCOUNTER12H 0xb8c
- #define CSR_MHPMCOUNTER13H 0xb8d
- #define CSR_MHPMCOUNTER14H 0xb8e
- #define CSR_MHPMCOUNTER15H 0xb8f
- #define CSR_MHPMCOUNTER16H 0xb90
- #define CSR_MHPMCOUNTER17H 0xb91
- #define CSR_MHPMCOUNTER18H 0xb92
- #define CSR_MHPMCOUNTER19H 0xb93
- #define CSR_MHPMCOUNTER20H 0xb94
- #define CSR_MHPMCOUNTER21H 0xb95
- #define CSR_MHPMCOUNTER22H 0xb96
- #define CSR_MHPMCOUNTER23H 0xb97
- #define CSR_MHPMCOUNTER24H 0xb98
- #define CSR_MHPMCOUNTER25H 0xb99
- #define CSR_MHPMCOUNTER26H 0xb9a
- #define CSR_MHPMCOUNTER27H 0xb9b
- #define CSR_MHPMCOUNTER28H 0xb9c
- #define CSR_MHPMCOUNTER29H 0xb9d
- #define CSR_MHPMCOUNTER30H 0xb9e
- #define CSR_MHPMCOUNTER31H 0xb9f
- #define CSR_MCOUNTINHIBIT 0x320
- #define CSR_MCYCLECFG 0x321

- #define CSR_MINSTRETCFG 0x322
- #define CSR_MHPMEVENT3 0x323
- #define CSR_MHPMEVENT4 0x324
- #define CSR_MHPMEVENT5 0x325
- #define CSR_MHPMEVENT6 0x326
- #define CSR_MHPMEVENT7 0x327
- #define CSR_MHPMEVENT8 0x328
- #define CSR_MHPMEVENT9 0x329
- #define CSR_MHPMEVENT10 0x32a
- #define CSR_MHPMEVENT11 0x32b
- #define CSR_MHPMEVENT12 0x32c
- #define CSR_MHPMEVENT13 0x32d
- #define CSR_MHPMEVENT14 0x32e
- #define CSR_MHPMEVENT15 0x32f
- #define CSR_MHPMEVENT16 0x330
- #define CSR_MHPMEVENT17 0x331
- #define CSR_MHPMEVENT18 0x332
- #define CSR_MHPMEVENT19 0x333
- #define CSR_MHPMEVENT20 0x334
- #define CSR_MHPMEVENT21 0x335
- #define CSR_MHPMEVENT22 0x336
- #define CSR_MHPMEVENT23 0x337
- #define CSR_MHPMEVENT24 0x338
- #define CSR_MHPMEVENT25 0x339
- #define CSR_MHPMEVENT26 0x33a
- #define CSR_MHPMEVENT27 0x33b
- #define CSR_MHPMEVENT28 0x33c
- #define CSR_MHPMEVENT29 0x33d
- #define CSR_MHPMEVENT30 0x33e
- #define CSR_MHPMEVENT31 0x33f
- #define CSR_MCYCLECFGH 0x721
- #define CSR_MINSTRETCFGH 0x722
- #define CSR_MHPMEVENT3H 0x723
- #define CSR_MHPMEVENT4H 0x724
- #define CSR_MHPMEVENT5H 0x725
- #define CSR_MHPMEVENT6H 0x726
- #define CSR_MHPMEVENT7H 0x727
- #define CSR_MHPMEVENT8H 0x728
- #define CSR_MHPMEVENT9H 0x729
- #define CSR_MHPMEVENT10H 0x72a
- #define CSR_MHPMEVENT11H 0x72b
- #define CSR_MHPMEVENT12H 0x72c
- #define CSR_MHPMEVENT13H 0x72d
- #define CSR_MHPMEVENT14H 0x72e
- #define CSR_MHPMEVENT15H 0x72f
- #define CSR_MHPMEVENT16H 0x730
- #define CSR_MHPMEVENT17H 0x731
- #define CSR_MHPMEVENT18H 0x732
- #define CSR_MHPMEVENT19H 0x733
- #define CSR_MHPMEVENT20H 0x734
- #define CSR_MHPMEVENT21H 0x735
- #define CSR_MHPMEVENT22H 0x736
- #define CSR_MHPMEVENT23H 0x737
- #define CSR_MHPMEVENT24H 0x738
- #define CSR_MHPMEVENT25H 0x739

- #define CSR_MHPMEVENT26H 0x73a
- #define CSR_MHPMEVENT27H 0x73b
- #define CSR_MHPMEVENT28H 0x73c
- #define CSR_MHPMEVENT29H 0x73d
- #define CSR_MHPMEVENT30H 0x73e
- #define CSR_MHPMEVENT31H 0x73f
- #define CSR_MSECCFG 0x747
- #define CSR_MSECCFGH 0x757
- #define MSECCFG_MML_SHIFT (0)
- #define MSECCFG_MML (_UL(1) << MSECCFG_MML_SHIFT)
- #define MSECCFG_MMWP_SHIFT (1)
- #define MSECCFG_MMWP (_UL(1) << MSECCFG_MMWP_SHIFT)
- #define MSECCFG_RLB_SHIFT (2)
- #define MSECCFG_RLB (_UL(1) << MSECCFG_RLB_SHIFT)
- #define MSECCFG_USEED_SHIFT (8)
- #define MSECCFG_USEED (_UL(1) << MSECCFG_USEED_SHIFT)
- #define MSECCFG_SSEED_SHIFT (9)
- #define MSECCFG_SSEED (_UL(1) << MSECCFG_SSEED_SHIFT)
- #define CSR_SCOUNTOVF 0xda0
- #define CSR_TSELECT 0x7a0
- #define CSR_TDATA1 0x7a1
- #define CSR_TDATA2 0x7a2
- #define CSR_TDATA3 0x7a3
- #define CSR_TINFO 0x7a4
- #define CSR_DCSR 0x7b0
- #define CSR_DPC 0x7b1
- #define CSR_DSCRATCH0 0x7b2
- #define CSR_DSCRATCH1 0x7b3
- #define CSR_MISELECT 0x350
- #define CSR_MIREG 0x351
- #define CSR_MIREG2 0x352
- #define CSR_MIREG3 0x353
- #define CSR_MIREG4 0x355
- #define CSR_MIREG5 0x356
- #define CSR_MIREG6 0x357
- #define CSR_MTOPEI 0x35c
- #define CSR_MTOPI 0xfb0
- #define CSR_MVIEN 0x308
- #define CSR_MVIP 0x309
- #define CSR_MSTATEEN0 0x30C
- #define CSR_MSTATEEN0H 0x31C
- #define CSR_MSTATEEN1 0x30D
- #define CSR_MSTATEEN1H 0x31D
- #define CSR_MSTATEEN2 0x30E
- #define CSR_MSTATEEN2H 0x31E
- #define CSR_MSTATEEN3 0x30F
- #define CSR_MSTATEEN3H 0x31F
- #define CSR_MIDELEGH 0x313
- #define CSR_MIEH 0x314
- #define CSR_MVIENH 0x318
- #define CSR_MVIPH 0x319
- #define CSR_MIPH 0x354
- #define CSR_VSTART 0x8
- #define CSR_VL 0xc20
- #define CSR_VTYPE 0xc21

- #define CSR_VLENB 0xc22
- #define CAUSE_MISALIGNED_FETCH 0x0
- #define CAUSE_FETCH_ACCESS 0x1
- #define CAUSE_ILLEGAL_INSTRUCTION 0x2
- #define CAUSE_BREAKPOINT 0x3
- #define CAUSE_MISALIGNED_LOAD 0x4
- #define CAUSE_LOAD_ACCESS 0x5
- #define CAUSE_MISALIGNED_STORE 0x6
- #define CAUSE_STORE_ACCESS 0x7
- #define CAUSE_USER_ECALL 0x8
- #define CAUSE_SUPERVISOR_ECALL 0x9
- #define CAUSE_VIRTUAL_SUPERVISOR_ECALL 0xa
- #define CAUSE_MACHINE_ECALL 0xb
- #define CAUSE_FETCH_PAGE_FAULT 0xc
- #define CAUSE_LOAD_PAGE_FAULT 0xd
- #define CAUSE_STORE_PAGE_FAULT 0xf
- #define CAUSE_DOUBLE_TRAP 0x10
- #define CAUSE_SW_CHECK_EXCP 0x12
- #define CAUSE_FETCH_GUEST_PAGE_FAULT 0x14
- #define CAUSE_LOAD_GUEST_PAGE_FAULT 0x15
- #define CAUSE_VIRTUAL_INST_FAULT 0x16
- #define CAUSE_STORE_GUEST_PAGE_FAULT 0x17
- #define SMSTATEEN_MAX_COUNT 4
- #define SMSTATEEN0_CS_SHIFT 0
- #define SMSTATEEN0_CS (_ULL(1) << SMSTATEEN0_CS_SHIFT)
- #define SMSTATEEN0_FCSR_SHIFT 1
- #define SMSTATEEN0_FCSR (_ULL(1) << SMSTATEEN0_FCSR_SHIFT)
- #define SMSTATEEN0_CONTEXT_SHIFT 57
- #define SMSTATEEN0_CONTEXT (_ULL(1) << SMSTATEEN0_CONTEXT_SHIFT)
- #define SMSTATEEN0_IMSIC_SHIFT 58
- #define SMSTATEEN0_IMSIC (_ULL(1) << SMSTATEEN0_IMSIC_SHIFT)
- #define SMSTATEEN0_AIA_SHIFT 59
- #define SMSTATEEN0_AIA (_ULL(1) << SMSTATEEN0_AIA_SHIFT)
- #define SMSTATEEN0_SVSLCT_SHIFT 60
- #define SMSTATEEN0_SVSLCT (_ULL(1) << SMSTATEEN0_SVSLCT_SHIFT)
- #define SMSTATEEN0_HSENVCFG_SHIFT 62
- #define SMSTATEEN0_HSENVCFG (_ULL(1) << SMSTATEEN0_HSENVCFG_SHIFT)
- #define SMSTATEEN_STATEN_SHIFT 63
- #define SMSTATEEN_STATEN (_ULL(1) << SMSTATEEN_STATEN_SHIFT)
- #define INSN_MATCH_LB 0x3
- #define INSN_MASK_LB 0x707f
- #define INSN_MATCH_LH 0x1003
- #define INSN_MASK_LH 0x707f
- #define INSN_MATCH_LW 0x2003
- #define INSN_MASK_LW 0x707f
- #define INSN_MATCH_LD 0x3003
- #define INSN_MASK_LD 0x707f
- #define INSN_MATCH_LBU 0x4003
- #define INSN_MASK_LBU 0x707f
- #define INSN_MATCH_LHU 0x5003
- #define INSN_MASK_LHU 0x707f
- #define INSN_MATCH_LWU 0x6003
- #define INSN_MASK_LWU 0x707f
- #define INSN_MATCH_SB 0x23
- #define INSN_MASK_SB 0x707f

- #define INSN_MATCH_SH 0x1023
- #define INSN_MASK_SH 0x707f
- #define INSN_MATCH_SW 0x2023
- #define INSN_MASK_SW 0x707f
- #define INSN_MATCH_SD 0x3023
- #define INSN_MASK_SD 0x707f
- #define INSN_MATCH_FLW 0x2007
- #define INSN_MASK_FLW 0x707f
- #define INSN_MATCH_FLD 0x3007
- #define INSN_MASK_FLD 0x707f
- #define INSN_MATCH_FLQ 0x4007
- #define INSN_MASK_FLQ 0x707f
- #define INSN_MATCH_FSW 0x2027
- #define INSN_MASK_FSW 0x707f
- #define INSN_MATCH_FSD 0x3027
- #define INSN_MASK_FSD 0x707f
- #define INSN_MATCH_FSQ 0x4027
- #define INSN_MASK_FSQ 0x707f
- #define INSN_MATCH_C_LD 0x6000
- #define INSN_MASK_C_LD 0xe003
- #define INSN_MATCH_C_SD 0xe000
- #define INSN_MASK_C_SD 0xe003
- #define INSN_MATCH_C_LW 0x4000
- #define INSN_MASK_C_LW 0xe003
- #define INSN_MATCH_C_SW 0xc000
- #define INSN_MASK_C_SW 0xe003
- #define INSN_MATCH_C_LDSP 0x6002
- #define INSN_MASK_C_LDSP 0xe003
- #define INSN_MATCH_C_SDSP 0xe002
- #define INSN_MASK_C_SDSP 0xe003
- #define INSN_MATCH_C_LWSP 0x4002
- #define INSN_MASK_C_LWSP 0xe003
- #define INSN_MATCH_C_SWSP 0xc002
- #define INSN_MASK_C_SWSP 0xe003
- #define INSN_MATCH_C_FLD 0x2000
- #define INSN_MASK_C_FLD 0xe003
- #define INSN_MATCH_C_FLW 0x6000
- #define INSN_MASK_C_FLW 0xe003
- #define INSN_MATCH_C_FSD 0xa000
- #define INSN_MASK_C_FSD 0xe003
- #define INSN_MATCH_C_FSW 0xe000
- #define INSN_MASK_C_FSW 0xe003
- #define INSN_MATCH_C_FLDSP 0x2002
- #define INSN_MASK_C_FLDSP 0xe003
- #define INSN_MATCH_C_FSDSP 0xa002
- #define INSN_MASK_C_FSDSP 0xe003
- #define INSN_MATCH_C_FLWSP 0x6002
- #define INSN_MASK_C_FLWSP 0xe003
- #define INSN_MATCH_C_FSWSP 0xe002
- #define INSN_MASK_C_FSWSP 0xe003
- #define INSN_MATCH_C_LHU 0x8400
- #define INSN_MASK_C_LHU 0xfc43
- #define INSN_MATCH_C_LH 0x8440
- #define INSN_MASK_C_LH 0xfc43
- #define INSN_MATCH_C_SH 0x8c00

- #define INSN_MASK_C_SH 0xfc43
- #define INSN_MASK_WFI 0xffffff00
- #define INSN_MATCH_WFI 0x10500000
- #define INSN_MASK_FENCE_TSO 0xffffffff
- #define INSN_MATCH_FENCE_TSO 0x8330000f
- #define INSN_MASK_VECTOR_UNIT_STRIDE 0xfdf0707f
- #define INSN_MASK_VECTOR_FAULT_ONLY_FIRST 0xfdf0707f
- #define INSN_MASK_VECTOR_STRIDE 0xfc00707f
- #define INSN_MASK_VECTOR_WHOLE_REG 0xfff0707f
- #define INSN_MASK_VECTOR_INDEXED 0xfc00707f
- #define INSN_MATCH_VLUXSEG(n, bits)
- #define INSN_MATCH_VSUXSEG(n, bits)
- #define INSN_MATCH_VLOXSEG(n, bits)
- #define INSN_MATCH_VSOXSEG(n, bits)
- #define INSN_MATCH_VLSSEG(n, bits)
- #define INSN_MATCH_VSSSEG(n, bits)
- #define INSN_MATCH_VSSEG(n, bits)
- #define INSN_MATCH_VLSEG(n, bits)
- #define INSN_MATCH_VLSEGFF(n, bits)
- #define INSN_MATCH_VLE16V 0x00005007
- #define INSN_MATCH_VLE32V 0x00006007
- #define INSN_MATCH_VLE64V 0x00007007
- #define INSN_MATCH_VSE16V 0x00005027
- #define INSN_MATCH_VSE32V 0x00006027
- #define INSN_MATCH_VSE64V 0x00007027
- #define INSN_MATCH_VLSE16V 0x08005007
- #define INSN_MATCH_VLSE32V 0x08006007
- #define INSN_MATCH_VLSE64V 0x08007007
- #define INSN_MATCH_VSSE16V 0x08005027
- #define INSN_MATCH_VSSE32V 0x08006027
- #define INSN_MATCH_VSSE64V 0x08007027
- #define INSN_MATCH_VLOXEI16V 0x0c005007
- #define INSN_MATCH_VLOXEI32V 0x0c006007
- #define INSN_MATCH_VLOXEI64V 0x0c007007
- #define INSN_MATCH_VSOXEI16V 0x0c005027
- #define INSN_MATCH_VSOXEI32V 0x0c006027
- #define INSN_MATCH_VSOXEI64V 0x0c007027
- #define INSN_MATCH_VLUXEI16V 0x04005007
- #define INSN_MATCH_VLUXEI32V 0x04006007
- #define INSN_MATCH_VLUXEI64V 0x04007007
- #define INSN_MATCH_VSUXEI16V 0x04005027
- #define INSN_MATCH_VSUXEI32V 0x04006027
- #define INSN_MATCH_VSUXEI64V 0x04007027
- #define INSN_MATCH_VLE16FFV 0x01005007
- #define INSN_MATCH_VLE32FFV 0x01006007
- #define INSN_MATCH_VLE64FFV 0x01007007
- #define INSN_MATCH_VL1RE8V 0x02800007
- #define INSN_MATCH_VL1RE16V 0x02805007
- #define INSN_MATCH_VL1RE32V 0x02806007
- #define INSN_MATCH_VL1RE64V 0x02807007
- #define INSN_MATCH_VL2RE8V 0x22800007
- #define INSN_MATCH_VL2RE16V 0x22805007
- #define INSN_MATCH_VL2RE32V 0x22806007
- #define INSN_MATCH_VL2RE64V 0x22807007
- #define INSN_MATCH_VL4RE8V 0x62800007

- #define INSN_MATCH_VL4RE16V 0x62805007
- #define INSN_MATCH_VL4RE32V 0x62806007
- #define INSN_MATCH_VL4RE64V 0x62807007
- #define INSN_MATCH_VL8RE8V 0xe2800007
- #define INSN_MATCH_VL8RE16V 0xe2805007
- #define INSN_MATCH_VL8RE32V 0xe2806007
- #define INSN_MATCH_VL8RE64V 0xe2807007
- #define INSN_MATCH_VS1RV 0x02800027
- #define INSN_MATCH_VS2RV 0x22800027
- #define INSN_MATCH_VS4RV 0x62800027
- #define INSN_MATCH_VS8RV 0xe2800027
- #define INSN_MASK_VECTOR_LOAD_STORE 0x7f
- #define INSN_MATCH_VECTOR_LOAD 0x07
- #define INSN_MATCH_VECTOR_STORE 0x27
- #define IS_VECTOR_LOAD_STORE(insn)
- #define IS_VECTOR_INSN_MATCH(insn, match, mask) (((insn) & (mask)) == ((match) & (mask)))
- #define IS_UNIT_STRIDE_MATCH(insn, match) IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_UNIT_STR
- #define IS_STRIDE_MATCH(insn, match) IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_STRIDE)
- #define IS_INDEXED_MATCH(insn, match) IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_INDEXED)
- #define IS_FAULT_ONLY_FIRST_MATCH(insn, match) IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_FAULT_ONLY_FIRST)
- #define IS_WHOLE_REG_MATCH(insn, match) IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_WHOLE_F
- #define IS_UNIT_STRIDE_LOAD(insn)
- #define IS_UNIT_STRIDE_STORE(insn)
- #define IS_STRIDE_LOAD(insn)
- #define IS_STRIDE_STORE(insn)
- #define IS_INDEXED_LOAD(insn)
- #define IS_INDEXED_STORE(insn)
- #define IS_FAULT_ONLY_FIRST_LOAD(insn)
- #define IS_WHOLE_REG_LOAD(insn)
- #define IS_WHOLE_REG_STORE(insn)
- #define VM_MASK 0x1
- #define VIEW_MASK 0x3
- #define VSEW_MASK 0x3
- #define VLMUL_MASK 0x7
- #define VD_MASK 0x1f
- #define VS2_MASK 0x1f
- #define INSN_16BIT_MASK 0x3
- #define INSN_32BIT_MASK 0x1c
- #define INSN_IS_16BIT(insn) (((insn) & INSN_16BIT_MASK) != INSN_16BIT_MASK)
- #define INSN_IS_32BIT(insn)
- #define INSN_LEN(insn) (INSN_IS_16BIT(insn) ? 2 : 4)
- #define LOG_REGBYTES 2
- #define REGBYTES (1 << LOG_REGBYTES)
- #define SH_VSEW 3
- #define SH_VIEW 12
- #define SH_VD 7
- #define SH_VS2 20
- #define SH_VM 25
- #define SH_MEW 28
- #define SH_RD 7
- #define SH_RS1 15
- #define SH_RS2 20
- #define SH_RS2C 2
- #define RV_X(x, s, n) (((x) >> (s)) & ((1 << (n)) - 1))

- #define RVC_LW_IMM(x)
- #define RVC_LD_IMM(x)
- #define RVC_LWSP_IMM(x)
- #define RVC_LDSP_IMM(x)
- #define RVC_SWSP_IMM(x)
- #define RVC_SDSP_IMM(x)
- #define RVC_RS1S(insn) (8 + RV_X(insn, SH_RD, 3))
- #define RVC_RS2S(insn) (8 + RV_X(insn, SH_RS2C, 3))
- #define RVC_RS2(insn) RV_X(insn, SH_RS2C, 5)
- #define SHIFT_RIGHT(x, y) ((y) < 0 ? ((x) << -(y)) : ((x) >> (y)))
- #define REG_MASK ((1 << (5 + LOG_REGBYTES)) - (1 << LOG_REGBYTES))
- #define REG_OFFSET(insn, pos) (SHIFT_RIGHT((insn), (pos) - LOG_REGBYTES) & REG_MASK)
- #define REG_PTR(insn, pos, regs) (ulong *)((ulong)(regs) + REG_OFFSET(insn, pos))
- #define GET_RM(insn) ((insn & MASK_FUNCT3) >> SHIFT_FUNCT3)
- #define GET_RS1_NUM(insn) ((insn & MASK_RS1) >> 15)
- #define GET_CSR_NUM(insn) ((insn & MASK_CSR) >> SHIFT_CSR)
- #define GET_RS1(insn, regs) (*REG_PTR(insn, SH_RS1, regs))
- #define GET_RS2(insn, regs) (*REG_PTR(insn, SH_RS2, regs))
- #define GET_RS1S(insn, regs) (*REG_PTR(RVC_RS1S(insn), 0, regs))
- #define GET_RS2S(insn, regs) (*REG_PTR(RVC_RS2S(insn), 0, regs))
- #define GET_RS2C(insn, regs) (*REG_PTR(insn, SH_RS2C, regs))
- #define GET_SP(regs) (*REG_PTR(2, 0, regs))
- #define SET_RD(insn, regs, val) (*REG_PTR(insn, SH_RD, regs) = (val))
- #define IMM_I(insn) ((s32)(insn) >> 20)
- #define IMM_S(insn)
- #define IS_MASKED(insn) (((insn >> SH_VM) & VM_MASK) == 0)
- #define GET_VD(insn) ((insn >> SH_VD) & VD_MASK)
- #define GET_VS2(insn) ((insn >> SH_VS2) & VS2_MASK)
- #define GET_VIEW(insn) (((insn) >> SH_VIEW) & VIEW_MASK)
- #define GET_MEW(insn) (((insn) >> SH_MEW) & 1)
- #define GET_VSEW(vtype) (((vtype) >> SH_VSEW) & VSEW_MASK)
- #define GET_VLMUL(vtype) ((vtype) & VLMUL_MASK)
- #define GET_LEN(view) (1UL << (view))
- #define GET_NF(insn) (1 + ((insn >> 29) & 7))
- #define GET_VEMUL(vlmul, view, vsew) ((vlmul + view - vsew) & 7)
- #define GET_EMUL(vemul) (1UL << ((vemul) >= 4 ? 0 : (vemul)))
- #define MASK_FUNCT3 0x7000
- #define MASK_RS1 0xf8000
- #define MASK_CSR 0xfff00000
- #define SHIFT_FUNCT3 12
- #define SHIFT_CSR 20
- #define CSRRW 1
- #define CSRRS 2
- #define CSRRC 3
- #define CSRRWI 5
- #define CSRRSI 6
- #define CSRRCI 7

30.38.1 Macro Definition Documentation

30.38.1.1 CAUSE_BREAKPOINT

```
#define CAUSE_BREAKPOINT 0x3
```

30.38.1.2 CAUSE_DOUBLE_TRAP

```
#define CAUSE_DOUBLE_TRAP 0x10
```

30.38.1.3 CAUSE_FETCH_ACCESS

```
#define CAUSE_FETCH_ACCESS 0x1
```

30.38.1.4 CAUSE_FETCH_GUEST_PAGE_FAULT

```
#define CAUSE_FETCH_GUEST_PAGE_FAULT 0x14
```

30.38.1.5 CAUSE_FETCH_PAGE_FAULT

```
#define CAUSE_FETCH_PAGE_FAULT 0xc
```

30.38.1.6 CAUSE_ILLEGAL_INSTRUCTION

```
#define CAUSE_ILLEGAL_INSTRUCTION 0x2
```

30.38.1.7 CAUSE_LOAD_ACCESS

```
#define CAUSE_LOAD_ACCESS 0x5
```

30.38.1.8 CAUSE_LOAD_GUEST_PAGE_FAULT

```
#define CAUSE_LOAD_GUEST_PAGE_FAULT 0x15
```

30.38.1.9 CAUSE_LOAD_PAGE_FAULT

```
#define CAUSE_LOAD_PAGE_FAULT 0xd
```

30.38.1.10 CAUSE_MACHINE_ECALL

```
#define CAUSE_MACHINE_ECALL 0xb
```

30.38.1.11 CAUSE_MISALIGNED_FETCH

```
#define CAUSE_MISALIGNED_FETCH 0x0
```

30.38.1.12 CAUSE_MISALIGNED_LOAD

```
#define CAUSE_MISALIGNED_LOAD 0x4
```

30.38.1.13 CAUSE_MISALIGNED_STORE

```
#define CAUSE_MISALIGNED_STORE 0x6
```

30.38.1.14 CAUSE_STORE_ACCESS

```
#define CAUSE_STORE_ACCESS 0x7
```

30.38.1.15 CAUSE_STORE_GUEST_PAGE_FAULT

```
#define CAUSE_STORE_GUEST_PAGE_FAULT 0x17
```

30.38.1.16 CAUSE_STORE_PAGE_FAULT

```
#define CAUSE_STORE_PAGE_FAULT 0xf
```

30.38.1.17 CAUSE_SUPERVISOR_ECALL

```
#define CAUSE_SUPERVISOR_ECALL 0x9
```

30.38.1.18 CAUSE_SW_CHECK_EXCP

```
#define CAUSE_SW_CHECK_EXCP 0x12
```

30.38.1.19 CAUSE_USER_ECALL

```
#define CAUSE_USER_ECALL 0x8
```

30.38.1.20 CAUSE_VIRTUAL_INST_FAULT

```
#define CAUSE_VIRTUAL_INST_FAULT 0x16
```

30.38.1.21 CAUSE_VIRTUAL_SUPERVISOR_ECALL

```
#define CAUSE_VIRTUAL_SUPERVISOR_ECALL 0xa
```

30.38.1.22 CSR_CYCLE

```
#define CSR_CYCLE 0xc00
```

30.38.1.23 CSR_CYCLEH

```
#define CSR_CYCLEH 0xc80
```

30.38.1.24 CSR_DCSR

```
#define CSR_DCSR 0x7b0
```

30.38.1.25 CSR_DPC

```
#define CSR_DPC 0x7b1
```

30.38.1.26 CSR_DSCRATCH0

```
#define CSR_DSCRATCH0 0x7b2
```

30.38.1.27 CSR_DSCRATCH1

```
#define CSR_DSCRATCH1 0x7b3
```

30.38.1.28 CSR_FCSR

```
#define CSR_FCSR 0x003
```

30.38.1.29 CSR_FFLAGS

```
#define CSR_FFLAGS 0x001
```

30.38.1.30 CSR_FRM

```
#define CSR_FRM 0x002
```

30.38.1.31 CSR_HCOUNTEREN

```
#define CSR_HCOUNTEREN 0x606
```

30.38.1.32 CSR_HEDELEG

```
#define CSR_HEDELEG 0x602
```

30.38.1.33 CSR_HENVCFG

```
#define CSR_HENVCFG 0x60a
```

30.38.1.34 CSR_HENVCFGH

```
#define CSR_HENVCFGH 0x61a
```

30.38.1.35 CSR_HGATP

```
#define CSR_HGATP 0x680
```

30.38.1.36 CSR_HGEIE

```
#define CSR_HGEIE 0x607
```

30.38.1.37 CSR_HGEIP

```
#define CSR_HGEIP 0xe12
```

30.38.1.38 CSR_HIDELEG

```
#define CSR_HIDELEG 0x603
```

30.38.1.39 CSR_HIDELEGH

```
#define CSR_HIDELEGH 0x613
```

30.38.1.40 CSR_HIE

```
#define CSR_HIE 0x604
```

30.38.1.41 CSR_HIP

```
#define CSR_HIP 0x644
```

30.38.1.42 CSR_HPMCOUNTER10

```
#define CSR_HPMCOUNTER10 0xc0a
```

30.38.1.43 CSR_HPMCOUNTER10H

```
#define CSR_HPMCOUNTER10H 0xc8a
```

30.38.1.44 CSR_HPMCOUNTER11

```
#define CSR_HPMCOUNTER11 0xc0b
```

30.38.1.45 CSR_HPMCOUNTER11H

```
#define CSR_HPMCOUNTER11H 0xc8b
```

30.38.1.46 CSR_HPMCOUNTER12

```
#define CSR_HPMCOUNTER12 0xc0c
```

30.38.1.47 CSR_HPMCOUNTER12H

```
#define CSR_HPMCOUNTER12H 0xc8c
```

30.38.1.48 CSR_HPMCOUNTER13

```
#define CSR_HPMCOUNTER13 0xc0d
```

30.38.1.49 CSR_HPMCOUNTER13H

```
#define CSR_HPMCOUNTER13H 0xc8d
```

30.38.1.50 CSR_HPMCOUNTER14

```
#define CSR_HPMCOUNTER14 0xc0e
```

30.38.1.51 CSR_HPMCOUNTER14H

```
#define CSR_HPMCOUNTER14H 0xc8e
```

30.38.1.52 CSR_HPMCOUNTER15

```
#define CSR_HPMCOUNTER15 0xc0f
```

30.38.1.53 CSR_HPMCOUNTER15H

```
#define CSR_HPMCOUNTER15H 0xc8f
```

30.38.1.54 CSR_HPMCOUNTER16

```
#define CSR_HPMCOUNTER16 0xc10
```

30.38.1.55 CSR_HPMCOUNTER16H

```
#define CSR_HPMCOUNTER16H 0xc90
```

30.38.1.56 CSR_HPMCOUNTER17

```
#define CSR_HPMCOUNTER17 0xc11
```

30.38.1.57 CSR_HPMCOUNTER17H

```
#define CSR_HPMCOUNTER17H 0xc91
```

30.38.1.58 CSR_HPMCOUNTER18

```
#define CSR_HPMCOUNTER18 0xc12
```

30.38.1.59 CSR_HPMCOUNTER18H

```
#define CSR_HPMCOUNTER18H 0xc92
```

30.38.1.60 CSR_HPMCOUNTER19

```
#define CSR_HPMCOUNTER19 0xc13
```

30.38.1.61 CSR_HPMCOUNTER19H

```
#define CSR_HPMCOUNTER19H 0xc93
```

30.38.1.62 CSR_HPMCOUNTER20

```
#define CSR_HPMCOUNTER20 0xc14
```

30.38.1.63 CSR_HPMCOUNTER20H

```
#define CSR_HPMCOUNTER20H 0xc94
```

30.38.1.64 CSR_HPMCOUNTER21

```
#define CSR_HPMCOUNTER21 0xc15
```

30.38.1.65 CSR_HPMCOUNTER21H

```
#define CSR_HPMCOUNTER21H 0xc95
```

30.38.1.66 CSR_HPMCOUNTER22

```
#define CSR_HPMCOUNTER22 0xc16
```

30.38.1.67 CSR_HPMCOUNTER22H

```
#define CSR_HPMCOUNTER22H 0xc96
```

30.38.1.68 CSR_HPMCOUNTER23

```
#define CSR_HPMCOUNTER23 0xc17
```

30.38.1.69 CSR_HPMCOUNTER23H

```
#define CSR_HPMCOUNTER23H 0xc97
```

30.38.1.70 CSR_HPMCOUNTER24

```
#define CSR_HPMCOUNTER24 0xc18
```

30.38.1.71 CSR_HPMCOUNTER24H

```
#define CSR_HPMCOUNTER24H 0xc98
```

30.38.1.72 CSR_HPMCOUNTER25

```
#define CSR_HPMCOUNTER25 0xc19
```

30.38.1.73 CSR_HPMCOUNTER25H

```
#define CSR_HPMCOUNTER25H 0xc99
```

30.38.1.74 CSR_HPMCOUNTER26

```
#define CSR_HPMCOUNTER26 0xc1a
```

30.38.1.75 CSR_HPMCOUNTER26H

```
#define CSR_HPMCOUNTER26H 0xc9a
```

30.38.1.76 CSR_HPMCOUNTER27

```
#define CSR_HPMCOUNTER27 0xc1b
```

30.38.1.77 CSR_HPMCOUNTER27H

```
#define CSR_HPMCOUNTER27H 0xc9b
```

30.38.1.78 CSR_HPMCOUNTER28

```
#define CSR_HPMCOUNTER28 0xc1c
```

30.38.1.79 CSR_HPMCOUNTER28H

```
#define CSR_HPMCOUNTER28H 0xc9c
```

30.38.1.80 CSR_HPMCOUNTER29

```
#define CSR_HPMCOUNTER29 0xc1d
```

30.38.1.81 CSR_HPMCOUNTER29H

```
#define CSR_HPMCOUNTER29H 0xc9d
```

30.38.1.82 CSR_HPMCOUNTER3

```
#define CSR_HPMCOUNTER3 0xc03
```

30.38.1.83 CSR_HPMCOUNTER30

```
#define CSR_HPMCOUNTER30 0xc1e
```

30.38.1.84 CSR_HPMCOUNTER30H

```
#define CSR_HPMCOUNTER30H 0xc9e
```

30.38.1.85 CSR_HPMCOUNTER31

```
#define CSR_HPMCOUNTER31 0xc1f
```

30.38.1.86 CSR_HPMCOUNTER31H

```
#define CSR_HPMCOUNTER31H 0xc9f
```

30.38.1.87 CSR_HPMCOUNTER3H

```
#define CSR_HPMCOUNTER3H 0xc83
```

30.38.1.88 CSR_HPMCOUNTER4

```
#define CSR_HPMCOUNTER4 0xc04
```

30.38.1.89 CSR_HPMCOUNTER4H

```
#define CSR_HPMCOUNTER4H 0xc84
```

30.38.1.90 CSR_HPMCOUNTER5

```
#define CSR_HPMCOUNTER5 0xc05
```

30.38.1.91 CSR_HPMCOUNTER5H

```
#define CSR_HPMCOUNTER5H 0xc85
```

30.38.1.92 CSR_HPMCOUNTER6

```
#define CSR_HPMCOUNTER6 0xc06
```

30.38.1.93 CSR_HPMCOUNTER6H

```
#define CSR_HPMCOUNTER6H 0xc86
```

30.38.1.94 CSR_HPMCOUNTER7

```
#define CSR_HPMCOUNTER7 0xc07
```

30.38.1.95 CSR_HPMCOUNTER7H

```
#define CSR_HPMCOUNTER7H 0xc87
```

30.38.1.96 CSR_HPMCOUNTER8

```
#define CSR_HPMCOUNTER8 0xc08
```

30.38.1.97 CSR_HPMCOUNTER8H

```
#define CSR_HPMCOUNTER8H 0xc88
```

30.38.1.98 CSR_HPMCOUNTER9

```
#define CSR_HPMCOUNTER9 0xc09
```

30.38.1.99 CSR_HPMCOUNTER9H

```
#define CSR_HPMCOUNTER9H 0xc89
```

30.38.1.100 CSR_HSTATEEN0

```
#define CSR_HSTATEEN0 0x60C
```

30.38.1.101 CSR_HSTATEEN0H

```
#define CSR_HSTATEEN0H 0x61C
```

30.38.1.102 CSR_HSTATEEN1

```
#define CSR_HSTATEEN1 0x60D
```

30.38.1.103 CSR_HSTATEEN1H

```
#define CSR_HSTATEEN1H 0x61D
```

30.38.1.104 CSR_HSTATEEN2

```
#define CSR_HSTATEEN2 0x60E
```

30.38.1.105 CSR_HSTATEEN2H

```
#define CSR_HSTATEEN2H 0x61E
```

30.38.1.106 CSR_HSTATEEN3

```
#define CSR_HSTATEEN3 0x60F
```

30.38.1.107 CSR_HSTATEEN3H

```
#define CSR_HSTATEEN3H 0x61F
```

30.38.1.108 CSR_HSTATUS

```
#define CSR_HSTATUS 0x600
```

30.38.1.109 CSR_HTIMEDELTA

```
#define CSR_HTIMEDELTA 0x605
```

30.38.1.110 CSR_HTIMEDELTAH

```
#define CSR_HTIMEDELTAH 0x615
```

30.38.1.111 CSR_HTINST

```
#define CSR_HTINST 0x64a
```

30.38.1.112 CSR_HTVAL

```
#define CSR_HTVAL 0x643
```

30.38.1.113 CSR_HVICTL

```
#define CSR_HVICTL 0x609
```

30.38.1.114 CSR_HVIEN

```
#define CSR_HVIEN 0x608
```

30.38.1.115 CSR_HVIENH

```
#define CSR_HVIENH 0x618
```

30.38.1.116 CSR_HVIP

```
#define CSR_HVIP 0x645
```

30.38.1.117 CSR_HVIPH

```
#define CSR_HVIPH 0x655
```

30.38.1.118 CSR_HVIPRIO1

```
#define CSR_HVIPRIO1 0x646
```

30.38.1.119 CSR_HVIPRIO1H

```
#define CSR_HVIPRIO1H 0x656
```

30.38.1.120 CSR_HVIPRIO2

```
#define CSR_HVIPRIO2 0x647
```

30.38.1.121 CSR_HVIPRIO2H

```
#define CSR_HVIPRIO2H 0x657
```

30.38.1.122 CSR_INSTRET

```
#define CSR_INSTRET 0xc02
```

30.38.1.123 CSR_INSTRETH

```
#define CSR_INSTRETH 0xc82
```

30.38.1.124 CSR_MARCHID

```
#define CSR_MARCHID 0xf12
```

30.38.1.125 CSR_MCAUSE

```
#define CSR_MCAUSE 0x342
```

30.38.1.126 CSR_MCONFIGPTR

```
#define CSR_MCONFIGPTR 0xf15
```

30.38.1.127 CSR_MCOUNTEREN

```
#define CSR_MCOUNTEREN 0x306
```

30.38.1.128 CSR_MCOUNTINHIBIT

```
#define CSR_MCOUNTINHIBIT 0x320
```

30.38.1.129 CSR_MCYCLE

```
#define CSR_MCYCLE 0xb00
```

30.38.1.130 CSR_MCYCLECFG

```
#define CSR_MCYCLECFG 0x321
```

30.38.1.131 CSR_MCYCLECFGH

```
#define CSR_MCYCLECFGH 0x721
```

30.38.1.132 CSR_MCYCLEH

```
#define CSR_MCYCLEH 0xb80
```

30.38.1.133 CSR_MEDELEG

```
#define CSR_MEDELEG 0x302
```

30.38.1.134 CSR_MENVCFG

```
#define CSR_MENVCFG 0x30a
```

30.38.1.135 CSR_MENVCFGH

```
#define CSR_MENVCFGH 0x31a
```

30.38.1.136 CSR_MEPC

```
#define CSR_MEPC 0x341
```

30.38.1.137 CSR_MHARTID

```
#define CSR_MHARTID 0xf14
```

30.38.1.138 CSR_MHPMCOUNTER10

```
#define CSR_MHPMCOUNTER10 0xb0a
```

30.38.1.139 CSR_MHPMCOUNTER10H

```
#define CSR_MHPMCOUNTER10H 0xb8a
```

30.38.1.140 CSR_MHPMCOUNTER11

```
#define CSR_MHPMCOUNTER11 0xb0b
```

30.38.1.141 CSR_MHPMCOUNTER11H

```
#define CSR_MHPMCOUNTER11H 0xb8b
```

30.38.1.142 CSR_MHPMCOUNTER12

```
#define CSR_MHPMCOUNTER12 0xb0c
```

30.38.1.143 CSR_MHPMCOUNTER12H

```
#define CSR_MHPMCOUNTER12H 0xb8c
```

30.38.1.144 CSR_MHPMCOUNTER13

```
#define CSR_MHPMCOUNTER13 0xb0d
```

30.38.1.145 CSR_MHPMCOUNTER13H

```
#define CSR_MHPMCOUNTER13H 0xb8d
```

30.38.1.146 CSR_MHPMCOUNTER14

```
#define CSR_MHPMCOUNTER14 0xb0e
```

30.38.1.147 CSR_MHPMCOUNTER14H

```
#define CSR_MHPMCOUNTER14H 0xb8e
```

30.38.1.148 CSR_MHPMCOUNTER15

```
#define CSR_MHPMCOUNTER15 0xb0f
```

30.38.1.149 CSR_MHPMCOUNTER15H

```
#define CSR_MHPMCOUNTER15H 0xb8f
```

30.38.1.150 CSR_MHPMCOUNTER16

```
#define CSR_MHPMCOUNTER16 0xb10
```

30.38.1.151 CSR_MHPMCOUNTER16H

```
#define CSR_MHPMCOUNTER16H 0xb90
```

30.38.1.152 CSR_MHPMCOUNTER17

```
#define CSR_MHPMCOUNTER17 0xb11
```

30.38.1.153 CSR_MHPMCOUNTER17H

```
#define CSR_MHPMCOUNTER17H 0xb91
```

30.38.1.154 CSR_MHPMCOUNTER18

```
#define CSR_MHPMCOUNTER18 0xb12
```

30.38.1.155 CSR_MHPMCOUNTER18H

```
#define CSR_MHPMCOUNTER18H 0xb92
```

30.38.1.156 CSR_MHPMCOUNTER19

```
#define CSR_MHPMCOUNTER19 0xb13
```

30.38.1.157 CSR_MHPMCOUNTER19H

```
#define CSR_MHPMCOUNTER19H 0xb93
```

30.38.1.158 CSR_MHPMCOUNTER20

```
#define CSR_MHPMCOUNTER20 0xb14
```

30.38.1.159 CSR_MHPMCOUNTER20H

```
#define CSR_MHPMCOUNTER20H 0xb94
```

30.38.1.160 CSR_MHPMCOUNTER21

```
#define CSR_MHPMCOUNTER21 0xb15
```

30.38.1.161 CSR_MHPMCOUNTER21H

```
#define CSR_MHPMCOUNTER21H 0xb95
```

30.38.1.162 CSR_MHPMCOUNTER22

```
#define CSR_MHPMCOUNTER22 0xb16
```

30.38.1.163 CSR_MHPMCOUNTER22H

```
#define CSR_MHPMCOUNTER22H 0xb96
```

30.38.1.164 CSR_MHPMCOUNTER23

```
#define CSR_MHPMCOUNTER23 0xb17
```

30.38.1.165 CSR_MHPMCOUNTER23H

```
#define CSR_MHPMCOUNTER23H 0xb97
```

30.38.1.166 CSR_MHPMCOUNTER24

```
#define CSR_MHPMCOUNTER24 0xb18
```

30.38.1.167 CSR_MHPMCOUNTER24H

```
#define CSR_MHPMCOUNTER24H 0xb98
```

30.38.1.168 CSR_MHPMCOUNTER25

```
#define CSR_MHPMCOUNTER25 0xb19
```

30.38.1.169 CSR_MHPMCOUNTER25H

```
#define CSR_MHPMCOUNTER25H 0xb99
```

30.38.1.170 CSR_MHPMCOUNTER26

```
#define CSR_MHPMCOUNTER26 0xb1a
```

30.38.1.171 CSR_MHPMCOUNTER26H

```
#define CSR_MHPMCOUNTER26H 0xb9a
```

30.38.1.172 CSR_MHPMCOUNTER27

```
#define CSR_MHPMCOUNTER27 0xb1b
```

30.38.1.173 CSR_MHPMCOUNTER27H

```
#define CSR_MHPMCOUNTER27H 0xb9b
```

30.38.1.174 CSR_MHPMCOUNTER28

```
#define CSR_MHPMCOUNTER28 0xb1c
```

30.38.1.175 CSR_MHPMCOUNTER28H

```
#define CSR_MHPMCOUNTER28H 0xb9c
```

30.38.1.176 CSR_MHPMCOUNTER29

```
#define CSR_MHPMCOUNTER29 0xb1d
```

30.38.1.177 CSR_MHPMCOUNTER29H

```
#define CSR_MHPMCOUNTER29H 0xb9d
```

30.38.1.178 CSR_MHPMCOUNTER3

```
#define CSR_MHPMCOUNTER3 0xb03
```

30.38.1.179 CSR_MHPMCOUNTER30

```
#define CSR_MHPMCOUNTER30 0xb1e
```

30.38.1.180 CSR_MHPMCOUNTER30H

```
#define CSR_MHPMCOUNTER30H 0xb9e
```

30.38.1.181 CSR_MHPMCOUNTER31

```
#define CSR_MHPMCOUNTER31 0xb1f
```

30.38.1.182 CSR_MHPMCOUNTER31H

```
#define CSR_MHPMCOUNTER31H 0xb9f
```

30.38.1.183 CSR_MHPMCOUNTER3H

```
#define CSR_MHPMCOUNTER3H 0xb83
```

30.38.1.184 CSR_MHPMCOUNTER4

```
#define CSR_MHPMCOUNTER4 0xb04
```

30.38.1.185 CSR_MHPMCOUNTER4H

```
#define CSR_MHPMCOUNTER4H 0xb84
```

30.38.1.186 CSR_MHPMCOUNTER5

```
#define CSR_MHPMCOUNTER5 0xb05
```

30.38.1.187 CSR_MHPMCOUNTER5H

```
#define CSR_MHPMCOUNTER5H 0xb85
```

30.38.1.188 CSR_MHPMCOUNTER6

```
#define CSR_MHPMCOUNTER6 0xb06
```

30.38.1.189 CSR_MHPMCOUNTER6H

```
#define CSR_MHPMCOUNTER6H 0xb86
```

30.38.1.190 CSR_MHPMCOUNTER7

```
#define CSR_MHPMCOUNTER7 0xb07
```

30.38.1.191 CSR_MHPMCOUNTER7H

```
#define CSR_MHPMCOUNTER7H 0xb87
```

30.38.1.192 CSR_MHPMCOUNTER8

```
#define CSR_MHPMCOUNTER8 0xb08
```

30.38.1.193 CSR_MHPMCOUNTER8H

```
#define CSR_MHPMCOUNTER8H 0xb88
```

30.38.1.194 CSR_MHPMCOUNTER9

```
#define CSR_MHPMCOUNTER9 0xb09
```

30.38.1.195 CSR_MHPMCOUNTER9H

```
#define CSR_MHPMCOUNTER9H 0xb89
```

30.38.1.196 CSR_MHPMEVENT10

```
#define CSR_MHPMEVENT10 0x32a
```

30.38.1.197 CSR_MHPMEVENT10H

```
#define CSR_MHPMEVENT10H 0x72a
```

30.38.1.198 CSR_MHPMEVENT11

```
#define CSR_MHPMEVENT11 0x32b
```

30.38.1.199 CSR_MHPMEVENT11H

```
#define CSR_MHPMEVENT11H 0x72b
```

30.38.1.200 CSR_MHPMEVENT12

```
#define CSR_MHPMEVENT12 0x32c
```

30.38.1.201 CSR_MHPMEVENT12H

```
#define CSR_MHPMEVENT12H 0x72c
```

30.38.1.202 CSR_MHPMEVENT13

```
#define CSR_MHPMEVENT13 0x32d
```

30.38.1.203 CSR_MHPMEVENT13H

```
#define CSR_MHPMEVENT13H 0x72d
```

30.38.1.204 CSR_MHPMEVENT14

```
#define CSR_MHPMEVENT14 0x32e
```

30.38.1.205 CSR_MHPMEVENT14H

```
#define CSR_MHPMEVENT14H 0x72e
```

30.38.1.206 CSR_MHPMEVENT15

```
#define CSR_MHPMEVENT15 0x32f
```

30.38.1.207 CSR_MHPMEVENT15H

```
#define CSR_MHPMEVENT15H 0x72f
```

30.38.1.208 CSR_MHPMEVENT16

```
#define CSR_MHPMEVENT16 0x330
```

30.38.1.209 CSR_MHPMEVENT16H

```
#define CSR_MHPMEVENT16H 0x730
```

30.38.1.210 CSR_MHPMEVENT17

```
#define CSR_MHPMEVENT17 0x331
```

30.38.1.211 CSR_MHPMEVENT17H

```
#define CSR_MHPMEVENT17H 0x731
```

30.38.1.212 CSR_MHPMEVENT18

```
#define CSR_MHPMEVENT18 0x332
```

30.38.1.213 CSR_MHPMEVENT18H

```
#define CSR_MHPMEVENT18H 0x732
```

30.38.1.214 CSR_MHPMEVENT19

```
#define CSR_MHPMEVENT19 0x333
```

30.38.1.215 CSR_MHPMEVENT19H

```
#define CSR_MHPMEVENT19H 0x733
```

30.38.1.216 CSR_MHPMEVENT20

```
#define CSR_MHPMEVENT20 0x334
```

30.38.1.217 CSR_MHPMEVENT20H

```
#define CSR_MHPMEVENT20H 0x734
```

30.38.1.218 CSR_MHPMEVENT21

```
#define CSR_MHPMEVENT21 0x335
```

30.38.1.219 CSR_MHPMEVENT21H

```
#define CSR_MHPMEVENT21H 0x735
```

30.38.1.220 CSR_MHPMEVENT22

```
#define CSR_MHPMEVENT22 0x336
```

30.38.1.221 CSR_MHPMEVENT22H

```
#define CSR_MHPMEVENT22H 0x736
```

30.38.1.222 CSR_MHPMEVENT23

```
#define CSR_MHPMEVENT23 0x337
```

30.38.1.223 CSR_MHPMEVENT23H

```
#define CSR_MHPMEVENT23H 0x737
```

30.38.1.224 CSR_MHPMEVENT24

```
#define CSR_MHPMEVENT24 0x338
```

30.38.1.225 CSR_MHPMEVENT24H

```
#define CSR_MHPMEVENT24H 0x738
```

30.38.1.226 CSR_MHPMEVENT25

```
#define CSR_MHPMEVENT25 0x339
```

30.38.1.227 CSR_MHPMEVENT25H

```
#define CSR_MHPMEVENT25H 0x739
```

30.38.1.228 CSR_MHPMEVENT26

```
#define CSR_MHPMEVENT26 0x33a
```

30.38.1.229 CSR_MHPMEVENT26H

```
#define CSR_MHPMEVENT26H 0x73a
```

30.38.1.230 CSR_MHPMEVENT27

```
#define CSR_MHPMEVENT27 0x33b
```

30.38.1.231 CSR_MHPMEVENT27H

```
#define CSR_MHPMEVENT27H 0x73b
```

30.38.1.232 CSR_MHPMEVENT28

```
#define CSR_MHPMEVENT28 0x33c
```

30.38.1.233 CSR_MHPMEVENT28H

```
#define CSR_MHPMEVENT28H 0x73c
```

30.38.1.234 CSR_MHPMEVENT29

```
#define CSR_MHPMEVENT29 0x33d
```

30.38.1.235 CSR_MHPMEVENT29H

```
#define CSR_MHPMEVENT29H 0x73d
```

30.38.1.236 CSR_MHPMEVENT3

```
#define CSR_MHPMEVENT3 0x323
```

30.38.1.237 CSR_MHPMEVENT30

```
#define CSR_MHPMEVENT30 0x33e
```

30.38.1.238 CSR_MHPMEVENT30H

```
#define CSR_MHPMEVENT30H 0x73e
```

30.38.1.239 CSR_MHPMEVENT31

```
#define CSR_MHPMEVENT31 0x33f
```

30.38.1.240 CSR_MHPMEVENT31H

```
#define CSR_MHPMEVENT31H 0x73f
```

30.38.1.241 CSR_MHPMEVENT3H

```
#define CSR_MHPMEVENT3H 0x723
```

30.38.1.242 CSR_MHPMEVENT4

```
#define CSR_MHPMEVENT4 0x324
```

30.38.1.243 CSR_MHPMEVENT4H

```
#define CSR_MHPMEVENT4H 0x724
```

30.38.1.244 CSR_MHPMEVENT5

```
#define CSR_MHPMEVENT5 0x325
```

30.38.1.245 CSR_MHPMEVENT5H

```
#define CSR_MHPMEVENT5H 0x725
```

30.38.1.246 CSR_MHPMEVENT6

```
#define CSR_MHPMEVENT6 0x326
```

30.38.1.247 CSR_MHPMEVENT6H

```
#define CSR_MHPMEVENT6H 0x726
```

30.38.1.248 CSR_MHPMEVENT7

```
#define CSR_MHPMEVENT7 0x327
```

30.38.1.249 CSR_MHPMEVENT7H

```
#define CSR_MHPMEVENT7H 0x727
```

30.38.1.250 CSR_MHPMEVENT8

```
#define CSR_MHPMEVENT8 0x328
```

30.38.1.251 CSR_MHPMEVENT8H

```
#define CSR_MHPMEVENT8H 0x728
```

30.38.1.252 CSR_MHPMEVENT9

```
#define CSR_MHPMEVENT9 0x329
```

30.38.1.253 CSR_MHPMEVENT9H

```
#define CSR_MHPMEVENT9H 0x729
```

30.38.1.254 CSR_MIDELEG

```
#define CSR_MIDELEG 0x303
```

30.38.1.255 CSR_MIDELEGH

```
#define CSR_MIDELEGH 0x313
```

30.38.1.256 CSR_MIE

```
#define CSR_MIE 0x304
```

30.38.1.257 CSR_MIEH

```
#define CSR_MIEH 0x314
```

30.38.1.258 CSR_MIMPID

```
#define CSR_MIMPID 0xf13
```

30.38.1.259 CSR_MINSTRET

```
#define CSR_MINSTRET 0xb02
```

30.38.1.260 CSR_MINSTRETCFG

```
#define CSR_MINSTRETCFG 0x322
```

30.38.1.261 CSR_MINSTRETCFGH

```
#define CSR_MINSTRETCFGH 0x722
```

30.38.1.262 CSR_MINSTRETH

```
#define CSR_MINSTRETH 0xb82
```

30.38.1.263 CSR_MIP

```
#define CSR_MIP 0x344
```

30.38.1.264 CSR_MIPH

```
#define CSR_MIPH 0x354
```

30.38.1.265 CSR_MIREG

```
#define CSR_MIREG 0x351
```

30.38.1.266 CSR_MIREG2

```
#define CSR_MIREG2 0x352
```

30.38.1.267 CSR_MIREG3

```
#define CSR_MIREG3 0x353
```

30.38.1.268 CSR_MIREG4

```
#define CSR_MIREG4 0x355
```

30.38.1.269 CSR_MIREG5

```
#define CSR_MIREG5 0x356
```

30.38.1.270 CSR_MIREG6

```
#define CSR_MIREG6 0x357
```

30.38.1.271 CSR_MISA

```
#define CSR_MISA 0x301
```

30.38.1.272 CSR_MISELECT

```
#define CSR_MISELECT 0x350
```

30.38.1.273 CSR_MSCRATCH

```
#define CSR_MSCRATCH 0x340
```

30.38.1.274 CSR_MSECCFG

```
#define CSR_MSECCFG 0x747
```

30.38.1.275 CSR_MSECCFGH

```
#define CSR_MSECCFGH 0x757
```

30.38.1.276 CSR_MSTATEEN0

```
#define CSR_MSTATEEN0 0x30C
```

30.38.1.277 CSR_MSTATEEN0H

```
#define CSR_MSTATEEN0H 0x31C
```

30.38.1.278 CSR_MSTATEEN1

```
#define CSR_MSTATEEN1 0x30D
```

30.38.1.279 CSR_MSTATEEN1H

```
#define CSR_MSTATEEN1H 0x31D
```

30.38.1.280 CSR_MSTATEEN2

```
#define CSR_MSTATEEN2 0x30E
```

30.38.1.281 CSR_MSTATEEN2H

```
#define CSR_MSTATEEN2H 0x31E
```

30.38.1.282 CSR_MSTATEEN3

```
#define CSR_MSTATEEN3 0x30F
```

30.38.1.283 CSR_MSTATEEN3H

```
#define CSR_MSTATEEN3H 0x31F
```

30.38.1.284 CSR_MSTATUS

```
#define CSR_MSTATUS 0x300
```

30.38.1.285 CSR_MSTATUSH

```
#define CSR_MSTATUSH 0x310
```

30.38.1.286 CSR_MTINST

```
#define CSR_MTINST 0x34a
```

30.38.1.287 CSR_MTOPEI

```
#define CSR_MTOPEI 0x35c
```

30.38.1.288 CSR_MTOPI

```
#define CSR_MTOPI 0xfb0
```

30.38.1.289 CSR_MTVAL

```
#define CSR_MTVAL 0x343
```

30.38.1.290 CSR_MTVAL2

```
#define CSR_MTVAL2 0x34b
```

30.38.1.291 CSR_MTVEC

```
#define CSR_MTVEC 0x305
```

30.38.1.292 CSR_MVENDORID

```
#define CSR_MVENDORID 0xf11
```

30.38.1.293 CSR_MVIEN

```
#define CSR_MVIEN 0x308
```

30.38.1.294 CSR_MVIENH

```
#define CSR_MVIENH 0x318
```

30.38.1.295 CSR_MVIP

```
#define CSR_MVIP 0x309
```

30.38.1.296 CSR_MVIPH

```
#define CSR_MVIPH 0x319
```

30.38.1.297 CSR_PMPADDR0

```
#define CSR_PMPADDR0 0x3b0
```

30.38.1.298 CSR_PMPADDR1

```
#define CSR_PMPADDR1 0x3b1
```

30.38.1.299 CSR_PMPADDR10

```
#define CSR_PMPADDR10 0x3ba
```

30.38.1.300 CSR_PMPADDR11

```
#define CSR_PMPADDR11 0x3bb
```

30.38.1.301 CSR_PMPADDR12

```
#define CSR_PMPADDR12 0x3bc
```

30.38.1.302 CSR_PMPADDR13

```
#define CSR_PMPADDR13 0x3bd
```

30.38.1.303 CSR_PMPADDR14

```
#define CSR_PMPADDR14 0x3be
```

30.38.1.304 CSR_PMPADDR15

```
#define CSR_PMPADDR15 0x3bf
```

30.38.1.305 CSR_PMPADDR16

```
#define CSR_PMPADDR16 0x3c0
```

30.38.1.306 CSR_PMPADDR17

```
#define CSR_PMPADDR17 0x3c1
```

30.38.1.307 CSR_PMPADDR18

```
#define CSR_PMPADDR18 0x3c2
```

30.38.1.308 CSR_PMPADDR19

```
#define CSR_PMPADDR19 0x3c3
```

30.38.1.309 CSR_PMPADDR2

```
#define CSR_PMPADDR2 0x3b2
```

30.38.1.310 CSR_PMPADDR20

```
#define CSR_PMPADDR20 0x3c4
```

30.38.1.311 CSR_PMPADDR21

```
#define CSR_PMPADDR21 0x3c5
```

30.38.1.312 CSR_PMPADDR22

```
#define CSR_PMPADDR22 0x3c6
```

30.38.1.313 CSR_PMPADDR23

```
#define CSR_PMPADDR23 0x3c7
```

30.38.1.314 CSR_PMPADDR24

```
#define CSR_PMPADDR24 0x3c8
```

30.38.1.315 CSR_PMPADDR25

```
#define CSR_PMPADDR25 0x3c9
```

30.38.1.316 CSR_PMPADDR26

```
#define CSR_PMPADDR26 0x3ca
```

30.38.1.317 CSR_PMPADDR27

```
#define CSR_PMPADDR27 0x3cb
```

30.38.1.318 CSR_PMPADDR28

```
#define CSR_PMPADDR28 0x3cc
```

30.38.1.319 CSR_PMPADDR29

```
#define CSR_PMPADDR29 0x3cd
```

30.38.1.320 CSR_PMPADDR3

```
#define CSR_PMPADDR3 0x3b3
```

30.38.1.321 CSR_PMPADDR30

```
#define CSR_PMPADDR30 0x3ce
```

30.38.1.322 CSR_PMPADDR31

```
#define CSR_PMPADDR31 0x3cf
```

30.38.1.323 CSR_PMPADDR32

```
#define CSR_PMPADDR32 0x3d0
```

30.38.1.324 CSR_PMPADDR33

```
#define CSR_PMPADDR33 0x3d1
```

30.38.1.325 CSR_PMPADDR34

```
#define CSR_PMPADDR34 0x3d2
```

30.38.1.326 CSR_PMPADDR35

```
#define CSR_PMPADDR35 0x3d3
```

30.38.1.327 CSR_PMPADDR36

```
#define CSR_PMPADDR36 0x3d4
```

30.38.1.328 CSR_PMPADDR37

```
#define CSR_PMPADDR37 0x3d5
```

30.38.1.329 CSR_PMPADDR38

```
#define CSR_PMPADDR38 0x3d6
```

30.38.1.330 CSR_PMPADDR39

```
#define CSR_PMPADDR39 0x3d7
```

30.38.1.331 CSR_PMPADDR4

```
#define CSR_PMPADDR4 0x3b4
```

30.38.1.332 CSR_PMPADDR40

```
#define CSR_PMPADDR40 0x3d8
```

30.38.1.333 CSR_PMPADDR41

```
#define CSR_PMPADDR41 0x3d9
```

30.38.1.334 CSR_PMPADDR42

```
#define CSR_PMPADDR42 0x3da
```

30.38.1.335 CSR_PMPADDR43

```
#define CSR_PMPADDR43 0x3db
```

30.38.1.336 CSR_PMPADDR44

```
#define CSR_PMPADDR44 0x3dc
```

30.38.1.337 CSR_PMPADDR45

```
#define CSR_PMPADDR45 0x3dd
```

30.38.1.338 CSR_PMPADDR46

```
#define CSR_PMPADDR46 0x3de
```

30.38.1.339 CSR_PMPADDR47

```
#define CSR_PMPADDR47 0x3df
```

30.38.1.340 CSR_PMPADDR48

```
#define CSR_PMPADDR48 0x3e0
```

30.38.1.341 CSR_PMPADDR49

```
#define CSR_PMPADDR49 0x3e1
```

30.38.1.342 CSR_PMPADDR5

```
#define CSR_PMPADDR5 0x3b5
```

30.38.1.343 CSR_PMPADDR50

```
#define CSR_PMPADDR50 0x3e2
```

30.38.1.344 CSR_PMPADDR51

```
#define CSR_PMPADDR51 0x3e3
```

30.38.1.345 CSR_PMPADDR52

```
#define CSR_PMPADDR52 0x3e4
```

30.38.1.346 CSR_PMPADDR53

```
#define CSR_PMPADDR53 0x3e5
```

30.38.1.347 CSR_PMPADDR54

```
#define CSR_PMPADDR54 0x3e6
```

30.38.1.348 CSR_PMPADDR55

```
#define CSR_PMPADDR55 0x3e7
```

30.38.1.349 CSR_PMPADDR56

```
#define CSR_PMPADDR56 0x3e8
```

30.38.1.350 CSR_PMPADDR57

```
#define CSR_PMPADDR57 0x3e9
```

30.38.1.351 CSR_PMPADDR58

```
#define CSR_PMPADDR58 0x3ea
```

30.38.1.352 CSR_PMPADDR59

```
#define CSR_PMPADDR59 0x3eb
```

30.38.1.353 CSR_PMPADDR6

```
#define CSR_PMPADDR6 0x3b6
```

30.38.1.354 CSR_PMPADDR60

```
#define CSR_PMPADDR60 0x3ec
```

30.38.1.355 CSR_PMPADDR61

```
#define CSR_PMPADDR61 0x3ed
```

30.38.1.356 CSR_PMPADDR62

```
#define CSR_PMPADDR62 0x3ee
```

30.38.1.357 CSR_PMPADDR63

```
#define CSR_PMPADDR63 0x3ef
```

30.38.1.358 CSR_PMPADDR7

```
#define CSR_PMPADDR7 0x3b7
```

30.38.1.359 CSR_PMPADDR8

```
#define CSR_PMPADDR8 0x3b8
```

30.38.1.360 CSR_PMPADDR9

```
#define CSR_PMPADDR9 0x3b9
```

30.38.1.361 CSR_PMPCFG0

```
#define CSR_PMPCFG0 0x3a0
```

30.38.1.362 CSR_PMPCFG1

```
#define CSR_PMPCFG1 0x3a1
```

30.38.1.363 CSR_PMPCFG10

```
#define CSR_PMPCFG10 0x3aa
```

30.38.1.364 CSR_PMPCFG11

```
#define CSR_PMPCFG11 0x3ab
```

30.38.1.365 CSR_PMPCFG12

```
#define CSR_PMPCFG12 0x3ac
```

30.38.1.366 CSR_PMPCFG13

```
#define CSR_PMPCFG13 0x3ad
```

30.38.1.367 CSR_PMPCFG14

```
#define CSR_PMPCFG14 0x3ae
```

30.38.1.368 CSR_PMPCFG15

```
#define CSR_PMPCFG15 0x3af
```

30.38.1.369 CSR_PMPCFG2

```
#define CSR_PMPCFG2 0x3a2
```

30.38.1.370 CSR_PMPCFG3

```
#define CSR_PMPCFG3 0x3a3
```

30.38.1.371 CSR_PMPCFG4

```
#define CSR_PMPCFG4 0x3a4
```

30.38.1.372 CSR_PMPCFG5

```
#define CSR_PMPCFG5 0x3a5
```

30.38.1.373 CSR_PMPCFG6

```
#define CSR_PMPCFG6 0x3a6
```

30.38.1.374 CSR_PMPCFG7

```
#define CSR_PMPCFG7 0x3a7
```

30.38.1.375 CSR_PMPCFG8

```
#define CSR_PMPCFG8 0x3a8
```

30.38.1.376 CSR_PMPCFG9

```
#define CSR_PMPCFG9 0x3a9
```

30.38.1.377 CSR_SATP

```
#define CSR_SATP 0x180
```

30.38.1.378 CSR_SCAUSE

```
#define CSR_SCAUSE 0x142
```

30.38.1.379 CSR_SCOUNTEREN

```
#define CSR_SCOUNTEREN 0x106
```

30.38.1.380 CSR_SCOUNTINHIBIT

```
#define CSR_SCOUNTINHIBIT 0x120
```

30.38.1.381 CSR_SCOUNTOVF

```
#define CSR_SCOUNTOVF 0xda0
```

30.38.1.382 CSR_SENVCFG

```
#define CSR_SENVCFG 0x10a
```

30.38.1.383 CSR_SEPC

```
#define CSR_SEPC 0x141
```

30.38.1.384 CSR_SIE

```
#define CSR_SIE 0x104
```

30.38.1.385 CSR_SIEH

```
#define CSR_SIEH 0x114
```

30.38.1.386 CSR_SIP

```
#define CSR_SIP 0x144
```

30.38.1.387 CSR_SIPH

```
#define CSR_SIPH 0x154
```

30.38.1.388 CSR_SIREG

```
#define CSR_SIREG 0x151
```

30.38.1.389 CSR_SIREG2

```
#define CSR_SIREG2 0x152
```

30.38.1.390 CSR_SIREG3

```
#define CSR_SIREG3 0x153
```

30.38.1.391 CSR_SIREG4

```
#define CSR_SIREG4 0x155
```

30.38.1.392 CSR_SIREG5

```
#define CSR_SIREG5 0x156
```

30.38.1.393 CSR_SIREG6

```
#define CSR_SIREG6 0x157
```

30.38.1.394 CSR_SISELECT

```
#define CSR_SISELECT 0x150
```

30.38.1.395 CSR_SSCRATCH

```
#define CSR_SSCRATCH 0x140
```

30.38.1.396 CSR_SSP

```
#define CSR_SSP 0x011
```

30.38.1.397 CSR_SSTATEEN0

```
#define CSR_SSTATEEN0 0x10C
```

30.38.1.398 CSR_SSTATEEN1

```
#define CSR_SSTATEEN1 0x10D
```

30.38.1.399 CSR_SSTATEEN2

```
#define CSR_SSTATEEN2 0x10E
```

30.38.1.400 CSR_SSTATEEN3

```
#define CSR_SSTATEEN3 0x10F
```

30.38.1.401 CSR_SSTATUS

```
#define CSR_SSTATUS 0x100
```

30.38.1.402 CSR_STIMECMP

```
#define CSR_STIMECMP 0x14D
```

30.38.1.403 CSR_STIMECMPH

```
#define CSR_STIMECMPH 0x15D
```

30.38.1.404 CSR_STOPEI

```
#define CSR_STOPEI 0x15c
```

30.38.1.405 CSR_STOPI

```
#define CSR_STOPI 0xdb0
```

30.38.1.406 CSR_STVAL

```
#define CSR_STVAL 0x143
```

30.38.1.407 CSR_STVEC

```
#define CSR_STVEC 0x105
```

30.38.1.408 CSR_TDATA1

```
#define CSR_TDATA1 0x7a1
```

30.38.1.409 CSR_TDATA2

```
#define CSR_TDATA2 0x7a2
```

30.38.1.410 CSR_TDATA3

```
#define CSR_TDATA3 0x7a3
```

30.38.1.411 CSR_TIME

```
#define CSR_TIME 0xc01
```

30.38.1.412 CSR_TIMEH

```
#define CSR_TIMEH 0xc81
```

30.38.1.413 CSR_TINFO

```
#define CSR_TINFO 0x7a4
```

30.38.1.414 CSR_TSELECT

```
#define CSR_TSELECT 0x7a0
```

30.38.1.415 CSR_UCAUSE

```
#define CSR_UCAUSE 0x042
```

30.38.1.416 CSR_UEPC

```
#define CSR_UEPC 0x041
```

30.38.1.417 CSR_UIE

```
#define CSR_UIE 0x004
```

30.38.1.418 CSR_UIP

```
#define CSR_UIP 0x044
```

30.38.1.419 CSR_USCRATCH

```
#define CSR_USCRATCH 0x040
```

30.38.1.420 CSR_USTATUS

```
#define CSR_USTATUS 0x000
```

30.38.1.421 CSR_UTVAL

```
#define CSR_UTVAL 0x043
```

30.38.1.422 CSR_UTVEC

```
#define CSR_UTVEC 0x005
```

30.38.1.423 CSR_VL

```
#define CSR_VL 0xc20
```

30.38.1.424 CSR_VLENB

```
#define CSR_VLENB 0xc22
```

30.38.1.425 CSR_VSATP

```
#define CSR_VSATP 0x280
```

30.38.1.426 CSR_VSCAUSE

```
#define CSR_VSCAUSE 0x242
```

30.38.1.427 CSR_VSEPC

```
#define CSR_VSEPC 0x241
```

30.38.1.428 CSR_VSIE

```
#define CSR_VSIE 0x204
```

30.38.1.429 CSR_VSIEH

```
#define CSR_VSIEH 0x214
```

30.38.1.430 CSR_VSIP

```
#define CSR_VSIP 0x244
```

30.38.1.431 CSR_VSIPH

```
#define CSR_VSIPH 0x254
```

30.38.1.432 CSR_VSIREG

```
#define CSR_VSIREG 0x251
```

30.38.1.433 CSR_VSIREG2

```
#define CSR_VSIREG2 0x252
```

30.38.1.434 CSR_VSIREG3

```
#define CSR_VSIREG3 0x253
```

30.38.1.435 CSR_VSIREG4

```
#define CSR_VSIREG4 0x255
```

30.38.1.436 CSR_VSIREG5

```
#define CSR_VSIREG5 0x256
```

30.38.1.437 CSR_VSIREG6

```
#define CSR_VSIREG6 0x257
```

30.38.1.438 CSR_VSISELECT

```
#define CSR_VSISELECT 0x250
```

30.38.1.439 CSR_VSSCRATCH

```
#define CSR_VSSCRATCH 0x240
```

30.38.1.440 CSR_VSSTATUS

```
#define CSR_VSSTATUS 0x200
```

30.38.1.441 CSR_VSTART

```
#define CSR_VSTART 0x8
```

30.38.1.442 CSR_VSTOPEI

```
#define CSR_VSTOPEI 0x25c
```

30.38.1.443 CSR_VSTOPI

```
#define CSR_VSTOPI 0xeb0
```

30.38.1.444 CSR_VSTVAL

```
#define CSR_VSTVAL 0x243
```

30.38.1.445 CSR_VSTVEC

```
#define CSR_VSTVEC 0x205
```

30.38.1.446 CSR_VTYPE

```
#define CSR_VTYPE 0xc21
```

30.38.1.447 CSRRC

```
#define CSRRC 3
```

30.38.1.448 CSRRCI

```
#define CSRRCI 7
```

30.38.1.449 CSRRS

```
#define CSRRS 2
```

30.38.1.450 CSRRSI

```
#define CSRRSI 6
```

30.38.1.451 CSRRW

```
#define CSRRW 1
```

30.38.1.452 CSRRWI

```
#define CSRRWI 5
```

30.38.1.453 ENVCFG_ADUE

```
#define ENVCFG_ADUE (_ULL(1) << ENVCFG_ADUE_SHIFT)
```

30.38.1.454 ENVCFG_ADUE_SHIFT

```
#define ENVCFG_ADUE_SHIFT 61
```

30.38.1.455 ENVCFG_CBCFE

```
#define ENVCFG_CBCFE (_UL(1) << 6)
```

30.38.1.456 ENVCFG_CBIE

```
#define ENVCFG_CBIE (_UL(0x3) << ENVCFG_CBIE_SHIFT)
```

30.38.1.457 ENVCFG_CBIE_FLUSH

```
#define ENVCFG_CBIE_FLUSH _UL(0x1)
```

30.38.1.458 ENVCFG_CBIE_ILL

```
#define ENVCFG_CBIE_ILL _UL(0x0)
```

30.38.1.459 ENVCFG_CBIE_INV

```
#define ENVCFG_CBIE_INV _UL(0x3)
```

30.38.1.460 ENVCFG_CBIE_SHIFT

```
#define ENVCFG_CBIE_SHIFT 4
```

30.38.1.461 ENVCFG_CBZE

```
#define ENVCFG_CBZE (_UL(1) << 7)
```

30.38.1.462 ENVCFG_CDE

```
#define ENVCFG_CDE (_ULL(1) << 60)
```

30.38.1.463 ENVCFG_DTE

```
#define ENVCFG_DTE (_ULL(1) << ENVCFG_DTE_SHIFT)
```

30.38.1.464 ENVCFG_DTE_SHIFT

```
#define ENVCFG_DTE_SHIFT 59
```

30.38.1.465 ENVCFG_FIOM

```
#define ENVCFG_FIOM _UL(0x1)
```

30.38.1.466 ENVCFG_LPE

```
#define ENVCFG_LPE (_UL(1) << ENVCFG_LPE_SHIFT)
```

30.38.1.467 ENVCFG_LPE_SHIFT

```
#define ENVCFG_LPE_SHIFT 2
```

30.38.1.468 ENVCFG_PBMTE

```
#define ENVCFG_PBMTE (_ULL(1) << 62)
```

30.38.1.469 ENVCFG_PMM

```
#define ENVCFG_PMM (_ULL(0x3) << 32)
```

30.38.1.470 ENVCFG_PMM_PMLEN_0

```
#define ENVCFG_PMM_PMLEN_0 (_ULL(0x0) << 32)
```

30.38.1.471 ENVCFG_PMM_PMLEN_16

```
#define ENVCFG_PMM_PMLEN_16 (_ULL(0x3) << 32)
```

30.38.1.472 ENVCFG_PMM_PMLEN_7

```
#define ENVCFG_PMM_PMLEN_7 (_ULL(0x2) << 32)
```

30.38.1.473 ENVCFG_SSE

```
#define ENVCFG_SSE (_UL(1) << ENVCFG_SSE_SHIFT)
```

30.38.1.474 ENVCFG_SSE_SHIFT

```
#define ENVCFG_SSE_SHIFT 3
```

30.38.1.475 ENVCFG_STCE

```
#define ENVCFG_STCE (_ULL(1) << 63)
```

30.38.1.476 GET_CSR_NUM

```
#define GET_CSR_NUM(
              insn ) ((insn & MASK_CSR) >> SHIFT_CSR)
```

30.38.1.477 GET_EMUL

```
#define GET_EMUL(
              vemul ) (1UL << ((vemul) >= 4 ?  0 :  (vemul)))
```

30.38.1.478 GET_LEN

```
#define GET_LEN(
              view ) (1UL << (view))
```

30.38.1.479 GET_MEW

```
#define GET_MEW(
              insn ) (((insn) >> SH_MEW) & 1)
```

30.38.1.480 GET_NF

```
#define GET_NF(
              insn ) (1 + ((insn >> 29) & 7))
```

30.38.1.481 GET_RM

```
#define GET_RM(
            insn ) ((insn & MASK_FUNCT3) >> SHIFT_FUNCT3)
```

30.38.1.482 GET_RS1

```
#define GET_RS1(
            insn,
            regs ) (*REG_PTR(insn, SH_RS1, regs))
```

30.38.1.483 GET_RS1_NUM

```
#define GET_RS1_NUM(
            insn ) ((insn & MASK_RS1) >> 15)
```

30.38.1.484 GET_RS1S

```
#define GET_RS1S(
            insn,
            regs ) (*REG_PTR(RVC_RS1S(insn), 0, regs))
```

30.38.1.485 GET_RS2

```
#define GET_RS2(
            insn,
            regs ) (*REG_PTR(insn, SH_RS2, regs))
```

30.38.1.486 GET_RS2C

```
#define GET_RS2C(
            insn,
            regs ) (*REG_PTR(insn, SH_RS2C, regs))
```

30.38.1.487 GET_RS2S

```
#define GET_RS2S(
            insn,
            regs ) (*REG_PTR(RVC_RS2S(insn), 0, regs))
```

30.38.1.488 GET_SP

```
#define GET_SP(
            regs ) (*REG_PTR(2, 0, regs))
```

30.38.1.489 GET_VD

```
#define GET_VD(
            insn ) ((insn >> SH_VD) & VD_MASK)
```

30.38.1.490 GET_VEMUL

```
#define GET_VEMUL(
            vlmul,
            view,
            vsew ) ((vlmul + view - vsew) & 7)
```

30.38.1.491 GET_VIEW

```
#define GET_VIEW(
            insn ) (((insn) >> SH_VIEW) & VIEW_MASK)
```

30.38.1.492 GET_VLMUL

```
#define GET_VLMUL(
            vtype ) ((vtype) & VLMUL_MASK)
```

30.38.1.493 GET_VS2

```
#define GET_VS2(
            insn ) ((insn >> SH_VS2) & VS2_MASK)
```

30.38.1.494 GET_VSEW

```
#define GET_VSEW(
            vtype ) (((vtype) >> SH_VSEW) & VSEW_MASK)
```

30.38.1.495 HGATP32_MODE_SHIFT

```
#define HGATP32_MODE_SHIFT 31
```

30.38.1.496 HGATP32_PPN

```
#define HGATP32_PPN _UL(0x003FFFFF)
```

30.38.1.497 HGATP32_VMID_MASK

```
#define HGATP32_VMID_MASK _UL(0x1FC00000)
```

30.38.1.498 HGATP32_VMID_SHIFT

```
#define HGATP32_VMID_SHIFT 22
```

30.38.1.499 HGATP64_MODE_SHIFT

```
#define HGATP64_MODE_SHIFT 60
```

30.38.1.500 HGATP64_PPN

```
#define HGATP64_PPN _ULL(0x00000FFFFFFFFFFF)
```

30.38.1.501 HGATP64_VMID_MASK

```
#define HGATP64_VMID_MASK _ULL(0x03FFF00000000000)
```

30.38.1.502 HGATP64_VMID_SHIFT

```
#define HGATP64_VMID_SHIFT 44
```

30.38.1.503 HGATP_MODE_OFF

```
#define HGATP_MODE_OFF _UL(0)
```

30.38.1.504 HGATP_MODE_SHIFT

```
#define HGATP_MODE_SHIFT HGATP32_MODE_SHIFT
```

30.38.1.505 HGATP_MODE_SV32X4

```
#define HGATP_MODE_SV32X4 _UL(1)
```

30.38.1.506 HGATP_MODE_SV39X4

```
#define HGATP_MODE_SV39X4 _UL(8)
```

30.38.1.507 HGATP_MODE_SV48X4

```
#define HGATP_MODE_SV48X4 _UL(9)
```

30.38.1.508 HGATP_PPN

```
#define HGATP_PPN HGATP32_PPN
```

30.38.1.509 HGATP_VMID_MASK

```
#define HGATP_VMID_MASK HGATP32_VMID_MASK
```

30.38.1.510 HGATP_VMID_SHIFT

```
#define HGATP_VMID_SHIFT HGATP32_VMID_SHIFT
```

30.38.1.511 HSTATUS_GVA

```
#define HSTATUS_GVA _UL(0x00000040)
```

30.38.1.512 HSTATUS_HU

```
#define HSTATUS_HU _UL(0x00000200)
```

30.38.1.513 HSTATUS_SPV

```
#define HSTATUS_SPV _UL(0x00000080)
```

30.38.1.514 HSTATUS_SPVP

```
#define HSTATUS_SPVP _UL(0x00000100)
```

30.38.1.515 HSTATUS_VGEIN

```
#define HSTATUS_VGEIN _UL(0x0003f000)
```

30.38.1.516 HSTATUS_VGEIN_SHIFT

```
#define HSTATUS_VGEIN_SHIFT 12
```

30.38.1.517 HSTATUS_VSBE

```
#define HSTATUS_VSBE _UL(0x00000020)
```

30.38.1.518 HSTATUS_VTSR

```
#define HSTATUS_VTSR _UL(0x00400000)
```

30.38.1.519 HSTATUS_VTVM

```
#define HSTATUS_VTVM _UL(0x00100000)
```

30.38.1.520 HSTATUS_VTW

```
#define HSTATUS_VTW _UL(0x00200000)
```

30.38.1.521 IMM_I

```
#define IMM_I(
            insn ) ((s32)(insn) >> 20)
```

30.38.1.522 IMM_S

```
#define IMM_S(
            insn )
```

Value:

```
(((s32)(insn) » 25 « 5) | \
(s32)(((insn) » 7) & 0x1f))
```

30.38.1.523 INSN_16BIT_MASK

```
#define INSN_16BIT_MASK 0x3
```

30.38.1.524 INSN_32BIT_MASK

```
#define INSN_32BIT_MASK 0x1c
```

30.38.1.525 INSN_IS_16BIT

```
#define INSN_IS_16BIT(
            insn )  (((insn) & INSN_16BIT_MASK) != INSN_16BIT_MASK)
```

30.38.1.526 INSN_IS_32BIT

```
#define INSN_IS_32BIT(
            insn )
```

Value:

```
(((insn) & INSN_16BIT_MASK) == INSN_16BIT_MASK && \
 ((insn) & INSN_32BIT_MASK) != INSN_32BIT_MASK)
```

30.38.1.527 INSN_LEN

```
#define INSN_LEN(
            insn ) (INSN_IS_16BIT(insn) ?  2 :  4)
```

30.38.1.528 INSN_MASK_C_FLD

```
#define INSN_MASK_C_FLD 0xe003
```

30.38.1.529 INSN_MASK_C_FLDSP

```
#define INSN_MASK_C_FLDSP 0xe003
```

30.38.1.530 INSN_MASK_C_FLW

```
#define INSN_MASK_C_FLW 0xe003
```

30.38.1.531 INSN_MASK_C_FLWSP

```
#define INSN_MASK_C_FLWSP 0xe003
```

30.38.1.532 INSN_MASK_C_FSD

```
#define INSN_MASK_C_FSD 0xe003
```

30.38.1.533 INSN_MASK_C_FSDSP

```
#define INSN_MASK_C_FSDSP 0xe003
```

30.38.1.534 INSN_MASK_C_FSW

```
#define INSN_MASK_C_FSW 0xe003
```

30.38.1.535 INSN_MASK_C_FSWSP

```
#define INSN_MASK_C_FSWSP 0xe003
```

30.38.1.536 INSN_MASK_C_LD

```
#define INSN_MASK_C_LD 0xe003
```

30.38.1.537 INSN_MASK_C_LDSP

```
#define INSN_MASK_C_LDSP 0xe003
```

30.38.1.538 INSN_MASK_C_LH

```
#define INSN_MASK_C_LH 0xfc43
```

30.38.1.539 INSN_MASK_C_LHU

```
#define INSN_MASK_C_LHU 0xfc43
```

30.38.1.540 INSN_MASK_C_LW

```
#define INSN_MASK_C_LW 0xe003
```

30.38.1.541 INSN_MASK_C_LWSP

```
#define INSN_MASK_C_LWSP 0xe003
```

30.38.1.542 INSN_MASK_C_SD

```
#define INSN_MASK_C_SD 0xe003
```

30.38.1.543 INSN_MASK_C_SDSP

```
#define INSN_MASK_C_SDSP 0xe003
```

30.38.1.544 INSN_MASK_C_SH

```
#define INSN_MASK_C_SH 0xfc43
```

30.38.1.545 INSN_MASK_C_SW

```
#define INSN_MASK_C_SW 0xe003
```

30.38.1.546 INSN_MASK_C_SWSP

```
#define INSN_MASK_C_SWSP 0xe003
```

30.38.1.547 INSN_MASK_FENCE_TSO

```
#define INSN_MASK_FENCE_TSO 0xffffffff
```

30.38.1.548 INSN_MASK_FLD

```
#define INSN_MASK_FLD 0x707f
```

30.38.1.549 INSN_MASK_FLQ

```
#define INSN_MASK_FLQ 0x707f
```

30.38.1.550 INSN_MASK_FLW

```
#define INSN_MASK_FLW 0x707f
```

30.38.1.551 INSN_MASK_FSD

```
#define INSN_MASK_FSD 0x707f
```

30.38.1.552 INSN_MASK_FSQ

```
#define INSN_MASK_FSQ 0x707f
```

30.38.1.553 INSN_MASK_FSW

```
#define INSN_MASK_FSW 0x707f
```

30.38.1.554 INSN_MASK_LB

```
#define INSN_MASK_LB 0x707f
```

30.38.1.555 INSN_MASK_LBU

```
#define INSN_MASK_LBU 0x707f
```

30.38.1.556 INSN_MASK_LD

```
#define INSN_MASK_LD 0x707f
```

30.38.1.557 INSN_MASK_LH

```
#define INSN_MASK_LH 0x707f
```

30.38.1.558 INSN_MASK_LHU

```
#define INSN_MASK_LHU 0x707f
```

30.38.1.559 INSN_MASK_LW

```
#define INSN_MASK_LW 0x707f
```

30.38.1.560 INSN_MASK_LWU

```
#define INSN_MASK_LWU 0x707f
```

30.38.1.561 INSN_MASK_SB

```
#define INSN_MASK_SB 0x707f
```

30.38.1.562 INSN_MASK_SD

```
#define INSN_MASK_SD 0x707f
```

30.38.1.563 INSN_MASK_SH

```
#define INSN_MASK_SH 0x707f
```

30.38.1.564 INSN_MASK_SW

```
#define INSN_MASK_SW 0x707f
```

30.38.1.565 INSN_MASK_VECTOR_FAULT_ONLY_FIRST

```
#define INSN_MASK_VECTOR_FAULT_ONLY_FIRST 0xfdf0707f
```

30.38.1.566 INSN_MASK_VECTOR_INDEXED

```
#define INSN_MASK_VECTOR_INDEXED 0xfc00707f
```

30.38.1.567 INSN_MASK_VECTOR_LOAD_STORE

```
#define INSN_MASK_VECTOR_LOAD_STORE 0x7f
```

30.38.1.568 INSN_MASK_VECTOR_STRIDE

```
#define INSN_MASK_VECTOR_STRIDE 0xfc00707f
```

30.38.1.569 INSN_MASK_VECTOR_UNIT_STRIDE

```
#define INSN_MASK_VECTOR_UNIT_STRIDE 0xfdf0707f
```

30.38.1.570 INSN_MASK_VECTOR_WHOLE_REG

```
#define INSN_MASK_VECTOR_WHOLE_REG 0xfff0707f
```

30.38.1.571 INSN_MASK_WFI

```
#define INSN_MASK_WFI 0xffffff00
```

30.38.1.572 INSN_MATCH_C_FLD

```
#define INSN_MATCH_C_FLD 0x2000
```

30.38.1.573 INSN_MATCH_C_FLDSP

```
#define INSN_MATCH_C_FLDSP 0x2002
```

30.38.1.574 INSN_MATCH_C_FLW

```
#define INSN_MATCH_C_FLW 0x6000
```

30.38.1.575 INSN_MATCH_C_FLWSP

```
#define INSN_MATCH_C_FLWSP 0x6002
```

30.38.1.576 INSN_MATCH_C_FSD

```
#define INSN_MATCH_C_FSD 0xa000
```

30.38.1.577 INSN_MATCH_C_FSDSP

```
#define INSN_MATCH_C_FSDSP 0xa002
```

30.38.1.578 INSN_MATCH_C_FSW

```
#define INSN_MATCH_C_FSW 0xe000
```

30.38.1.579 INSN_MATCH_C_FSWSP

```
#define INSN_MATCH_C_FSWSP 0xe002
```

30.38.1.580 INSN_MATCH_C_LD

```
#define INSN_MATCH_C_LD 0x6000
```

30.38.1.581 INSN_MATCH_C_LDSP

```
#define INSN_MATCH_C_LDSP 0x6002
```

30.38.1.582 INSN_MATCH_C_LH

```
#define INSN_MATCH_C_LH 0x8440
```

30.38.1.583 INSN_MATCH_C_LHU

```
#define INSN_MATCH_C_LHU 0x8400
```

30.38.1.584 INSN_MATCH_C_LW

```
#define INSN_MATCH_C_LW 0x4000
```

30.38.1.585 INSN_MATCH_C_LWSP

```
#define INSN_MATCH_C_LWSP 0x4002
```

30.38.1.586 INSN_MATCH_C_SD

```
#define INSN_MATCH_C_SD 0xe000
```

30.38.1.587 INSN_MATCH_C_SDSP

```
#define INSN_MATCH_C_SDSP 0xe002
```

30.38.1.588 INSN_MATCH_C_SH

```
#define INSN_MATCH_C_SH 0x8c00
```

30.38.1.589 INSN_MATCH_C_SW

```
#define INSN_MATCH_C_SW 0xc000
```

30.38.1.590 INSN_MATCH_C_SWSP

```
#define INSN_MATCH_C_SWSP 0xc002
```

30.38.1.591 INSN_MATCH_FENCE_TSO

```
#define INSN_MATCH_FENCE_TSO 0x8330000f
```

30.38.1.592 INSN_MATCH_FLD

```
#define INSN_MATCH_FLD 0x3007
```

30.38.1.593 INSN_MATCH_FLQ

```
#define INSN_MATCH_FLQ 0x4007
```

30.38.1.594 INSN_MATCH_FLW

```
#define INSN_MATCH_FLW 0x2007
```

30.38.1.595 INSN_MATCH_FSD

```
#define INSN_MATCH_FSD 0x3027
```

30.38.1.596 INSN_MATCH_FSQ

```
#define INSN_MATCH_FSQ 0x4027
```

30.38.1.597 INSN_MATCH_FSW

```
#define INSN_MATCH_FSW 0x2027
```

30.38.1.598 INSN_MATCH_LB

```
#define INSN_MATCH_LB 0x3
```

30.38.1.599 INSN_MATCH_LBU

```
#define INSN_MATCH_LBU 0x4003
```

30.38.1.600 INSN_MATCH_LD

```
#define INSN_MATCH_LD 0x3003
```

30.38.1.601 INSN_MATCH_LH

```
#define INSN_MATCH_LH 0x1003
```

30.38.1.602 INSN_MATCH_LHU

```
#define INSN_MATCH_LHU 0x5003
```

30.38.1.603 INSN_MATCH_LW

```
#define INSN_MATCH_LW 0x2003
```

30.38.1.604 INSN_MATCH_LWU

```
#define INSN_MATCH_LWU 0x6003
```

30.38.1.605 INSN_MATCH_SB

```
#define INSN_MATCH_SB 0x23
```

30.38.1.606 INSN_MATCH_SD

```
#define INSN_MATCH_SD 0x3023
```

30.38.1.607 INSN_MATCH_SH

```
#define INSN_MATCH_SH 0x1023
```

30.38.1.608 INSN_MATCH_SW

```
#define INSN_MATCH_SW 0x2023
```

30.38.1.609 INSN_MATCH_VECTOR_LOAD

```
#define INSN_MATCH_VECTOR_LOAD 0x07
```

30.38.1.610 INSN_MATCH_VECTOR_STORE

```
#define INSN_MATCH_VECTOR_STORE 0x27
```

30.38.1.611 INSN_MATCH_VL1RE16V

```
#define INSN_MATCH_VL1RE16V 0x02805007
```

30.38.1.612 INSN_MATCH_VL1RE32V

```
#define INSN_MATCH_VL1RE32V 0x02806007
```

30.38.1.613 INSN_MATCH_VL1RE64V

```
#define INSN_MATCH_VL1RE64V 0x02807007
```

30.38.1.614 INSN_MATCH_VL1RE8V

```
#define INSN_MATCH_VL1RE8V 0x02800007
```

30.38.1.615 INSN_MATCH_VL2RE16V

```
#define INSN_MATCH_VL2RE16V 0x22805007
```

30.38.1.616 INSN_MATCH_VL2RE32V

```
#define INSN_MATCH_VL2RE32V 0x22806007
```

30.38.1.617 INSN_MATCH_VL2RE64V

```
#define INSN_MATCH_VL2RE64V 0x22807007
```

30.38.1.618 INSN_MATCH_VL2RE8V

```
#define INSN_MATCH_VL2RE8V 0x22800007
```

30.38.1.619 INSN_MATCH_VL4RE16V

```
#define INSN_MATCH_VL4RE16V 0x62805007
```

30.38.1.620 INSN_MATCH_VL4RE32V

```
#define INSN_MATCH_VL4RE32V 0x62806007
```

30.38.1.621 INSN_MATCH_VL4RE64V

```
#define INSN_MATCH_VL4RE64V 0x62807007
```

30.38.1.622 INSN_MATCH_VL4RE8V

```
#define INSN_MATCH_VL4RE8V 0x62800007
```

30.38.1.623 INSN_MATCH_VL8RE16V

```
#define INSN_MATCH_VL8RE16V 0xe2805007
```

30.38.1.624 INSN_MATCH_VL8RE32V

```
#define INSN_MATCH_VL8RE32V 0xe2806007
```

30.38.1.625 INSN_MATCH_VL8RE64V

```
#define INSN_MATCH_VL8RE64V 0xe2807007
```

30.38.1.626 INSN_MATCH_VL8RE8V

```
#define INSN_MATCH_VL8RE8V 0xe2800007
```

30.38.1.627 INSN_MATCH_VLE16FFV

```
#define INSN_MATCH_VLE16FFV 0x01005007
```

30.38.1.628 INSN_MATCH_VLE16V

```
#define INSN_MATCH_VLE16V 0x00005007
```

30.38.1.629 INSN_MATCH_VLE32FFV

```
#define INSN_MATCH_VLE32FFV 0x01006007
```

30.38.1.630 INSN_MATCH_VLE32V

```
#define INSN_MATCH_VLE32V 0x00006007
```

30.38.1.631 INSN_MATCH_VLE64FFV

```
#define INSN_MATCH_VLE64FFV 0x01007007
```

30.38.1.632 INSN_MATCH_VLE64V

```
#define INSN_MATCH_VLE64V 0x00007007
```

30.38.1.633 INSN_MATCH_VLOXEI16V

```
#define INSN_MATCH_VLOXEI16V 0x0c005007
```

30.38.1.634 INSN_MATCH_VLOXEI32V

```
#define INSN_MATCH_VLOXEI32V 0x0c006007
```

30.38.1.635 INSN_MATCH_VLOXEI64V

```
#define INSN_MATCH_VLOXEI64V 0x0c007007
```

30.38.1.636 INSN_MATCH_VLOXSEG

```
#define INSN_MATCH_VLOXSEG(
            n,
            bits )
```

Value:

```
(((n) - 1) « 29) | 0x0c000007 | \
((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.637 INSN_MATCH_VLSE16V

```
#define INSN_MATCH_VLSE16V 0x08005007
```

30.38.1.638 INSN_MATCH_VLSE32V

```
#define INSN_MATCH_VLSE32V 0x08006007
```

30.38.1.639 INSN_MATCH_VLSE64V

```
#define INSN_MATCH_VLSE64V 0x08007007
```

30.38.1.640 INSN_MATCH_VLSEG

```
#define INSN_MATCH_VLSEG(
            n,
            bits )
```

Value:

```
        ((((n) - 1) « 29) | 0x00004007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.641 INSN_MATCH_VLSEGFF

```
#define INSN_MATCH_VLSEGFF(
            n,
            bits )
```

Value:

```
        ((((n) - 1) « 29) | 0x1000007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.642 INSN_MATCH_VLSSEG

```
#define INSN_MATCH_VLSSEG(
            n,
            bits )
```

Value:

```
        ((((n) - 1) « 29) | 0x08000007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.643 INSN_MATCH_VLUXEI16V

```
#define INSN_MATCH_VLUXEI16V 0x04005007
```

30.38.1.644 INSN_MATCH_VLUXEI32V

```
#define INSN_MATCH_VLUXEI32V 0x04006007
```

30.38.1.645 INSN_MATCH_VLUXEI64V

```
#define INSN_MATCH_VLUXEI64V 0x04007007
```

30.38.1.646 INSN_MATCH_VLUXSEG

```
#define INSN_MATCH_VLUXSEG(
            n,
            bits )
```

Value:

```
        ((((n) - 1) « 29) | 0x04000007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.647 INSN_MATCH_VS1RV

```
#define INSN_MATCH_VS1RV 0x02800027
```

30.38.1.648 INSN_MATCH_VS2RV

```
#define INSN_MATCH_VS2RV 0x22800027
```

30.38.1.649 INSN_MATCH_VS4RV

```
#define INSN_MATCH_VS4RV 0x62800027
```

30.38.1.650 INSN_MATCH_VS8RV

```
#define INSN_MATCH_VS8RV 0xe2800027
```

30.38.1.651 INSN_MATCH_VSE16V

```
#define INSN_MATCH_VSE16V 0x00005027
```

30.38.1.652 INSN_MATCH_VSE32V

```
#define INSN_MATCH_VSE32V 0x00006027
```

30.38.1.653 INSN_MATCH_VSE64V

```
#define INSN_MATCH_VSE64V 0x00007027
```

30.38.1.654 INSN_MATCH_VSOXEI16V

```
#define INSN_MATCH_VSOXEI16V 0x0c005027
```

30.38.1.655 INSN_MATCH_VSOXEI32V

```
#define INSN_MATCH_VSOXEI32V 0x0c006027
```

30.38.1.656 INSN_MATCH_VSOXEI64V

```
#define INSN_MATCH_VSOXEI64V 0x0c007027
```

30.38.1.657 INSN_MATCH_VSOXSEG

```
#define INSN_MATCH_VSOXSEG(
            n,
            bits )
```

Value:

```
(((n) - 1) « 29) | 0x0c000027 | \
((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.658 INSN_MATCH_VSSE16V

```
#define INSN_MATCH_VSSE16V 0x08005027
```

30.38.1.659 INSN_MATCH_VSSE32V

```
#define INSN_MATCH_VSSE32V 0x08006027
```

30.38.1.660 INSN_MATCH_VSSE64V

```
#define INSN_MATCH_VSSE64V 0x08007027
```

30.38.1.661 INSN_MATCH_VSSEG

```
#define INSN_MATCH_VSSEG(
            n,
            bits )
```

Value:

```
(((n) - 1) « 29) | 0x00004027 | \
((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.662 INSN_MATCH_VSSSEG

```
#define INSN_MATCH_VSSSEG(
            n,
            bits )
```

Value:

```
(((n) - 1) « 29) | 0x08000027 | \
((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.663 INSN_MATCH_VSUXEI16V

```
#define INSN_MATCH_VSUXEI16V 0x04005027
```

30.38.1.664 INSN_MATCH_VSUXEI32V

```
#define INSN_MATCH_VSUXEI32V 0x04006027
```

30.38.1.665 INSN_MATCH_VSUXEI64V

```
#define INSN_MATCH_VSUXEI64V 0x04007027
```

30.38.1.666 INSN_MATCH_VSUXSEG

```
#define INSN_MATCH_VSUXSEG(
            n,
            bits )
```

Value:

```
(((n) - 1) « 29) | 0x04000027 | \
((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
```

30.38.1.667 INSN_MATCH_WFI

```
#define INSN_MATCH_WFI 0x10500000
```

30.38.1.668 IRQ_M_EXT

```
#define IRQ_M_EXT 11
```

30.38.1.669 IRQ_M_SOFT

```
#define IRQ_M_SOFT 3
```

30.38.1.670 IRQ_M_TIMER

```
#define IRQ_M_TIMER 7
```

30.38.1.671 IRQ_PMU_OVF

```
#define IRQ_PMU_OVF 13
```

30.38.1.672 IRQ_S_EXT

```
#define IRQ_S_EXT 9
```

30.38.1.673 IRQ_S_GEXT

```
#define IRQ_S_GEXT 12
```

30.38.1.674 IRQ_S_SOFT

```
#define IRQ_S_SOFT 1
```

30.38.1.675 IRQ_S_TIMER

```
#define IRQ_S_TIMER 5
```

30.38.1.676 IRQ_VS_EXT

```
#define IRQ_VS_EXT 10
```

30.38.1.677 IRQ_VS_SOFT

```
#define IRQ_VS_SOFT 2
```

30.38.1.678 IRQ_VS_TIMER

```
#define IRQ_VS_TIMER 6
```

30.38.1.679 IS_FAULT_ONLY_FIRST_LOAD

```
#define IS_FAULT_ONLY_FIRST_LOAD(
            insn )
```

Value:

```
( \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLE16FFV) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLE32FFV) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLE64FFV) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(2, 16)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(3, 16)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(4, 16)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(5, 16)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(6, 16)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(7, 16)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(8, 16)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(2, 32)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(3, 32)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(4, 32)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(5, 32)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(6, 32)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(7, 32)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(8, 32)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(2, 64)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(3, 64)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(4, 64)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(5, 64)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(6, 64)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(7, 64)) || \
IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(8, 64)))
```

30.38.1.680 IS_FAULT_ONLY_FIRST_MATCH

```
#define IS_FAULT_ONLY_FIRST_MATCH(
            insn,
            match )  IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_FAULT_ONLY_FIRST)
```

30.38.1.681 IS_INDEXED_LOAD

```
#define IS_INDEXED_LOAD(
            insn )
```

30.38.1.682 IS_INDEXED_MATCH

```
#define IS_INDEXED_MATCH(
            insn,
            match )  IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_INDEXED)
```

30.38.1.683 IS_INDEXED_STORE

```
#define IS_INDEXED_STORE(
            insn )
```

30.38.1.684 IS_MASKED

```
#define IS_MASKED(
            insn ) (((insn >> SH_VM) & VM_MASK) == 0)
```

30.38.1.685 IS_STRIDE_LOAD

```
#define IS_STRIDE_LOAD(
            insn )
```

Value:

```
( \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSE16V) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSE32V) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSE64V) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(2, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(3, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(4, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(5, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(6, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(7, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(8, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(2, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(3, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(4, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(5, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(6, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(7, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(8, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(2, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(3, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(4, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(5, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(6, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(7, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(8, 64)))
```

30.38.1.686 IS_STRIDE_MATCH

```
#define IS_STRIDE_MATCH(
            insn,
            match )  IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_STRIDE)
```

30.38.1.687 IS_STRIDE_STORE

```
#define IS_STRIDE_STORE(
            insn )
```

Value:

```
( \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSE16V) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSE32V) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSE64V) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(2, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(3, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(4, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(5, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(6, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(7, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(8, 16)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(2, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(3, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(4, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(5, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(6, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(7, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(8, 32)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(2, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(3, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(4, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(5, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(6, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(7, 64)) || \
IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(8, 64)))
```

30.38.1.688 IS_UNIT_STRIDE_LOAD

```
#define IS_UNIT_STRIDE_LOAD(
```

```
                insn )
```

Value:

```
    ( \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLE16V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLE32V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLE64V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(2, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(3, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(4, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(5, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(6, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(7, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(8, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(2, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(3, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(4, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(5, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(6, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(7, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(8, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(2, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(3, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(4, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(5, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(6, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(7, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(8, 64)))
```

30.38.1.689 IS_UNIT_STRIDE_MATCH

```
#define IS_UNIT_STRIDE_MATCH(
                insn,
                match )  IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_UNIT_STRIDE)
```

30.38.1.690 IS_UNIT_STRIDE_STORE

```
#define IS_UNIT_STRIDE_STORE(
                insn )
```

Value:

```
    ( \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSE16V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSE32V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSE64V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(2, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(3, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(4, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(5, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(6, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(7, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(8, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(2, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(3, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(4, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(5, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(6, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(7, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(8, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(2, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(3, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(4, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(5, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(6, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(7, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(8, 64)))
```

30.38.1.691 IS_VECTOR_INSN_MATCH

```
#define IS_VECTOR_INSN_MATCH(
                insn,
                match,
                mask )  (((insn) & (mask)) == ((match) & (mask)))
```

30.38.1.692 IS_VECTOR_LOAD_STORE

```
#define IS_VECTOR_LOAD_STORE(
                insn )
```

Value:

```
((((insn) & INSN_MASK_VECTOR_LOAD_STORE) == INSN_MATCH_VECTOR_LOAD) || \
(((insn) & INSN_MASK_VECTOR_LOAD_STORE) == INSN_MATCH_VECTOR_STORE))
```

30.38.1.693 IS_WHOLE_REG_LOAD

```
#define IS_WHOLE_REG_LOAD(
            insn )
```

Value:

```
( \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL1RE8V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL1RE16V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL1RE32V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL1RE64V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL2RE8V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL2RE16V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL2RE32V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL2RE64V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL4RE8V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL4RE16V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL4RE32V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL4RE64V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL8RE8V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL8RE16V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL8RE32V) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL8RE64V))
```

30.38.1.694 IS_WHOLE_REG_MATCH

```
#define IS_WHOLE_REG_MATCH(
            insn,
            match )  IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_WHOLE_REG)
```

30.38.1.695 IS_WHOLE_REG_STORE

```
#define IS_WHOLE_REG_STORE(
            insn )
```

Value:

```
( \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VS1RV) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VS2RV) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VS4RV) || \
IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VS8RV))
```

30.38.1.696 LOG_REGBYTES

```
#define LOG_REGBYTES 2
```

30.38.1.697 MASK_CSR

```
#define MASK_CSR 0xfff00000
```

30.38.1.698 MASK_FUNCT3

```
#define MASK_FUNCT3 0x7000
```

30.38.1.699 MASK_RS1

```
#define MASK_RS1 0xf8000
```

30.38.1.700 MCAUSE_IRQ_MASK

```
#define MCAUSE_IRQ_MASK (_UL(1) << (__riscv_xlen - 1))
```

30.38.1.701 MHPMEVENT_MINH

```
#define MHPMEVENT_MINH (MHPMEVENTH_MINH << 32)
```

30.38.1.702 MHPMEVENT_OF

```
#define MHPMEVENT_OF (MHPMEVENTH_OF << 32)
```

30.38.1.703 MHPMEVENT_SINH

```
#define MHPMEVENT_SINH (MHPMEVENTH_SINH << 32)
```

30.38.1.704 MHPMEVENT_SSCOF_MASK

```
#define MHPMEVENT_SSCOF_MASK _ULL(0xFF00000000000000)
```

30.38.1.705 MHPMEVENT_UINH

```
#define MHPMEVENT_UINH (MHPMEVENTH_UINH << 32)
```

30.38.1.706 MHPMEVENT_VSINH

```
#define MHPMEVENT_VSINH (MHPMEVENTH_VSINH << 32)
```

30.38.1.707 MHPMEVENT_VUINH

```
#define MHPMEVENT_VUINH (MHPMEVENTH_VUINH << 32)
```

30.38.1.708 MHPMEVENTH_MINH

```
#define MHPMEVENTH_MINH (_ULL(1) << 30)
```

30.38.1.709 MHPMEVENTH_OF

```
#define MHPMEVENTH_OF (_ULL(1) << 31)
```

30.38.1.710 MHPMEVENTH_SINH

```
#define MHPMEVENTH_SINH (_ULL(1) << 29)
```

30.38.1.711 MHPMEVENTH_UINH

```
#define MHPMEVENTH_UINH (_ULL(1) << 28)
```

30.38.1.712 MHPMEVENTH_VSINH

```
#define MHPMEVENTH_VSINH (_ULL(1) << 27)
```

30.38.1.713 MHPMEVENTH_VUINH

```
#define MHPMEVENTH_VUINH (_ULL(1) << 26)
```

30.38.1.714 MIP_LCOFIP

```
#define MIP_LCOFIP (_UL(1) << IRQ_PMU_OVF)
```

30.38.1.715 MIP_MEIP

```
#define MIP_MEIP (_UL(1) << IRQ_M_EXT)
```

30.38.1.716 MIP_MSIP

```
#define MIP_MSIP (_UL(1) << IRQ_M_SOFT)
```

30.38.1.717 MIP_MTIP

```
#define MIP_MTIP (_UL(1) << IRQ_M_TIMER)
```

30.38.1.718 MIP_SEIP

```
#define MIP_SEIP (_UL(1) << IRQ_S_EXT)
```

30.38.1.719 MIP_SGEIP

```
#define MIP_SGEIP (_UL(1) << IRQ_S_GEXT)
```

30.38.1.720 MIP_SSIP

```
#define MIP_SSIP (_UL(1) << IRQ_S_SOFT)
```

30.38.1.721 MIP_STIP

```
#define MIP_STIP (_UL(1) << IRQ_S_TIMER)
```

30.38.1.722 MIP_VSEIP

```
#define MIP_VSEIP (_UL(1) << IRQ_VS_EXT)
```

30.38.1.723 MIP_VSSIP

```
#define MIP_VSSIP (_UL(1) << IRQ_VS_SOFT)
```

30.38.1.724 MIP_VSTIP

```
#define MIP_VSTIP (_UL(1) << IRQ_VS_TIMER)
```

30.38.1.725 MSECCFG_MML

```
#define MSECCFG_MML (_UL(1) << MSECCFG_MML_SHIFT)
```

30.38.1.726 MSECCFG_MML_SHIFT

```
#define MSECCFG_MML_SHIFT (0)
```

30.38.1.727 MSECCFG_MMWP

```
#define MSECCFG_MMWP (_UL(1) << MSECCFG_MMWP_SHIFT)
```

30.38.1.728 MSECCFG_MMWP_SHIFT

```
#define MSECCFG_MMWP_SHIFT (1)
```

30.38.1.729 MSECCFG_RLB

```
#define MSECCFG_RLB (_UL(1) << MSECCFG_RLB_SHIFT)
```

30.38.1.730 MSECCFG_RLB_SHIFT

```
#define MSECCFG_RLB_SHIFT (2)
```

30.38.1.731 MSECCFG_SSEED

```
#define MSECCFG_SSEED (_UL(1) << MSECCFG_SSEED_SHIFT)
```

30.38.1.732 MSECCFG_SSEED_SHIFT

```
#define MSECCFG_SSEED_SHIFT (9)
```

30.38.1.733 MSECCFG_USEED

```
#define MSECCFG_USEED (_UL(1) << MSECCFG_USEED_SHIFT)
```

30.38.1.734 MSECCFG_USEED_SHIFT

```
#define MSECCFG_USEED_SHIFT (8)
```

30.38.1.735 MSTATUS32_SD [1/2]

```
#define MSTATUS32_SD _UL(0x80000000)
```

30.38.1.736 MSTATUS32_SD [2/2]

```
#define MSTATUS32_SD _UL(0x80000000)
```

30.38.1.737 MSTATUS64_SD

```
#define MSTATUS64_SD _ULL(0x8000000000000000)
```

30.38.1.738 MSTATUS_FS

```
#define MSTATUS_FS _UL(0x00006000)
```

30.38.1.739 MSTATUS_MIE

```
#define MSTATUS_MIE _UL(0x00000008)
```

30.38.1.740 MSTATUS_MPIE

```
#define MSTATUS_MPIE _UL(0x00000080)
```

30.38.1.741 MSTATUS_MPP

```
#define MSTATUS_MPP (_UL(3) << MSTATUS_MPP_SHIFT)
```

30.38.1.742 MSTATUS_MPP_SHIFT

```
#define MSTATUS_MPP_SHIFT 11
```

30.38.1.743 MSTATUS_MPRV

```
#define MSTATUS_MPRV _UL(0x00020000)
```

30.38.1.744 MSTATUS_MXR

```
#define MSTATUS_MXR _UL(0x00080000)
```

30.38.1.745 MSTATUS_SD

```
#define MSTATUS_SD MSTATUS32_SD
```

30.38.1.746 MSTATUS_SIE

```
#define MSTATUS_SIE _UL(0x00000002)
```

30.38.1.747 MSTATUS_SPELP

```
#define MSTATUS_SPELP _UL(0x00800000)
```

30.38.1.748 MSTATUS_SPIE

```
#define MSTATUS_SPIE (_UL(1) << MSTATUS_SPIE_SHIFT)
```

30.38.1.749 MSTATUS_SPIE_SHIFT

```
#define MSTATUS_SPIE_SHIFT 5
```

30.38.1.750 MSTATUS_SPP

```
#define MSTATUS_SPP (_UL(1) << MSTATUS_SPP_SHIFT)
```

30.38.1.751 MSTATUS_SPP_SHIFT

```
#define MSTATUS_SPP_SHIFT 8
```

30.38.1.752 MSTATUS_SUM

```
#define MSTATUS_SUM _UL(0x00040000)
```

30.38.1.753 MSTATUS_TSR

```
#define MSTATUS_TSR _UL(0x00400000)
```

30.38.1.754 MSTATUS_TVM

```
#define MSTATUS_TVM _UL(0x00100000)
```

30.38.1.755 MSTATUS_TW

```
#define MSTATUS_TW _UL(0x00200000)
```

30.38.1.756 MSTATUS_UBE

```
#define MSTATUS_UBE _UL(0x00000040)
```

30.38.1.757 MSTATUS_VS

```
#define MSTATUS_VS _UL(0x00000600)
```

30.38.1.758 MSTATUS_XS

```
#define MSTATUS_XS _UL(0x00018000)
```

30.38.1.759 MSTATUSH_GVA

```
#define MSTATUSH_GVA _UL(0x00000040)
```

30.38.1.760 MSTATUSH_GVA_SHIFT

```
#define MSTATUSH_GVA_SHIFT 6
```

30.38.1.761 MSTATUSH_MBE

```
#define MSTATUSH_MBE _UL(0x00000020)
```

30.38.1.762 MSTATUSH_MDT

```
#define MSTATUSH_MDT _UL(0x00000400)
```

30.38.1.763 MSTATUSH_MPELP

```
#define MSTATUSH_MPELP _UL(0x00000200)
```

30.38.1.764 MSTATUSH_MPV

```
#define MSTATUSH_MPV _UL(0x00000080)
```

30.38.1.765 MSTATUSH_SBE

```
#define MSTATUSH_SBE _UL(0x00000010)
```

30.38.1.766 PMP_A

```
#define PMP_A _UL(0x18)
```

30.38.1.767 PMP_A_NA4

```
#define PMP_A_NA4 _UL(0x10)
```

30.38.1.768 PMP_A_NAPOT

```
#define PMP_A_NAPOT _UL(0x18)
```

30.38.1.769 PMP_A_TOR

```
#define PMP_A_TOR _UL(0x08)
```

30.38.1.770 PMP_ADDR_MASK

```
#define PMP_ADDR_MASK _UL(0xFFFFFFFF)
```

30.38.1.771 PMP_COUNT

```
#define PMP_COUNT 64
```

30.38.1.772 PMP_L

```
#define PMP_L _UL(0x80)
```

30.38.1.773 PMP_R

```
#define PMP_R _UL(0x01)
```

30.38.1.774 PMP_SHIFT

```
#define PMP_SHIFT 2
```

30.38.1.775 PMP_W

```
#define PMP_W _UL(0x02)
```

30.38.1.776 PMP_X

```
#define PMP_X _UL(0x04)
```

30.38.1.777 PRV_M

```
#define PRV_M _UL(3)
```

30.38.1.778 PRV_S

```
#define PRV_S _UL(1)
```

30.38.1.779 PRV_U

```
#define PRV_U _UL(0)
```

30.38.1.780 REG_MASK

```
#define REG_MASK  ((1 << (5 + LOG_REGBYTES)) - (1 << LOG_REGBYTES))
```

30.38.1.781 REG_OFFSET

```
#define REG_OFFSET(
            insn,
            pos )  (SHIFT_RIGHT((insn), (pos) - LOG_REGBYTES) & REG_MASK)
```

30.38.1.782 REG_PTR

```
#define REG_PTR(
            insn,
            pos,
            regs )  (ulong *)((ulong)(regs) + REG_OFFSET(insn, pos))
```

30.38.1.783 REGBYTES

```
#define REGBYTES (1 << LOG_REGBYTES)
```

30.38.1.784 RV_X

```
#define RV_X(
            x,
            s,
            n ) (((x) >> (s)) & ((1 << (n)) - 1))
```

30.38.1.785 RVC_LD_IMM

```
#define RVC_LD_IMM(
            x )
```

Value:

```
((RV_X(x, 10, 3) « 3) | \
(RV_X(x, 5, 2) « 6))
```

30.38.1.786 RVC_LDSP_IMM

```
#define RVC_LDSP_IMM(
            x )
```

Value:

```
((RV_X(x, 5, 2) « 3) | \
(RV_X(x, 12, 1) « 5) | \
(RV_X(x, 2, 3) « 6))
```

30.38.1.787 RVC_LW_IMM

```
#define RVC_LW_IMM(
            x )
```

Value:

```
((RV_X(x, 6, 1) « 2) | \
(RV_X(x, 10, 3) « 3) | \
(RV_X(x, 5, 1) « 6))
```

30.38.1.788 RVC_LWSP_IMM

```
#define RVC_LWSP_IMM(
            x )
```

Value:

```
((RV_X(x, 4, 3) « 2) | \
(RV_X(x, 12, 1) « 5) | \
(RV_X(x, 2, 2) « 6))
```

30.38.1.789 RVC_RS1S

```
#define RVC_RS1S(
            insn ) (8 + RV_X(insn, SH_RD, 3))
```

30.38.1.790 RVC_RS2

```
#define RVC_RS2(
            insn ) RV_X(insn, SH_RS2C, 5)
```

30.38.1.791 RVC_RS2S

```
#define RVC_RS2S(
            insn ) (8 + RV_X(insn, SH_RS2C, 3))
```

30.38.1.792 RVC_SDSP_IMM

```
#define RVC_SDSP_IMM(
            x )
```

Value:

```
((RV_X(x, 10, 3) « 3) | \
(RV_X(x, 7, 3) « 6))
```

30.38.1.793 RVC_SWSP_IMM

```
#define RVC_SWSP_IMM(
            x )
```

Value:

```
((RV_X(x, 9, 4) « 2) | \
(RV_X(x, 7, 2) « 6))
```

30.38.1.794 SATP32_ASID

```
#define SATP32_ASID _UL(0x7FC00000)
```

30.38.1.795 SATP32_MODE

```
#define SATP32_MODE _UL(0x80000000)
```

30.38.1.796 SATP32_PPN

```
#define SATP32_PPN _UL(0x003FFFFF)
```

30.38.1.797 SATP64_ASID

```
#define SATP64_ASID _ULL(0x0FFFF00000000000)
```

30.38.1.798 SATP64_MODE

```
#define SATP64_MODE _ULL(0xF000000000000000)
```

30.38.1.799 SATP64_PPN

```
#define SATP64_PPN _ULL(0x00000FFFFFFFFFFF)
```

30.38.1.800 SATP_MODE

```
#define SATP_MODE SATP32_MODE
```

30.38.1.801 SATP_MODE_OFF

```
#define SATP_MODE_OFF _UL(0)
```

30.38.1.802 SATP_MODE_SV32

```
#define SATP_MODE_SV32 _UL(1)
```

30.38.1.803 SATP_MODE_SV39

```
#define SATP_MODE_SV39 _UL(8)
```

30.38.1.804 SATP_MODE_SV48

```
#define SATP_MODE_SV48 _UL(9)
```

30.38.1.805 SATP_MODE_SV57

```
#define SATP_MODE_SV57 _UL(10)
```

30.38.1.806 SATP_MODE_SV64

```
#define SATP_MODE_SV64 _UL(11)
```

30.38.1.807 SET_RD

```
#define SET_RD(
              insn,
              regs,
              val ) (*REG_PTR(insn, SH_RD, regs) = (val))
```

30.38.1.808 SH_MEW

```
#define SH_MEW 28
```

30.38.1.809 SH_RD

```
#define SH_RD 7
```

30.38.1.810 SH_RS1

```
#define SH_RS1 15
```

30.38.1.811 SH_RS2

```
#define SH_RS2 20
```

30.38.1.812 SH_RS2C

```
#define SH_RS2C 2
```

30.38.1.813 SH_VD

```
#define SH_VD 7
```

30.38.1.814 SH_VIEW

```
#define SH_VIEW 12
```

30.38.1.815 SH_VM

```
#define SH_VM 25
```

30.38.1.816 SH_VS2

```
#define SH_VS2 20
```

30.38.1.817 SH_VSEW

```
#define SH_VSEW 3
```

30.38.1.818 SHIFT_CSR

```
#define SHIFT_CSR 20
```

30.38.1.819 SHIFT_FUNCT3

```
#define SHIFT_FUNCT3 12
```

30.38.1.820 SHIFT_RIGHT

```
#define SHIFT_RIGHT(
            x,
            y )  ((y) < 0 ?  ((x) << -(y)) :  ((x) >> (y)))
```

30.38.1.821 SIP_SSIP

```
#define SIP_SSIP MIP_SSIP
```

30.38.1.822 SIP_STIP

```
#define SIP_STIP MIP_STIP
```

30.38.1.823 SMSTATEEN0_AIA

```
#define SMSTATEEN0_AIA (_ULL(1) << SMSTATEEN0_AIA_SHIFT)
```

30.38.1.824 SMSTATEEN0_AIA_SHIFT

```
#define SMSTATEEN0_AIA_SHIFT 59
```

30.38.1.825 SMSTATEEN0_CONTEXT

```
#define SMSTATEEN0_CONTEXT (_ULL(1) << SMSTATEEN0_CONTEXT_SHIFT)
```

30.38.1.826 SMSTATEEN0_CONTEXT_SHIFT

```
#define SMSTATEEN0_CONTEXT_SHIFT 57
```

30.38.1.827 SMSTATEEN0_CS

```
#define SMSTATEEN0_CS (_ULL(1) << SMSTATEEN0_CS_SHIFT)
```

30.38.1.828 SMSTATEEN0_CS_SHIFT

```
#define SMSTATEEN0_CS_SHIFT 0
```

30.38.1.829 SMSTATEEN0_FCSR

```
#define SMSTATEEN0_FCSR (_ULL(1) << SMSTATEEN0_FCSR_SHIFT)
```

30.38.1.830 SMSTATEEN0_FCSR_SHIFT

```
#define SMSTATEEN0_FCSR_SHIFT 1
```

30.38.1.831 SMSTATEEN0_HSENVCFG

```
#define SMSTATEEN0_HSENVCFG (_ULL(1) << SMSTATEEN0_HSENVCFG_SHIFT)
```

30.38.1.832 SMSTATEEN0_HSENVCFG_SHIFT

```
#define SMSTATEEN0_HSENVCFG_SHIFT 62
```

30.38.1.833 SMSTATEEN0_IMSIC

```
#define SMSTATEEN0_IMSIC (_ULL(1) << SMSTATEEN0_IMSIC_SHIFT)
```

30.38.1.834 SMSTATEEN0_IMSIC_SHIFT

```
#define SMSTATEEN0_IMSIC_SHIFT 58
```

30.38.1.835 SMSTATEEN0_SVSLCT

```
#define SMSTATEEN0_SVSLCT (_ULL(1) << SMSTATEEN0_SVSLCT_SHIFT)
```

30.38.1.836 SMSTATEEN0_SVSLCT_SHIFT

```
#define SMSTATEEN0_SVSLCT_SHIFT 60
```

30.38.1.837 SMSTATEEN_MAX_COUNT

```
#define SMSTATEEN_MAX_COUNT 4
```

30.38.1.838 SMSTATEEN_STATEN

```
#define SMSTATEEN_STATEN (_ULL(1) << SMSTATEEN_STATEN_SHIFT)
```

30.38.1.839 SMSTATEEN_STATEN_SHIFT

```
#define SMSTATEEN_STATEN_SHIFT 63
```

30.38.1.840 SSTATUS32_SD

```
#define SSTATUS32_SD MSTATUS32_SD
```

30.38.1.841 SSTATUS64_SD

```
#define SSTATUS64_SD MSTATUS64_SD
```

30.38.1.842 SSTATUS64_UXL

```
#define SSTATUS64_UXL MSTATUS_UXL
```

30.38.1.843 SSTATUS_FS

```
#define SSTATUS_FS MSTATUS_FS
```

30.38.1.844 SSTATUS_MXR

```
#define SSTATUS_MXR MSTATUS_MXR
```

30.38.1.845 SSTATUS_SD

```
#define SSTATUS_SD SSTATUS32_SD
```

30.38.1.846 SSTATUS_SIE

```
#define SSTATUS_SIE MSTATUS_SIE
```

30.38.1.847 SSTATUS_SPIE

```
#define SSTATUS_SPIE MSTATUS_SPIE
```

30.38.1.848 SSTATUS_SPIE_SHIFT

```
#define SSTATUS_SPIE_SHIFT MSTATUS_SPIE_SHIFT
```

30.38.1.849 SSTATUS_SPP

```
#define SSTATUS_SPP MSTATUS_SPP
```

30.38.1.850 SSTATUS_SPP_SHIFT

```
#define SSTATUS_SPP_SHIFT MSTATUS_SPP_SHIFT
```

30.38.1.851 SSTATUS_SUM

```
#define SSTATUS_SUM MSTATUS_SUM
```

30.38.1.852 SSTATUS_VS

```
#define SSTATUS_VS MSTATUS_VS
```

30.38.1.853 SSTATUS_XS

```
#define SSTATUS_XS MSTATUS_XS
```

30.38.1.854 TOPI_IID_MASK

```
#define TOPI_IID_MASK 0xfff
```

30.38.1.855 TOPI_IID_SHIFT

```
#define TOPI_IID_SHIFT 16
```

30.38.1.856 TOPI_IPRIO_MASK

```
#define TOPI_IPRIO_MASK 0xff
```

30.38.1.857 VD_MASK

```
#define VD_MASK 0x1f
```

30.38.1.858 VIEW_MASK

```
#define VIEW_MASK 0x3
```

30.38.1.859 VLMUL_MASK

```
#define VLMUL_MASK 0x7
```

30.38.1.860 VM_MASK

```
#define VM_MASK 0x1
```

30.38.1.861 VS2_MASK

```
#define VS2_MASK 0x1f
```

30.38.1.862 VSEW_MASK

```
#define VSEW_MASK 0x3
```

30.39 riscv_encoding.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __RISCV_ENCODING_H__
#define __RISCV_ENCODING_H__

#include <sbi/sbi_const.h>

/* clang-format off */
#define MSTATUS_SIE         _UL(0x00000002)
#define MSTATUS_MIE         _UL(0x00000008)
#define MSTATUS_SPIE_SHIFT      5
#define MSTATUS_SPIE            (_UL(1) « MSTATUS_SPIE_SHIFT)
#define MSTATUS_UBE         _UL(0x00000040)
#define MSTATUS_MPIE            _UL(0x00000080)
#define MSTATUS_SPP_SHIFT       8
#define MSTATUS_SPP         (_UL(1) « MSTATUS_SPP_SHIFT)
#define MSTATUS_MPP_SHIFT       11
#define MSTATUS_MPP         (_UL(3) « MSTATUS_MPP_SHIFT)
#define MSTATUS_FS          _UL(0x00006000)
#define MSTATUS_XS          _UL(0x00018000)
#define MSTATUS_VS          _UL(0x00000600)
#define MSTATUS_MPRV            _UL(0x00020000)
#define MSTATUS_SUM         _UL(0x00040000)
#define MSTATUS_MXR         _UL(0x00080000)
#define MSTATUS_TVM         _UL(0x00100000)
#define MSTATUS_TW          _UL(0x00200000)
#define MSTATUS_TSR         _UL(0x00400000)
#define MSTATUS_SPELP       _UL(0x00800000)
#define MSTATUS32_SD            _UL(0x80000000)
#if __riscv_xlen == 64
#define MSTATUS_UXL         _ULL(0x0000000300000000)
#define MSTATUS_SXL         _ULL(0x0000000C00000000)
#define MSTATUS_SBE         _ULL(0x0000001000000000)
#define MSTATUS_MBE         _ULL(0x0000002000000000)
#define MSTATUS_GVA         _ULL(0x0000004000000000)
#define MSTATUS_GVA_SHIFT       38
#define MSTATUS_MPV         _ULL(0x0000008000000000)
#define MSTATUS_MPELP           _ULL(0x0000020000000000)
#define MSTATUS_MDT         _ULL(0x0000040000000000)
#else
#define MSTATUSH_SBE            _UL(0x00000010)
```

```
#define MSTATUSH_MBE            _UL(0x00000020)
#define MSTATUSH_GVA            _UL(0x00000040)
#define MSTATUSH_GVA_SHIFT      6
#define MSTATUSH_MPV            _UL(0x00000080)
#define MSTATUSH_MPELP          _UL(0x00000200)
#define MSTATUSH_MDT            _UL(0x00000400)
#endif
#define MSTATUS32_SD            _UL(0x80000000)
#define MSTATUS64_SD            _ULL(0x8000000000000000)

#define SSTATUS_SIE         MSTATUS_SIE
#define SSTATUS_SPIE_SHIFT      MSTATUS_SPIE_SHIFT
#define SSTATUS_SPIE            MSTATUS_SPIE
#define SSTATUS_SPP_SHIFT       MSTATUS_SPP_SHIFT
#define SSTATUS_SPP         MSTATUS_SPP
#define SSTATUS_FS          MSTATUS_FS
#define SSTATUS_XS          MSTATUS_XS
#define SSTATUS_VS          MSTATUS_VS
#define SSTATUS_SUM         MSTATUS_SUM
#define SSTATUS_MXR         MSTATUS_MXR
#define SSTATUS32_SD            MSTATUS32_SD
#define SSTATUS64_UXL           MSTATUS_UXL
#define SSTATUS64_SD            MSTATUS64_SD

#if __riscv_xlen == 64
#define HSTATUS_VSXL            _UL(0x300000000)
#define HSTATUS_VSXL_SHIFT      32
#endif
#define HSTATUS_VTSR            _UL(0x00400000)
#define HSTATUS_VTW         _UL(0x00200000)
#define HSTATUS_VTVM            _UL(0x00100000)
#define HSTATUS_VGEIN           _UL(0x0003f000)
#define HSTATUS_VGEIN_SHIFT     12
#define HSTATUS_HU          _UL(0x00000200)
#define HSTATUS_SPVP            _UL(0x00000100)
#define HSTATUS_SPV         _UL(0x00000080)
#define HSTATUS_GVA         _UL(0x00000040)
#define HSTATUS_VSBE            _UL(0x00000020)

#define MCAUSE_IRQ_MASK         (_UL(1) << (__riscv_xlen - 1))

#define IRQ_S_SOFT          1
#define IRQ_VS_SOFT         2
#define IRQ_M_SOFT          3
#define IRQ_S_TIMER         5
#define IRQ_VS_TIMER            6
#define IRQ_M_TIMER         7
#define IRQ_S_EXT           9
#define IRQ_VS_EXT          10
#define IRQ_M_EXT           11
#define IRQ_S_GEXT          12
#define IRQ_PMU_OVF         13

#define MIP_SSIP            (_UL(1) << IRQ_S_SOFT)
#define MIP_VSSIP           (_UL(1) << IRQ_VS_SOFT)
#define MIP_MSIP            (_UL(1) << IRQ_M_SOFT)
#define MIP_STIP            (_UL(1) << IRQ_S_TIMER)
#define MIP_VSTIP           (_UL(1) << IRQ_VS_TIMER)
#define MIP_MTIP            (_UL(1) << IRQ_M_TIMER)
#define MIP_SEIP            (_UL(1) << IRQ_S_EXT)
#define MIP_VSEIP           (_UL(1) << IRQ_VS_EXT)
#define MIP_MEIP            (_UL(1) << IRQ_M_EXT)
#define MIP_SGEIP           (_UL(1) << IRQ_S_GEXT)
#define MIP_LCOFIP          (_UL(1) << IRQ_PMU_OVF)

#define SIP_SSIP            MIP_SSIP
#define SIP_STIP            MIP_STIP

#define PRV_U               _UL(0)
#define PRV_S               _UL(1)
#define PRV_M               _UL(3)

#define SATP32_MODE         _UL(0x80000000)
#define SATP32_ASID         _UL(0x7FC00000)
#define SATP32_PPN          _UL(0x003FFFFF)
#define SATP64_MODE         _ULL(0xF000000000000000)
#define SATP64_ASID         _ULL(0x0FFFF00000000000)
#define SATP64_PPN          _ULL(0x00000FFFFFFFFFFF)

#define SATP_MODE_OFF           _UL(0)
#define SATP_MODE_SV32          _UL(1)
#define SATP_MODE_SV39          _UL(8)
#define SATP_MODE_SV48          _UL(9)
#define SATP_MODE_SV57          _UL(10)
#define SATP_MODE_SV64          _UL(11)

#define HGATP_MODE_OFF          _UL(0)
```

```
#define HGATP_MODE_SV32X4       _UL(1)
#define HGATP_MODE_SV39X4       _UL(8)
#define HGATP_MODE_SV48X4       _UL(9)

#define HGATP32_MODE_SHIFT      31
#define HGATP32_VMID_SHIFT      22
#define HGATP32_VMID_MASK       _UL(0x1FC00000)
#define HGATP32_PPN         _UL(0x003FFFFF)

#define HGATP64_MODE_SHIFT      60
#define HGATP64_VMID_SHIFT      44
#define HGATP64_VMID_MASK       _ULL(0x03FFF00000000000)
#define HGATP64_PPN         _ULL(0x00000FFFFFFFFFFF)

#define PMP_R               _UL(0x01)
#define PMP_W               _UL(0x02)
#define PMP_X               _UL(0x04)
#define PMP_A               _UL(0x18)
#define PMP_A_TOR           _UL(0x08)
#define PMP_A_NA4           _UL(0x10)
#define PMP_A_NAPOT         _UL(0x18)
#define PMP_L               _UL(0x80)

#define PMP_SHIFT           2
#define PMP_COUNT           64
#if __riscv_xlen == 64
#define PMP_ADDR_MASK           ((_ULL(0x1) << 54) - 1)
#else
#define PMP_ADDR_MASK           _UL(0xFFFFFFFF)
#endif

#if __riscv_xlen == 64
#define MSTATUS_SD          MSTATUS64_SD
#define SSTATUS_SD          SSTATUS64_SD
#define SATP_MODE           SATP64_MODE

#define HGATP_PPN           HGATP64_PPN
#define HGATP_VMID_SHIFT        HGATP64_VMID_SHIFT
#define HGATP_VMID_MASK         HGATP64_VMID_MASK
#define HGATP_MODE_SHIFT        HGATP64_MODE_SHIFT
#else
#define MSTATUS_SD          MSTATUS32_SD
#define SSTATUS_SD          SSTATUS32_SD
#define SATP_MODE           SATP32_MODE

#define HGATP_PPN           HGATP32_PPN
#define HGATP_VMID_SHIFT        HGATP32_VMID_SHIFT
#define HGATP_VMID_MASK         HGATP32_VMID_MASK
#define HGATP_MODE_SHIFT        HGATP32_MODE_SHIFT
#endif

#define TOPI_IID_SHIFT          16
#define TOPI_IID_MASK           0xfff
#define TOPI_IPRIO_MASK     0xff

#if __riscv_xlen == 64
#define MHPMEVENT_OF            (_UL(1) << 63)
#define MHPMEVENT_MINH          (_UL(1) << 62)
#define MHPMEVENT_SINH          (_UL(1) << 61)
#define MHPMEVENT_UINH          (_UL(1) << 60)
#define MHPMEVENT_VSINH         (_UL(1) << 59)
#define MHPMEVENT_VUINH         (_UL(1) << 58)
#else
#define MHPMEVENTH_OF           (_ULL(1) << 31)
#define MHPMEVENTH_MINH         (_ULL(1) << 30)
#define MHPMEVENTH_SINH         (_ULL(1) << 29)
#define MHPMEVENTH_UINH         (_ULL(1) << 28)
#define MHPMEVENTH_VSINH        (_ULL(1) << 27)
#define MHPMEVENTH_VUINH        (_ULL(1) << 26)

#define MHPMEVENT_OF            (MHPMEVENTH_OF << 32)
#define MHPMEVENT_MINH          (MHPMEVENTH_MINH << 32)
#define MHPMEVENT_SINH          (MHPMEVENTH_SINH << 32)
#define MHPMEVENT_UINH          (MHPMEVENTH_UINH << 32)
#define MHPMEVENT_VSINH         (MHPMEVENTH_VSINH << 32)
#define MHPMEVENT_VUINH         (MHPMEVENTH_VUINH << 32)

#endif

#define MHPMEVENT_SSCOF_MASK        _ULL(0xFF00000000000000)

#define ENVCFG_STCE         (_ULL(1) << 63)
#define ENVCFG_PBMTE            (_ULL(1) << 62)
#define ENVCFG_ADUE_SHIFT       61
#define ENVCFG_ADUE         (_ULL(1) << ENVCFG_ADUE_SHIFT)
#define ENVCFG_CDE          (_ULL(1) << 60)
#define ENVCFG_DTE_SHIFT        59
```

```
#define ENVCFG_DTE          (_ULL(1) << ENVCFG_DTE_SHIFT)
#define ENVCFG_PMM          (_ULL(0x3) << 32)
#define ENVCFG_PMM_PMLEN_0      (_ULL(0x0) << 32)
#define ENVCFG_PMM_PMLEN_7      (_ULL(0x2) << 32)
#define ENVCFG_PMM_PMLEN_16     (_ULL(0x3) << 32)
#define ENVCFG_CBZE         (_UL(1) << 7)
#define ENVCFG_CBCFE            (_UL(1) << 6)
#define ENVCFG_CBIE_SHIFT       4
#define ENVCFG_CBIE         (_UL(0x3) << ENVCFG_CBIE_SHIFT)
#define ENVCFG_CBIE_ILL         _UL(0x0)
#define ENVCFG_CBIE_FLUSH       _UL(0x1)
#define ENVCFG_CBIE_INV         _UL(0x3)
#define ENVCFG_SSE_SHIFT        3
#define ENVCFG_SSE          (_UL(1) << ENVCFG_SSE_SHIFT)
#define ENVCFG_LPE_SHIFT        2
#define ENVCFG_LPE          (_UL(1) << ENVCFG_LPE_SHIFT)
#define ENVCFG_FIOM         _UL(0x1)

/* ===== User-level CSRs ===== */

/* User Trap Setup (N-extension) */
#define CSR_USTATUS         0x000
#define CSR_UIE             0x004
#define CSR_UTVEC           0x005
#define CSR_SSP             0x011

/* User Trap Handling (N-extension) */
#define CSR_USCRATCH            0x040
#define CSR_UEPC            0x041
#define CSR_UCAUSE          0x042
#define CSR_UTVAL           0x043
#define CSR_UIP             0x044

/* User Floating-point CSRs */
#define CSR_FFLAGS          0x001
#define CSR_FRM             0x002
#define CSR_FCSR            0x003

/* User Counters/Timers */
#define CSR_CYCLE           0xc00
#define CSR_TIME            0xc01
#define CSR_INSTRET         0xc02
#define CSR_HPMCOUNTER3         0xc03
#define CSR_HPMCOUNTER4         0xc04
#define CSR_HPMCOUNTER5         0xc05
#define CSR_HPMCOUNTER6         0xc06
#define CSR_HPMCOUNTER7         0xc07
#define CSR_HPMCOUNTER8         0xc08
#define CSR_HPMCOUNTER9         0xc09
#define CSR_HPMCOUNTER10        0xc0a
#define CSR_HPMCOUNTER11        0xc0b
#define CSR_HPMCOUNTER12        0xc0c
#define CSR_HPMCOUNTER13        0xc0d
#define CSR_HPMCOUNTER14        0xc0e
#define CSR_HPMCOUNTER15        0xc0f
#define CSR_HPMCOUNTER16        0xc10
#define CSR_HPMCOUNTER17        0xc11
#define CSR_HPMCOUNTER18        0xc12
#define CSR_HPMCOUNTER19        0xc13
#define CSR_HPMCOUNTER20        0xc14
#define CSR_HPMCOUNTER21        0xc15
#define CSR_HPMCOUNTER22        0xc16
#define CSR_HPMCOUNTER23        0xc17
#define CSR_HPMCOUNTER24        0xc18
#define CSR_HPMCOUNTER25        0xc19
#define CSR_HPMCOUNTER26        0xc1a
#define CSR_HPMCOUNTER27        0xc1b
#define CSR_HPMCOUNTER28        0xc1c
#define CSR_HPMCOUNTER29        0xc1d
#define CSR_HPMCOUNTER30        0xc1e
#define CSR_HPMCOUNTER31        0xc1f
#define CSR_CYCLEH          0xc80
#define CSR_TIMEH           0xc81
#define CSR_INSTRETH            0xc82
#define CSR_HPMCOUNTER3H        0xc83
#define CSR_HPMCOUNTER4H        0xc84
#define CSR_HPMCOUNTER5H        0xc85
#define CSR_HPMCOUNTER6H        0xc86
#define CSR_HPMCOUNTER7H        0xc87
#define CSR_HPMCOUNTER8H        0xc88
#define CSR_HPMCOUNTER9H        0xc89
#define CSR_HPMCOUNTER10H       0xc8a
#define CSR_HPMCOUNTER11H       0xc8b
#define CSR_HPMCOUNTER12H       0xc8c
#define CSR_HPMCOUNTER13H       0xc8d
#define CSR_HPMCOUNTER14H       0xc8e
#define CSR_HPMCOUNTER15H       0xc8f
```

```
#define CSR_HPMCOUNTER16H       0xc90
#define CSR_HPMCOUNTER17H       0xc91
#define CSR_HPMCOUNTER18H       0xc92
#define CSR_HPMCOUNTER19H       0xc93
#define CSR_HPMCOUNTER20H       0xc94
#define CSR_HPMCOUNTER21H       0xc95
#define CSR_HPMCOUNTER22H       0xc96
#define CSR_HPMCOUNTER23H       0xc97
#define CSR_HPMCOUNTER24H       0xc98
#define CSR_HPMCOUNTER25H       0xc99
#define CSR_HPMCOUNTER26H       0xc9a
#define CSR_HPMCOUNTER27H       0xc9b
#define CSR_HPMCOUNTER28H       0xc9c
#define CSR_HPMCOUNTER29H       0xc9d
#define CSR_HPMCOUNTER30H       0xc9e
#define CSR_HPMCOUNTER31H       0xc9f

/* ===== Supervisor-level CSRs ===== */

/* Supervisor Trap Setup */
#define CSR_SSTATUS         0x100
#define CSR_SIE             0x104
#define CSR_STVEC           0x105
#define CSR_SCOUNTEREN          0x106

/* Supervisor Configuration */
#define CSR_SENVCFG         0x10a

/* Supervisor Conter Inhibit */
#define CSR_SCOUNTINHIBIT       0x120

/* Supervisor Trap Handling */
#define CSR_SSCRATCH            0x140
#define CSR_SEPC            0x141
#define CSR_SCAUSE          0x142
#define CSR_STVAL           0x143
#define CSR_SIP             0x144

/* Sstc extension */
#define CSR_STIMECMP            0x14D
#define CSR_STIMECMPH           0x15D

/* Supervisor Protection and Translation */
#define CSR_SATP            0x180

/* Supervisor Indirect Register Alias */
#define CSR_SISELECT            0x150
#define CSR_SIREG           0x151
#define CSR_SIREG2                  0x152
#define CSR_SIREG3                  0x153
#define CSR_SIREG4                  0x155
#define CSR_SIREG5                  0x156
#define CSR_SIREG6                  0x157

/* Supervisor-Level Interrupts (AIA) */
#define CSR_STOPEI          0x15c
#define CSR_STOPI           0xdb0

/* Supervisor-Level High-Half CSRs (AIA) */
#define CSR_SIEH            0x114
#define CSR_SIPH            0x154

/* Supervisor stateen CSRs */
#define CSR_SSTATEEN0           0x10C
#define CSR_SSTATEEN1           0x10D
#define CSR_SSTATEEN2           0x10E
#define CSR_SSTATEEN3           0x10F

/* ===== Hypervisor-level CSRs ===== */

/* Hypervisor Trap Setup (H-extension) */
#define CSR_HSTATUS         0x600
#define CSR_HEDELEG         0x602
#define CSR_HIDELEG         0x603
#define CSR_HIE             0x604
#define CSR_HCOUNTEREN          0x606
#define CSR_HGEIE           0x607

/* Hypervisor Configuration */
#define CSR_HENVCFG         0x60a
#define CSR_HENVCFGH            0x61a

/* Hypervisor Trap Handling (H-extension) */
#define CSR_HTVAL           0x643
#define CSR_HIP             0x644
#define CSR_HVIP            0x645
#define CSR_HTINST          0x64a
```

```
#define CSR_HGEIP           0xe12

/* Hypervisor Protection and Translation (H-extension) */
#define CSR_HGATP           0x680

/* Hypervisor Counter/Timer Virtualization Registers (H-extension) */
#define CSR_HTIMEDELTA          0x605
#define CSR_HTIMEDELTAH         0x615

/* Virtual Supervisor Registers (H-extension) */
#define CSR_VSSTATUS            0x200
#define CSR_VSIE            0x204
#define CSR_VSTVEC          0x205
#define CSR_VSSCRATCH           0x240
#define CSR_VSEPC           0x241
#define CSR_VSCAUSE         0x242
#define CSR_VSTVAL          0x243
#define CSR_VSIP            0x244
#define CSR_VSATP           0x280

/* Virtual Interrupts and Interrupt Priorities (H-extension with AIA) */
#define CSR_HVIEN           0x608
#define CSR_HVICTL          0x609
#define CSR_HVIPRIO1            0x646
#define CSR_HVIPRIO2            0x647

/* Virtual Supervisor Indirect Alias */
#define CSR_VSISELECT           0x250
#define CSR_VSIREG          0x251
#define CSR_VSIREG2                 0x252
#define CSR_VSIREG3                 0x253
#define CSR_VSIREG4                 0x255
#define CSR_VSIREG5                 0x256
#define CSR_VSIREG6                 0x257

/* VS-Level Interrupts (H-extension with AIA) */
#define CSR_VSTOPEI         0x25c
#define CSR_VSTOPI          0xeb0

/* Hypervisor and VS-Level High-Half CSRs (H-extension with AIA) */
#define CSR_HIDELEGH            0x613
#define CSR_HVIENH          0x618
#define CSR_HVIPH           0x655
#define CSR_HVIPRIO1H           0x656
#define CSR_HVIPRIO2H           0x657
#define CSR_VSIEH           0x214
#define CSR_VSIPH           0x254

/* Hypervisor stateen CSRs */
#define CSR_HSTATEEN0           0x60C
#define CSR_HSTATEEN0H          0x61C
#define CSR_HSTATEEN1           0x60D
#define CSR_HSTATEEN1H          0x61D
#define CSR_HSTATEEN2           0x60E
#define CSR_HSTATEEN2H          0x61E
#define CSR_HSTATEEN3           0x60F
#define CSR_HSTATEEN3H          0x61F

/* ===== Machine-level CSRs ===== */

/* Machine Information Registers */
#define CSR_MVENDORID           0xf11
#define CSR_MARCHID         0xf12
#define CSR_MIMPID          0xf13
#define CSR_MHARTID         0xf14
#define CSR_MCONFIGPTR          0xf15

/* Machine Trap Setup */
#define CSR_MSTATUS         0x300
#define CSR_MISA            0x301
#define CSR_MEDELEG         0x302
#define CSR_MIDELEG         0x303
#define CSR_MIE             0x304
#define CSR_MTVEC           0x305
#define CSR_MCOUNTEREN          0x306
#define CSR_MSTATUSH            0x310

/* Machine Configuration */
#define CSR_MENVCFG         0x30a
#define CSR_MENVCFGH            0x31a

/* Machine Trap Handling */
#define CSR_MSCRATCH            0x340
#define CSR_MEPC            0x341
#define CSR_MCAUSE          0x342
#define CSR_MTVAL           0x343
#define CSR_MIP             0x344
```

```
#define CSR_MTINST          0x34a
#define CSR_MTVAL2          0x34b

/* Machine Memory Protection */
#define CSR_PMPCFG0         0x3a0
#define CSR_PMPCFG1         0x3a1
#define CSR_PMPCFG2         0x3a2
#define CSR_PMPCFG3         0x3a3
#define CSR_PMPCFG4         0x3a4
#define CSR_PMPCFG5         0x3a5
#define CSR_PMPCFG6         0x3a6
#define CSR_PMPCFG7         0x3a7
#define CSR_PMPCFG8         0x3a8
#define CSR_PMPCFG9         0x3a9
#define CSR_PMPCFG10            0x3aa
#define CSR_PMPCFG11            0x3ab
#define CSR_PMPCFG12            0x3ac
#define CSR_PMPCFG13            0x3ad
#define CSR_PMPCFG14            0x3ae
#define CSR_PMPCFG15            0x3af
#define CSR_PMPADDR0            0x3b0
#define CSR_PMPADDR1            0x3b1
#define CSR_PMPADDR2            0x3b2
#define CSR_PMPADDR3            0x3b3
#define CSR_PMPADDR4            0x3b4
#define CSR_PMPADDR5            0x3b5
#define CSR_PMPADDR6            0x3b6
#define CSR_PMPADDR7            0x3b7
#define CSR_PMPADDR8            0x3b8
#define CSR_PMPADDR9            0x3b9
#define CSR_PMPADDR10           0x3ba
#define CSR_PMPADDR11           0x3bb
#define CSR_PMPADDR12           0x3bc
#define CSR_PMPADDR13           0x3bd
#define CSR_PMPADDR14           0x3be
#define CSR_PMPADDR15           0x3bf
#define CSR_PMPADDR16           0x3c0
#define CSR_PMPADDR17           0x3c1
#define CSR_PMPADDR18           0x3c2
#define CSR_PMPADDR19           0x3c3
#define CSR_PMPADDR20           0x3c4
#define CSR_PMPADDR21           0x3c5
#define CSR_PMPADDR22           0x3c6
#define CSR_PMPADDR23           0x3c7
#define CSR_PMPADDR24           0x3c8
#define CSR_PMPADDR25           0x3c9
#define CSR_PMPADDR26           0x3ca
#define CSR_PMPADDR27           0x3cb
#define CSR_PMPADDR28           0x3cc
#define CSR_PMPADDR29           0x3cd
#define CSR_PMPADDR30           0x3ce
#define CSR_PMPADDR31           0x3cf
#define CSR_PMPADDR32           0x3d0
#define CSR_PMPADDR33           0x3d1
#define CSR_PMPADDR34           0x3d2
#define CSR_PMPADDR35           0x3d3
#define CSR_PMPADDR36           0x3d4
#define CSR_PMPADDR37           0x3d5
#define CSR_PMPADDR38           0x3d6
#define CSR_PMPADDR39           0x3d7
#define CSR_PMPADDR40           0x3d8
#define CSR_PMPADDR41           0x3d9
#define CSR_PMPADDR42           0x3da
#define CSR_PMPADDR43           0x3db
#define CSR_PMPADDR44           0x3dc
#define CSR_PMPADDR45           0x3dd
#define CSR_PMPADDR46           0x3de
#define CSR_PMPADDR47           0x3df
#define CSR_PMPADDR48           0x3e0
#define CSR_PMPADDR49           0x3e1
#define CSR_PMPADDR50           0x3e2
#define CSR_PMPADDR51           0x3e3
#define CSR_PMPADDR52           0x3e4
#define CSR_PMPADDR53           0x3e5
#define CSR_PMPADDR54           0x3e6
#define CSR_PMPADDR55           0x3e7
#define CSR_PMPADDR56           0x3e8
#define CSR_PMPADDR57           0x3e9
#define CSR_PMPADDR58           0x3ea
#define CSR_PMPADDR59           0x3eb
#define CSR_PMPADDR60           0x3ec
#define CSR_PMPADDR61           0x3ed
#define CSR_PMPADDR62           0x3ee
#define CSR_PMPADDR63           0x3ef

/* Machine Counters/Timers */
#define CSR_MCYCLE          0xb00
```

```
#define CSR_MINSTRET            0xb02
#define CSR_MHPMCOUNTER3        0xb03
#define CSR_MHPMCOUNTER4        0xb04
#define CSR_MHPMCOUNTER5        0xb05
#define CSR_MHPMCOUNTER6        0xb06
#define CSR_MHPMCOUNTER7        0xb07
#define CSR_MHPMCOUNTER8        0xb08
#define CSR_MHPMCOUNTER9        0xb09
#define CSR_MHPMCOUNTER10       0xb0a
#define CSR_MHPMCOUNTER11       0xb0b
#define CSR_MHPMCOUNTER12       0xb0c
#define CSR_MHPMCOUNTER13       0xb0d
#define CSR_MHPMCOUNTER14       0xb0e
#define CSR_MHPMCOUNTER15       0xb0f
#define CSR_MHPMCOUNTER16       0xb10
#define CSR_MHPMCOUNTER17       0xb11
#define CSR_MHPMCOUNTER18       0xb12
#define CSR_MHPMCOUNTER19       0xb13
#define CSR_MHPMCOUNTER20       0xb14
#define CSR_MHPMCOUNTER21       0xb15
#define CSR_MHPMCOUNTER22       0xb16
#define CSR_MHPMCOUNTER23       0xb17
#define CSR_MHPMCOUNTER24       0xb18
#define CSR_MHPMCOUNTER25       0xb19
#define CSR_MHPMCOUNTER26       0xb1a
#define CSR_MHPMCOUNTER27       0xb1b
#define CSR_MHPMCOUNTER28       0xb1c
#define CSR_MHPMCOUNTER29       0xb1d
#define CSR_MHPMCOUNTER30       0xb1e
#define CSR_MHPMCOUNTER31       0xb1f
#define CSR_MCYCLEH         0xb80
#define CSR_MINSTRETH           0xb82
#define CSR_MHPMCOUNTER3H       0xb83
#define CSR_MHPMCOUNTER4H       0xb84
#define CSR_MHPMCOUNTER5H       0xb85
#define CSR_MHPMCOUNTER6H       0xb86
#define CSR_MHPMCOUNTER7H       0xb87
#define CSR_MHPMCOUNTER8H       0xb88
#define CSR_MHPMCOUNTER9H       0xb89
#define CSR_MHPMCOUNTER10H      0xb8a
#define CSR_MHPMCOUNTER11H      0xb8b
#define CSR_MHPMCOUNTER12H      0xb8c
#define CSR_MHPMCOUNTER13H      0xb8d
#define CSR_MHPMCOUNTER14H      0xb8e
#define CSR_MHPMCOUNTER15H      0xb8f
#define CSR_MHPMCOUNTER16H      0xb90
#define CSR_MHPMCOUNTER17H      0xb91
#define CSR_MHPMCOUNTER18H      0xb92
#define CSR_MHPMCOUNTER19H      0xb93
#define CSR_MHPMCOUNTER20H      0xb94
#define CSR_MHPMCOUNTER21H      0xb95
#define CSR_MHPMCOUNTER22H      0xb96
#define CSR_MHPMCOUNTER23H      0xb97
#define CSR_MHPMCOUNTER24H      0xb98
#define CSR_MHPMCOUNTER25H      0xb99
#define CSR_MHPMCOUNTER26H      0xb9a
#define CSR_MHPMCOUNTER27H      0xb9b
#define CSR_MHPMCOUNTER28H      0xb9c
#define CSR_MHPMCOUNTER29H      0xb9d
#define CSR_MHPMCOUNTER30H      0xb9e
#define CSR_MHPMCOUNTER31H      0xb9f

/* Machine Counter Setup */
#define CSR_MCOUNTINHIBIT       0x320
#define CSR_MCYCLECFG           0x321
#define CSR_MINSTRETCFG         0x322
#define CSR_MHPMEVENT3          0x323
#define CSR_MHPMEVENT4          0x324
#define CSR_MHPMEVENT5          0x325
#define CSR_MHPMEVENT6          0x326
#define CSR_MHPMEVENT7          0x327
#define CSR_MHPMEVENT8          0x328
#define CSR_MHPMEVENT9          0x329
#define CSR_MHPMEVENT10         0x32a
#define CSR_MHPMEVENT11         0x32b
#define CSR_MHPMEVENT12         0x32c
#define CSR_MHPMEVENT13         0x32d
#define CSR_MHPMEVENT14         0x32e
#define CSR_MHPMEVENT15         0x32f
#define CSR_MHPMEVENT16         0x330
#define CSR_MHPMEVENT17         0x331
#define CSR_MHPMEVENT18         0x332
#define CSR_MHPMEVENT19         0x333
#define CSR_MHPMEVENT20         0x334
#define CSR_MHPMEVENT21         0x335
#define CSR_MHPMEVENT22         0x336
#define CSR_MHPMEVENT23         0x337
```

```
#define CSR_MHPMEVENT24         0x338
#define CSR_MHPMEVENT25         0x339
#define CSR_MHPMEVENT26         0x33a
#define CSR_MHPMEVENT27         0x33b
#define CSR_MHPMEVENT28         0x33c
#define CSR_MHPMEVENT29         0x33d
#define CSR_MHPMEVENT30         0x33e
#define CSR_MHPMEVENT31         0x33f

/* For RV32 */
#define CSR_MCYCLECFGH          0x721
#define CSR_MINSTRETCFGH        0x722
#define CSR_MHPMEVENT3H         0x723
#define CSR_MHPMEVENT4H         0x724
#define CSR_MHPMEVENT5H         0x725
#define CSR_MHPMEVENT6H         0x726
#define CSR_MHPMEVENT7H         0x727
#define CSR_MHPMEVENT8H         0x728
#define CSR_MHPMEVENT9H         0x729
#define CSR_MHPMEVENT10H        0x72a
#define CSR_MHPMEVENT11H        0x72b
#define CSR_MHPMEVENT12H        0x72c
#define CSR_MHPMEVENT13H        0x72d
#define CSR_MHPMEVENT14H        0x72e
#define CSR_MHPMEVENT15H        0x72f
#define CSR_MHPMEVENT16H        0x730
#define CSR_MHPMEVENT17H        0x731
#define CSR_MHPMEVENT18H        0x732
#define CSR_MHPMEVENT19H        0x733
#define CSR_MHPMEVENT20H        0x734
#define CSR_MHPMEVENT21H        0x735
#define CSR_MHPMEVENT22H        0x736
#define CSR_MHPMEVENT23H        0x737
#define CSR_MHPMEVENT24H        0x738
#define CSR_MHPMEVENT25H        0x739
#define CSR_MHPMEVENT26H        0x73a
#define CSR_MHPMEVENT27H        0x73b
#define CSR_MHPMEVENT28H        0x73c
#define CSR_MHPMEVENT29H        0x73d
#define CSR_MHPMEVENT30H        0x73e
#define CSR_MHPMEVENT31H        0x73f

/* Machine Security Configuration CSR (mseccfg) */
#define CSR_MSECCFG         0x747
#define CSR_MSECCFGH            0x757

#define MSECCFG_MML_SHIFT       (0)
#define MSECCFG_MML         (_UL(1) << MSECCFG_MML_SHIFT)
#define MSECCFG_MMWP_SHIFT      (1)
#define MSECCFG_MMWP            (_UL(1) << MSECCFG_MMWP_SHIFT)
#define MSECCFG_RLB_SHIFT       (2)
#define MSECCFG_RLB         (_UL(1) << MSECCFG_RLB_SHIFT)
#define MSECCFG_USEED_SHIFT     (8)
#define MSECCFG_USEED           (_UL(1) << MSECCFG_USEED_SHIFT)
#define MSECCFG_SSEED_SHIFT     (9)
#define MSECCFG_SSEED           (_UL(1) << MSECCFG_SSEED_SHIFT)

/* Counter Overflow CSR */
#define CSR_SCOUNTOVF           0xda0

/* Debug/Trace Registers */
#define CSR_TSELECT         0x7a0
#define CSR_TDATA1          0x7a1
#define CSR_TDATA2          0x7a2
#define CSR_TDATA3          0x7a3
#define CSR_TINFO           0x7a4

/* Debug Mode Registers */
#define CSR_DCSR            0x7b0
#define CSR_DPC             0x7b1
#define CSR_DSCRATCH0           0x7b2
#define CSR_DSCRATCH1           0x7b3

/* Machine Indirect Register Alias */
#define CSR_MISELECT            0x350
#define CSR_MIREG           0x351
#define CSR_MIREG2                  0x352
#define CSR_MIREG3                  0x353
#define CSR_MIREG4                  0x355
#define CSR_MIREG5                  0x356
#define CSR_MIREG6                  0x357

/* Machine-Level Interrupts (AIA) */
#define CSR_MTOPEI          0x35c
#define CSR_MTOPI           0xfb0

/* Virtual Interrupts for Supervisor Level (AIA) */
```

```
#define CSR_MVIEN           0x308
#define CSR_MVIP            0x309

/* Smstateen extension registers */
/* Machine stateen CSRs */
#define CSR_MSTATEEN0           0x30C
#define CSR_MSTATEEN0H          0x31C
#define CSR_MSTATEEN1           0x30D
#define CSR_MSTATEEN1H          0x31D
#define CSR_MSTATEEN2           0x30E
#define CSR_MSTATEEN2H          0x31E
#define CSR_MSTATEEN3           0x30F
#define CSR_MSTATEEN3H          0x31F

/* Machine-Level High-Half CSRs (AIA) */
#define CSR_MIDELEGH            0x313
#define CSR_MIEH            0x314
#define CSR_MVIENH          0x318
#define CSR_MVIPH           0x319
#define CSR_MIPH            0x354

/* Vector extension registers */
#define CSR_VSTART          0x8
#define CSR_VL              0xc20
#define CSR_VTYPE           0xc21
#define CSR_VLENB           0xc22

/* ===== Trap/Exception Causes ===== */

#define CAUSE_MISALIGNED_FETCH      0x0
#define CAUSE_FETCH_ACCESS      0x1
#define CAUSE_ILLEGAL_INSTRUCTION   0x2
#define CAUSE_BREAKPOINT        0x3
#define CAUSE_MISALIGNED_LOAD       0x4
#define CAUSE_LOAD_ACCESS       0x5
#define CAUSE_MISALIGNED_STORE      0x6
#define CAUSE_STORE_ACCESS      0x7
#define CAUSE_USER_ECALL        0x8
#define CAUSE_SUPERVISOR_ECALL      0x9
#define CAUSE_VIRTUAL_SUPERVISOR_ECALL  0xa
#define CAUSE_MACHINE_ECALL     0xb
#define CAUSE_FETCH_PAGE_FAULT      0xc
#define CAUSE_LOAD_PAGE_FAULT       0xd
#define CAUSE_STORE_PAGE_FAULT      0xf
#define CAUSE_DOUBLE_TRAP       0x10
#define CAUSE_SW_CHECK_EXCP     0x12
#define CAUSE_FETCH_GUEST_PAGE_FAULT    0x14
#define CAUSE_LOAD_GUEST_PAGE_FAULT 0x15
#define CAUSE_VIRTUAL_INST_FAULT    0x16
#define CAUSE_STORE_GUEST_PAGE_FAULT    0x17

/* Common defines for all smstateen */
#define SMSTATEEN_MAX_COUNT     4
#define SMSTATEEN0_CS_SHIFT     0
#define SMSTATEEN0_CS           (_ULL(1) « SMSTATEEN0_CS_SHIFT)
#define SMSTATEEN0_FCSR_SHIFT       1
#define SMSTATEEN0_FCSR         (_ULL(1) « SMSTATEEN0_FCSR_SHIFT)
#define SMSTATEEN0_CONTEXT_SHIFT    57
#define SMSTATEEN0_CONTEXT      (_ULL(1) « SMSTATEEN0_CONTEXT_SHIFT)
#define SMSTATEEN0_IMSIC_SHIFT      58
#define SMSTATEEN0_IMSIC        (_ULL(1) « SMSTATEEN0_IMSIC_SHIFT)
#define SMSTATEEN0_AIA_SHIFT        59
#define SMSTATEEN0_AIA          (_ULL(1) « SMSTATEEN0_AIA_SHIFT)
#define SMSTATEEN0_SVSLCT_SHIFT     60
#define SMSTATEEN0_SVSLCT       (_ULL(1) « SMSTATEEN0_SVSLCT_SHIFT)
#define SMSTATEEN0_HSENVCFG_SHIFT   62
#define SMSTATEEN0_HSENVCFG     (_ULL(1) « SMSTATEEN0_HSENVCFG_SHIFT)
#define SMSTATEEN_STATEN_SHIFT      63
#define SMSTATEEN_STATEN        (_ULL(1) « SMSTATEEN_STATEN_SHIFT)

/* ===== Instruction Encodings ===== */

#define INSN_MATCH_LB           0x3
#define INSN_MASK_LB            0x707f
#define INSN_MATCH_LH           0x1003
#define INSN_MASK_LH            0x707f
#define INSN_MATCH_LW           0x2003
#define INSN_MASK_LW            0x707f
#define INSN_MATCH_LD           0x3003
#define INSN_MASK_LD            0x707f
#define INSN_MATCH_LBU          0x4003
#define INSN_MASK_LBU           0x707f
#define INSN_MATCH_LHU          0x5003
#define INSN_MASK_LHU           0x707f
#define INSN_MATCH_LWU          0x6003
#define INSN_MASK_LWU           0x707f
#define INSN_MATCH_SB           0x23
```

```
#define INSN_MASK_SB            0x707f
#define INSN_MATCH_SH           0x1023
#define INSN_MASK_SH            0x707f
#define INSN_MATCH_SW           0x2023
#define INSN_MASK_SW            0x707f
#define INSN_MATCH_SD           0x3023
#define INSN_MASK_SD            0x707f

#define INSN_MATCH_FLW          0x2007
#define INSN_MASK_FLW           0x707f
#define INSN_MATCH_FLD          0x3007
#define INSN_MASK_FLD           0x707f
#define INSN_MATCH_FLQ          0x4007
#define INSN_MASK_FLQ           0x707f
#define INSN_MATCH_FSW          0x2027
#define INSN_MASK_FSW           0x707f
#define INSN_MATCH_FSD          0x3027
#define INSN_MASK_FSD           0x707f
#define INSN_MATCH_FSQ          0x4027
#define INSN_MASK_FSQ           0x707f

#define INSN_MATCH_C_LD         0x6000
#define INSN_MASK_C_LD          0xe003
#define INSN_MATCH_C_SD         0xe000
#define INSN_MASK_C_SD          0xe003
#define INSN_MATCH_C_LW         0x4000
#define INSN_MASK_C_LW          0xe003
#define INSN_MATCH_C_SW         0xc000
#define INSN_MASK_C_SW          0xe003
#define INSN_MATCH_C_LDSP       0x6002
#define INSN_MASK_C_LDSP        0xe003
#define INSN_MATCH_C_SDSP       0xe002
#define INSN_MASK_C_SDSP        0xe003
#define INSN_MATCH_C_LWSP       0x4002
#define INSN_MASK_C_LWSP        0xe003
#define INSN_MATCH_C_SWSP       0xc002
#define INSN_MASK_C_SWSP        0xe003

#define INSN_MATCH_C_FLD        0x2000
#define INSN_MASK_C_FLD         0xe003
#define INSN_MATCH_C_FLW        0x6000
#define INSN_MASK_C_FLW         0xe003
#define INSN_MATCH_C_FSD        0xa000
#define INSN_MASK_C_FSD         0xe003
#define INSN_MATCH_C_FSW        0xe000
#define INSN_MASK_C_FSW         0xe003
#define INSN_MATCH_C_FLDSP      0x2002
#define INSN_MASK_C_FLDSP       0xe003
#define INSN_MATCH_C_FSDSP      0xa002
#define INSN_MASK_C_FSDSP       0xe003
#define INSN_MATCH_C_FLWSP      0x6002
#define INSN_MASK_C_FLWSP       0xe003
#define INSN_MATCH_C_FSWSP      0xe002
#define INSN_MASK_C_FSWSP       0xe003

#define INSN_MATCH_C_LHU        0x8400
#define INSN_MASK_C_LHU     0xfc43
#define INSN_MATCH_C_LH     0x8440
#define INSN_MASK_C_LH          0xfc43
#define INSN_MATCH_C_SH     0x8c00
#define INSN_MASK_C_SH          0xfc43

#define INSN_MASK_WFI           0xffffff00
#define INSN_MATCH_WFI          0x10500000

#define INSN_MASK_FENCE_TSO     0xffffffff
#define INSN_MATCH_FENCE_TSO        0x8330000f

#define INSN_MASK_VECTOR_UNIT_STRIDE        0xfdf0707f
#define INSN_MASK_VECTOR_FAULT_ONLY_FIRST   0xfdf0707f
#define INSN_MASK_VECTOR_STRIDE         0xfc00707f
#define INSN_MASK_VECTOR_WHOLE_REG      0xfff0707f
#define INSN_MASK_VECTOR_INDEXED        0xfc00707f

#define INSN_MATCH_VLUXSEG(n, bits) ((((n) - 1) « 29) | 0x04000007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
#define INSN_MATCH_VSUXSEG(n, bits) ((((n) - 1) « 29) | 0x04000027 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
#define INSN_MATCH_VLOXSEG(n, bits) ((((n) - 1) « 29) | 0x0c000007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
#define INSN_MATCH_VSOXSEG(n, bits) ((((n) - 1) « 29) | 0x0c000027 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
#define INSN_MATCH_VLSSEG(n, bits)  ((((n) - 1) « 29) | 0x08000007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
#define INSN_MATCH_VSSSEG(n, bits)  ((((n) - 1) « 29) | 0x08000027 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
#define INSN_MATCH_VSSEG(n, bits)   ((((n) - 1) « 29) | 0x00004027 | \
```

```
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
#define INSN_MATCH_VLSEG(n, bits)    ((((n) - 1) « 29) | 0x00004007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)
#define INSN_MATCH_VLSEGFF(n, bits) ((((n) - 1) « 29) | 0x1000007 | \
        ((bits) == 16 ? 5 : (bits) == 32 ? 6 : 7) « 12)

#define INSN_MATCH_VLE16V       0x00005007
#define INSN_MATCH_VLE32V       0x00006007
#define INSN_MATCH_VLE64V       0x00007007
#define INSN_MATCH_VSE16V       0x00005027
#define INSN_MATCH_VSE32V       0x00006027
#define INSN_MATCH_VSE64V       0x00007027
#define INSN_MATCH_VLSE16V      0x08005007
#define INSN_MATCH_VLSE32V      0x08006007
#define INSN_MATCH_VLSE64V      0x08007007
#define INSN_MATCH_VSSE16V      0x08005027
#define INSN_MATCH_VSSE32V      0x08006027
#define INSN_MATCH_VSSE64V      0x08007027
#define INSN_MATCH_VLOXEI16V        0x0c005007
#define INSN_MATCH_VLOXEI32V        0x0c006007
#define INSN_MATCH_VLOXEI64V        0x0c007007
#define INSN_MATCH_VSOXEI16V        0x0c005027
#define INSN_MATCH_VSOXEI32V        0x0c006027
#define INSN_MATCH_VSOXEI64V        0x0c007027
#define INSN_MATCH_VLUXEI16V        0x04005007
#define INSN_MATCH_VLUXEI32V        0x04006007
#define INSN_MATCH_VLUXEI64V        0x04007007
#define INSN_MATCH_VSUXEI16V        0x04005027
#define INSN_MATCH_VSUXEI32V        0x04006027
#define INSN_MATCH_VSUXEI64V        0x04007027
#define INSN_MATCH_VLE16FFV     0x01005007
#define INSN_MATCH_VLE32FFV     0x01006007
#define INSN_MATCH_VLE64FFV     0x01007007
#define INSN_MATCH_VL1RE8V      0x02800007
#define INSN_MATCH_VL1RE16V     0x02805007
#define INSN_MATCH_VL1RE32V     0x02806007
#define INSN_MATCH_VL1RE64V     0x02807007
#define INSN_MATCH_VL2RE8V      0x22800007
#define INSN_MATCH_VL2RE16V     0x22805007
#define INSN_MATCH_VL2RE32V     0x22806007
#define INSN_MATCH_VL2RE64V     0x22807007
#define INSN_MATCH_VL4RE8V      0x62800007
#define INSN_MATCH_VL4RE16V     0x62805007
#define INSN_MATCH_VL4RE32V     0x62806007
#define INSN_MATCH_VL4RE64V     0x62807007
#define INSN_MATCH_VL8RE8V      0xe2800007
#define INSN_MATCH_VL8RE16V     0xe2805007
#define INSN_MATCH_VL8RE32V     0xe2806007
#define INSN_MATCH_VL8RE64V     0xe2807007
#define INSN_MATCH_VS1RV        0x02800027
#define INSN_MATCH_VS2RV        0x22800027
#define INSN_MATCH_VS4RV        0x62800027
#define INSN_MATCH_VS8RV        0xe2800027

#define INSN_MASK_VECTOR_LOAD_STORE 0x7f
#define INSN_MATCH_VECTOR_LOAD      0x07
#define INSN_MATCH_VECTOR_STORE     0x27

#define IS_VECTOR_LOAD_STORE(insn) \
    ((((insn) & INSN_MASK_VECTOR_LOAD_STORE) == INSN_MATCH_VECTOR_LOAD) || \
    (((insn) & INSN_MASK_VECTOR_LOAD_STORE) == INSN_MATCH_VECTOR_STORE))

#define IS_VECTOR_INSN_MATCH(insn, match, mask) \
    (((insn) & (mask)) == ((match) & (mask)))

#define IS_UNIT_STRIDE_MATCH(insn, match) \
    IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_UNIT_STRIDE)

#define IS_STRIDE_MATCH(insn, match) \
    IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_STRIDE)

#define IS_INDEXED_MATCH(insn, match) \
    IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_INDEXED)

#define IS_FAULT_ONLY_FIRST_MATCH(insn, match) \
    IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_FAULT_ONLY_FIRST)

#define IS_WHOLE_REG_MATCH(insn, match) \
    IS_VECTOR_INSN_MATCH(insn, match, INSN_MASK_VECTOR_WHOLE_REG)

#define IS_UNIT_STRIDE_LOAD(insn) ( \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLE16V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLE32V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLE64V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(2, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(3, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(4, 16)) || \
```

```
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(5, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(6, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(7, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(8, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(2, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(3, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(4, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(5, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(6, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(7, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(8, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(2, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(3, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(4, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(5, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(6, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(7, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VLSEG(8, 64)))

#define IS_UNIT_STRIDE_STORE(insn) ( \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSE16V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSE32V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSE64V) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(2, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(3, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(4, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(5, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(6, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(7, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(8, 16)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(2, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(3, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(4, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(5, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(6, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(7, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(8, 32)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(2, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(3, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(4, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(5, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(6, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(7, 64)) || \
    IS_UNIT_STRIDE_MATCH(insn, INSN_MATCH_VSSEG(8, 64)))

#define IS_STRIDE_LOAD(insn) ( \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSE16V) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSE32V) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSE64V) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(2, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(3, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(4, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(5, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(6, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(7, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(8, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(2, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(3, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(4, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(5, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(6, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(7, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(8, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(2, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(3, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(4, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(5, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(6, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(7, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VLSSEG(8, 64)))

#define IS_STRIDE_STORE(insn) ( \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSE16V) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSE32V) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSE64V) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(2, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(3, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(4, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(5, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(6, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(7, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(8, 16)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(2, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(3, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(4, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(5, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(6, 32)) || \
```

```
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(7, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(8, 32)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(2, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(3, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(4, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(5, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(6, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(7, 64)) || \
    IS_STRIDE_MATCH(insn, INSN_MATCH_VSSSEG(8, 64)))

#define IS_INDEXED_LOAD(insn) ( \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXEI16V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXEI32V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXEI64V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXEI16V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXEI32V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXEI64V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(2, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(3, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(4, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(5, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(6, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(7, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(8, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(2, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(3, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(4, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(5, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(6, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(7, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(8, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(2, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(3, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(4, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(5, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(6, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(7, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLUXSEG(8, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(2, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(3, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(4, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(5, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(6, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(7, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(8, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(2, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(3, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(4, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(5, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(6, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(7, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(8, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(2, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(3, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(4, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(5, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(6, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(7, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VLOXSEG(8, 64)))

#define IS_INDEXED_STORE(insn) ( \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXEI16V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXEI32V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXEI64V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXEI16V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXEI32V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXEI64V) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(2, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(3, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(4, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(5, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(6, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(7, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(8, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(2, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(3, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(4, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(5, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(6, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(7, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(8, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(2, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(3, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(4, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(5, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(6, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(7, 64)) || \
```

```
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSUXSEG(8, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(2, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(3, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(4, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(5, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(6, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(7, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(8, 16)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(2, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(3, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(4, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(5, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(6, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(7, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(8, 32)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(2, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(3, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(4, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(5, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(6, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(7, 64)) || \
    IS_INDEXED_MATCH(insn, INSN_MATCH_VSOXSEG(8, 64)))

#define IS_FAULT_ONLY_FIRST_LOAD(insn) ( \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLE16FFV) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLE32FFV) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLE64FFV) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(2, 16)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(3, 16)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(4, 16)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(5, 16)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(6, 16)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(7, 16)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(8, 16)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(2, 32)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(3, 32)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(4, 32)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(5, 32)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(6, 32)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(7, 32)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(8, 32)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(2, 64)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(3, 64)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(4, 64)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(5, 64)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(6, 64)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(7, 64)) || \
    IS_FAULT_ONLY_FIRST_MATCH(insn, INSN_MATCH_VLSEGFF(8, 64)))

    #define IS_WHOLE_REG_LOAD(insn) ( \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL1RE8V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL1RE16V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL1RE32V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL1RE64V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL2RE8V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL2RE16V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL2RE32V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL2RE64V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL4RE8V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL4RE16V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL4RE32V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL4RE64V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL8RE8V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL8RE16V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL8RE32V) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VL8RE64V))

#define IS_WHOLE_REG_STORE(insn) ( \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VS1RV) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VS2RV) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VS4RV) || \
    IS_WHOLE_REG_MATCH(insn, INSN_MATCH_VS8RV))


#if __riscv_xlen == 64

/* 64-bit read for VS-stage address translation (RV64) */
#define INSN_PSEUDO_VS_LOAD     0x00003000
/* 64-bit write for VS-stage address translation (RV64) */
#define INSN_PSEUDO_VS_STORE    0x00003020

#elif __riscv_xlen == 32

/* 32-bit read for VS-stage address translation (RV32) */
#define INSN_PSEUDO_VS_LOAD     0x00002000

/* 32-bit write for VS-stage address translation (RV32) */
```

```
#define INSN_PSEUDO_VS_STORE    0x00002020

#else
#error "Unexpected __riscv_xlen"
#endif

#define VM_MASK             0x1
#define VIEW_MASK               0x3
#define VSEW_MASK               0x3
#define VLMUL_MASK              0x7
#define VD_MASK             0x1f
#define VS2_MASK                0x1f
#define INSN_16BIT_MASK         0x3
#define INSN_32BIT_MASK         0x1c

#define INSN_IS_16BIT(insn)     \
    (((insn) & INSN_16BIT_MASK) != INSN_16BIT_MASK)
#define INSN_IS_32BIT(insn)     \
    (((insn) & INSN_16BIT_MASK) == INSN_16BIT_MASK && \
     ((insn) & INSN_32BIT_MASK) != INSN_32BIT_MASK)

#define INSN_LEN(insn)          (INSN_IS_16BIT(insn) ? 2 : 4)

#if __riscv_xlen == 64
#define LOG_REGBYTES            3
#else
#define LOG_REGBYTES            2
#endif
#define REGBYTES            (1 << LOG_REGBYTES)

#define SH_VSEW         3
#define SH_VIEW         12
#define SH_VD               7
#define SH_VS2              20
#define SH_VM               25
#define SH_MEW              28
#define SH_RD               7
#define SH_RS1              15
#define SH_RS2              20
#define SH_RS2C             2

#define RV_X(x, s, n)           (((x) >> (s)) & ((1 << (n)) - 1))
#define RVC_LW_IMM(x)           ((RV_X(x, 6, 1) << 2) | \
                        (RV_X(x, 10, 3) << 3) | \
                        (RV_X(x, 5, 1) << 6))
#define RVC_LD_IMM(x)           ((RV_X(x, 10, 3) << 3) | \
                        (RV_X(x, 5, 2) << 6))
#define RVC_LWSP_IMM(x)         ((RV_X(x, 4, 3) << 2) | \
                        (RV_X(x, 12, 1) << 5) | \
                        (RV_X(x, 2, 2) << 6))
#define RVC_LDSP_IMM(x)         ((RV_X(x, 5, 2) << 3) | \
                        (RV_X(x, 12, 1) << 5) | \
                        (RV_X(x, 2, 3) << 6))
#define RVC_SWSP_IMM(x)         ((RV_X(x, 9, 4) << 2) | \
                        (RV_X(x, 7, 2) << 6))
#define RVC_SDSP_IMM(x)         ((RV_X(x, 10, 3) << 3) | \
                        (RV_X(x, 7, 3) << 6))
#define RVC_RS1S(insn)          (8 + RV_X(insn, SH_RD, 3))
#define RVC_RS2S(insn)          (8 + RV_X(insn, SH_RS2C, 3))
#define RVC_RS2(insn)           RV_X(insn, SH_RS2C, 5)

#define SHIFT_RIGHT(x, y)       \
    ((y) < 0 ? ((x) << -(y)) : ((x) >> (y)))

#define REG_MASK            \
    ((1 << (5 + LOG_REGBYTES)) - (1 << LOG_REGBYTES))

#define REG_OFFSET(insn, pos)       \
    (SHIFT_RIGHT((insn), (pos) - LOG_REGBYTES) & REG_MASK)

#define REG_PTR(insn, pos, regs)    \
    (ulong *)((ulong)(regs) + REG_OFFSET(insn, pos))

#define GET_RM(insn)            ((insn & MASK_FUNCT3) >> SHIFT_FUNCT3)

#define GET_RS1_NUM(insn)       ((insn & MASK_RS1) >> 15)
#define GET_CSR_NUM(insn)       ((insn & MASK_CSR) >> SHIFT_CSR)

#define GET_RS1(insn, regs)     (*REG_PTR(insn, SH_RS1, regs))
#define GET_RS2(insn, regs)     (*REG_PTR(insn, SH_RS2, regs))
#define GET_RS1S(insn, regs)        (*REG_PTR(RVC_RS1S(insn), 0, regs))
#define GET_RS2S(insn, regs)        (*REG_PTR(RVC_RS2S(insn), 0, regs))
#define GET_RS2C(insn, regs)        (*REG_PTR(insn, SH_RS2C, regs))
#define GET_SP(regs)            (*REG_PTR(2, 0, regs))
#define SET_RD(insn, regs, val)     (*REG_PTR(insn, SH_RD, regs) = (val))
#define IMM_I(insn)         ((s32)(insn) >> 20)
#define IMM_S(insn)         (((s32)(insn) >> 25 << 5) | \
```

```
                    (s32)(((insn) » 7) & 0x1f))

#define IS_MASKED(insn)      (((insn » SH_VM) & VM_MASK) == 0)
#define GET_VD(insn)             ((insn » SH_VD) & VD_MASK)
#define GET_VS2(insn)            ((insn » SH_VS2) & VS2_MASK)
#define GET_VIEW(insn)           (((insn) » SH_VIEW) & VIEW_MASK)
#define GET_MEW(insn)            (((insn) » SH_MEW) & 1)
#define GET_VSEW(vtype)      (((vtype) » SH_VSEW) & VSEW_MASK)
#define GET_VLMUL(vtype)         ((vtype) & VLMUL_MASK)
#define GET_LEN(view)            (1UL « (view))
#define GET_NF(insn)             (1 + ((insn » 29) & 7))
#define GET_VEMUL(vlmul, view, vsew)    ((vlmul + view - vsew) & 7)
#define GET_EMUL(vemul)      (1UL « ((vemul) >= 4 ? 0 : (vemul)))

#define MASK_FUNCT3          0x7000
#define MASK_RS1             0xf8000
#define MASK_CSR             0xfff00000

#define SHIFT_FUNCT3             12
#define SHIFT_CSR            20

#define CSRRW 1
#define CSRRS 2
#define CSRRC 3
#define CSRRWI 5
#define CSRRSI 6
#define CSRRCI 7

/* clang-format on */

#endif
```

30.40 include/sbi/riscv_fp.h File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for riscv_fp.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define GET_PRECISION(insn) (((insn) >> 25) & 3)
- #define PRECISION_S 0
- #define PRECISION_D 1

30.40.1 Macro Definition Documentation

30.40.1.1 GET_PRECISION

```
#define GET_PRECISION(
            insn ) (((insn) >> 25) & 3)
```

30.40.1.2 PRECISION_D

```
#define PRECISION_D 1
```

30.40.1.3 PRECISION_S

```
#define PRECISION_S 0
```

30.41 riscv_fp.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __RISCV_FP_H__
#define __RISCV_FP_H__

#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_types.h>

#define GET_PRECISION(insn) (((insn) » 25) & 3)
#define PRECISION_S 0
#define PRECISION_D 1

#ifdef __riscv_flen

#define GET_F32_REG(insn, pos, regs)
\
    ({                                                                                  \
        register s32 value asm("a0") =                                                  \
```

```
            SHIFT_RIGHT(insn, (pos)-3) & 0xf8;                                          \
        ulong tmp;                                                                      \
        asm("1: auipc %0, %%pcrel_hi(get_f32_reg); add %0, %0, %1; jalr t0, %0, %%pcrel_lo(1b)" \
            : "=&r"(tmp), "+&r"(value)::"t0");                                          \
        value;                                                                          \
    })
#define SET_F32_REG(insn, pos, regs, val) \
    ({ \
        register u32 value asm("a0") = (val);                                           \
        ulong offset = SHIFT_RIGHT(insn, (pos)-3) & 0xf8;                               \
        ulong tmp;                                                                      \
        asm volatile(                                                                   \
            "1: auipc %0, %%pcrel_hi(put_f32_reg); add %0, %0, %2; jalr t0, %0, %%pcrel_lo(1b)" \
            : "=&r"(tmp)                                                                \
            : "r"(value), "r"(offset)                                                   \
            : "t0");                                                                    \
    })
#define init_fp_reg(i) SET_F32_REG((i) « 3, 3, 0, 0)

#if __riscv_xlen == 64
#define GET_F64_REG(insn, pos, regs) \
    ({                                                                                  \
        register ulong value asm("a0") = SHIFT_RIGHT(insn, (pos)-3) & 0xf8;             \
        ulong tmp;                                                                      \
        asm("1: auipc %0, %%pcrel_hi(get_f64_reg); add %0, %0, %1; jalr t0, %0, %%pcrel_lo(1b)" \
            : "=&r"(tmp), "+&r"(value)::"t0");                                          \
        value;                                                                          \
    })
#else
#define GET_F64_REG(insn, pos, regs) \
    ({                                                                                  \
        u64 value;                                                                      \
        ulong offset = SHIFT_RIGHT(insn, (pos)-3) & 0xf8;                               \
        register ulong ptr asm("a0") = (ulong)&value;                                   \
        asm ("1: auipc t1, %%pcrel_hi(get_f64_reg); add t1, t1, %2; jalr t0, t1, %%pcrel_lo(1b)" \
            : "=m"(value) : "r"(ptr), "r"(offset) : "t0", "t1");                        \
        value;                                                                          \
    })
#endif

#define SET_F64_REG(insn, pos, regs, val) \
    ({ \
        uint64_t __val = (val);                                                         \
        register ulong value asm("a0") =                                                \
            sizeof(ulong) == 4 ? (ulong)&__val : (ulong)__val;                          \
        ulong offset = SHIFT_RIGHT(insn, (pos)-3) & 0xf8;                               \
        ulong tmp;                                                                      \
        asm volatile(                                                                   \
            "1: auipc %0, %%pcrel_hi(put_f64_reg); add %0, %0, %2; jalr t0, %0, %%pcrel_lo(1b)" \
            : "=&r"(tmp)                                                                \
            : "r"(value), "r"(offset)                                                   \
            : "t0");                                                                    \
    })
#define GET_FCSR() csr_read(CSR_FCSR)
#define SET_FCSR(value) csr_write(CSR_FCSR, (value))
#define GET_FRM() csr_read(CSR_FRM)
#define SET_FRM(value) csr_write(CSR_FRM, (value))
#define GET_FFLAGS() csr_read(CSR_FFLAGS)
#define SET_FFLAGS(value) csr_write(CSR_FFLAGS, (value))

#define SET_FS_DIRTY(regs) (regs->mstatus |= MSTATUS_FS)

#define GET_F32_RS1(insn, regs) (GET_F32_REG(insn, 15, regs))
#define GET_F32_RS2(insn, regs) (GET_F32_REG(insn, 20, regs))
#define GET_F32_RS3(insn, regs) (GET_F32_REG(insn, 27, regs))
#define GET_F64_RS1(insn, regs) (GET_F64_REG(insn, 15, regs))
#define GET_F64_RS2(insn, regs) (GET_F64_REG(insn, 20, regs))
#define GET_F64_RS3(insn, regs) (GET_F64_REG(insn, 27, regs))
#define SET_F32_RD(insn, regs, val) \
    (SET_F32_REG(insn, 7, regs, val), SET_FS_DIRTY(regs))
#define SET_F64_RD(insn, regs, val) \
    (SET_F64_REG(insn, 7, regs, val), SET_FS_DIRTY(regs))

#define GET_F32_RS2C(insn, regs) (GET_F32_REG(insn, 2, regs))
#define GET_F32_RS2S(insn, regs) (GET_F32_REG(RVC_RS2S(insn), 0, regs))
#define GET_F64_RS2C(insn, regs) (GET_F64_REG(insn, 2, regs))
#define GET_F64_RS2S(insn, regs) (GET_F64_REG(RVC_RS2S(insn), 0, regs))

#endif

#endif
```

30.42 include/sbi/riscv_io.h File Reference

```
#include <sbi/riscv_barrier.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for riscv_io.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define __io_rbr() do {} while (0)
- #define __io_rar() do {} while (0)
- #define __io_rbw() do {} while (0)
- #define __io_raw() do {} while (0)
- #define readb_relaxed(c) ({ u8 __v; __io_rbr(); __v = __raw_readb(c); __io_rar(); __v; })
- #define readw_relaxed(c) ({ u16 __v; __io_rbr(); __v = __raw_readw(c); __io_rar(); __v; })
- #define readl_relaxed(c) ({ u32 __v; __io_rbr(); __v = __raw_readl(c); __io_rar(); __v; })
- #define writeb_relaxed(v, c) ({ __io_rbw(); __raw_writeb((v),(c)); __io_raw(); })
- #define writew_relaxed(v, c) ({ __io_rbw(); __raw_writew((v),(c)); __io_raw(); })
- #define writel_relaxed(v, c) ({ __io_rbw(); __raw_writel((v),(c)); __io_raw(); })
- #define readq_relaxed(c) ({ u64 __v; __io_rbr(); __v = __raw_readq(c); __io_rar(); __v; })
- #define writeq_relaxed(v, c) ({ __io_rbw(); __raw_writeq((v),(c)); __io_raw(); })
- #define __io_br() do {} while (0)
- #define __io_ar() __asm__ __volatile__ ("fence i,r" : : : "memory");
- #define __io_bw() __asm__ __volatile__ ("fence w,o" : : : "memory");
- #define __io_aw() do {} while (0)
- #define readb(c) ({ u8 __v; __io_br(); __v = __raw_readb(c); __io_ar(); __v; })
- #define readw(c) ({ u16 __v; __io_br(); __v = __raw_readw(c); __io_ar(); __v; })
- #define readl(c) ({ u32 __v; __io_br(); __v = __raw_readl(c); __io_ar(); __v; })
- #define writeb(v, c) ({ __io_bw(); __raw_writeb((v),(c)); __io_aw(); })
- #define writew(v, c) ({ __io_bw(); __raw_writew((v),(c)); __io_aw(); })
- #define writel(v, c) ({ __io_bw(); __raw_writel((v),(c)); __io_aw(); })
- #define readq(c) ({ u64 __v; __io_br(); __v = __raw_readq(c); __io_ar(); __v; })
- #define writeq(v, c) ({ __io_bw(); __raw_writeq((v),(c)); __io_aw(); })

Functions

- static void __raw_writeb (u8 val, volatile void *addr)
- static void __raw_writew (u16 val, volatile void *addr)
- static void __raw_writel (u32 val, volatile void *addr)
- static void __raw_writeq (u64 val, volatile void *addr)
- static u8 __raw_readb (const volatile void *addr)
- static u16 __raw_readw (const volatile void *addr)
- static u32 __raw_readl (const volatile void *addr)
- static u64 __raw_readq (const volatile void *addr)

30.42.1 Macro Definition Documentation

30.42.1.1 __io_ar

```
#define __io_ar( ) __asm__ __volatile__ ("fence i,r" : : : "memory");
```

30.42.1.2 __io_aw

```
#define __io_aw( ) do {} while (0)
```

30.42.1.3 __io_br

```
#define __io_br( ) do {} while (0)
```

30.42.1.4 __io_bw

```
#define __io_bw( ) __asm__ __volatile__ ("fence w,o" : : : "memory");
```

30.42.1.5 __io_rar

```
#define __io_rar( ) do {} while (0)
```

30.42.1.6 __io_raw

```
#define __io_raw( ) do {} while (0)
```

30.42.1.7 __io_rbr

```
#define __io_rbr( ) do {} while (0)
```

30.42.1.8 __io_rbw

```
#define __io_rbw( ) do {} while (0)
```

30.42.1.9 readb

```
#define readb(
            c ) ({ u8 __v; __io_br(); __v = __raw_readb(c); __io_ar(); __v; })
```

30.42.1.10 readb_relaxed

```
#define readb_relaxed(
            c ) ({ u8 __v; __io_rbr(); __v = __raw_readb(c); __io_rar(); __v; })
```

30.42.1.11 readl

```
#define readl(
            c ) ({ u32 __v; __io_br(); __v = __raw_readl(c); __io_ar(); __v; })
```

30.42.1.12 readl_relaxed

```
#define readl_relaxed(
            c ) ({ u32 __v; __io_rbr(); __v = __raw_readl(c); __io_rar(); __v; })
```

30.42.1.13 readq

```
#define readq(
            c ) ({ u64 __v; __io_br(); __v = __raw_readq(c); __io_ar(); __v; })
```

30.42.1.14 readq_relaxed

```
#define readq_relaxed(
            c ) ({ u64 __v; __io_rbr(); __v = __raw_readq(c); __io_rar(); __v; })
```

30.42.1.15 readw

```
#define readw(
            c ) ({ u16 __v; __io_br(); __v = __raw_readw(c); __io_ar(); __v; })
```

30.42.1.16 readw_relaxed

```
#define readw_relaxed(
            c ) ({ u16 __v; __io_rbr(); __v = __raw_readw(c); __io_rar(); __v; })
```

30.42.1.17 writeb

```
#define writeb(
            v,
            c ) ({ __io_bw(); __raw_writeb((v),(c)); __io_aw(); })
```

30.42.1.18 writeb_relaxed

```
#define writeb_relaxed(
            v,
            c ) ({ __io_rbw(); __raw_writeb((v),(c)); __io_raw(); })
```

30.42.1.19 writel

```
#define writel(
            v,
            c ) ({ __io_bw(); __raw_writel((v),(c)); __io_aw(); })
```

30.42.1.20 writel_relaxed

```
#define writel_relaxed(
            v,
            c ) ({ __io_rbw(); __raw_writel((v),(c)); __io_raw(); })
```

30.42.1.21 writeq

```
#define writeq(
            v,
            c ) ({ __io_bw(); __raw_writeq((v),(c)); __io_aw(); })
```

30.42.1.22 writeq_relaxed

```
#define writeq_relaxed(
            v,
            c ) ({ __io_rbw(); __raw_writeq((v),(c)); __io_raw(); })
```

30.42.1.23 writew

```
#define writew(
            v,
            c ) ({ __io_bw(); __raw_writew((v),(c)); __io_aw(); })
```

30.42.1.24 writew_relaxed

```
#define writew_relaxed(
            v,
            c ) ({ __io_rbw(); __raw_writew((v),(c)); __io_raw(); })
```

30.42.2 Function Documentation

30.42.2.1 __raw_readb()

```
static u8 __raw_readb (
            const volatile void * addr )  [inline], [static]
```

30.42.2.2 __raw_readl()

```
static u32 __raw_readl (
            const volatile void * addr )  [inline], [static]
```

30.42.2.3 __raw_readq()

```
static u64 __raw_readq (
            const volatile void * addr )  [inline], [static]
```

30.42.2.4 __raw_readw()

```
static u16 __raw_readw (
            const volatile void * addr )  [inline], [static]
```

30.42.2.5 __raw_writeb()

```
static void __raw_writeb (
            u8 val,
            volatile void * addr )  [inline], [static]
```

30.42.2.6 __raw_writel()

```
static void __raw_writel (
            u32 val,
            volatile void * addr )  [inline], [static]
```

30.42.2.7 __raw_writeq()

```
static void __raw_writeq (
            u64 val,
            volatile void * addr )  [inline], [static]
```

30.42.2.8 __raw_writew()

```
static void __raw_writew (
            u16 val,
            volatile void * addr )  [inline], [static]
```

30.43 riscv_io.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __RISCV_IO_H__
#define __RISCV_IO_H__

#include <sbi/riscv_barrier.h>
#include <sbi/sbi_types.h>

static inline void __raw_writeb(u8 val, volatile void *addr)
{
    asm volatile("sb %0, 0(%1)" : : "r"(val), "r"(addr));
}

static inline void __raw_writew(u16 val, volatile void *addr)
{
    asm volatile("sh %0, 0(%1)" : : "r"(val), "r"(addr));
}

static inline void __raw_writel(u32 val, volatile void *addr)
{
    asm volatile("sw %0, 0(%1)" : : "r"(val), "r"(addr));
}

#if __riscv_xlen != 32
static inline void __raw_writeq(u64 val, volatile void *addr)
{
    asm volatile("sd %0, 0(%1)" : : "r"(val), "r"(addr));
}
#endif

static inline u8 __raw_readb(const volatile void *addr)
{
    u8 val;

    asm volatile("lb %0, 0(%1)" : "=r"(val) : "r"(addr));
    return val;
}

static inline u16 __raw_readw(const volatile void *addr)
{
    u16 val;

    asm volatile("lh %0, 0(%1)" : "=r"(val) : "r"(addr));
    return val;
}

static inline u32 __raw_readl(const volatile void *addr)
{
    u32 val;

    asm volatile("lw %0, 0(%1)" : "=r"(val) : "r"(addr));
    return val;
}

#if __riscv_xlen != 32
static inline u64 __raw_readq(const volatile void *addr)
{
    u64 val;

    asm volatile("ld %0, 0(%1)" : "=r"(val) : "r"(addr));
    return val;
}
#endif

/* FIXME: These are now the same as asm-generic */

/* clang-format off */

#define __io_rbr()      do {} while (0)
#define __io_rar()      do {} while (0)
#define __io_rbw()      do {} while (0)
#define __io_raw()      do {} while (0)

#define readb_relaxed(c)    ({ u8  __v; __io_rbr(); __v = __raw_readb(c); __io_rar(); __v; })
#define readw_relaxed(c)    ({ u16 __v; __io_rbr(); __v = __raw_readw(c); __io_rar(); __v; })
#define readl_relaxed(c)    ({ u32 __v; __io_rbr(); __v = __raw_readl(c); __io_rar(); __v; })
```

```

#define writeb_relaxed(v,c) ({ __io_rbw(); __raw_writeb((v),(c)); __io_raw(); })
#define writew_relaxed(v,c) ({ __io_rbw(); __raw_writew((v),(c)); __io_raw(); })
#define writel_relaxed(v,c) ({ __io_rbw(); __raw_writel((v),(c)); __io_raw(); })

#if __riscv_xlen != 32
#define readq_relaxed(c)    ({ u64 __v; __io_rbr(); __v = __raw_readq(c); __io_rar(); __v; })
#define writeq_relaxed(v,c) ({ __io_rbw(); __raw_writeq((v),(c)); __io_raw(); })
#endif

#define __io_br()   do {} while (0)
#define __io_ar()   __asm__ __volatile__ ("fence i,r" : : : "memory");
#define __io_bw()   __asm__ __volatile__ ("fence w,o" : : : "memory");
#define __io_aw()   do {} while (0)

#define readb(c)    ({ u8  __v; __io_br(); __v = __raw_readb(c); __io_ar(); __v; })
#define readw(c)    ({ u16 __v; __io_br(); __v = __raw_readw(c); __io_ar(); __v; })
#define readl(c)    ({ u32 __v; __io_br(); __v = __raw_readl(c); __io_ar(); __v; })

#define writeb(v,c) ({ __io_bw(); __raw_writeb((v),(c)); __io_aw(); })
#define writew(v,c) ({ __io_bw(); __raw_writew((v),(c)); __io_aw(); })
#define writel(v,c) ({ __io_bw(); __raw_writel((v),(c)); __io_aw(); })

#if __riscv_xlen != 32
#define readq(c)    ({ u64 __v; __io_br(); __v = __raw_readq(c); __io_ar(); __v; })
#define writeq(v,c) ({ __io_bw(); __raw_writeq((v),(c)); __io_aw(); })
#endif

/* clang-format on */

#endif
```

30.44 include/sbi/riscv_locks.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for riscv_locks.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct __aligned

Macros

- #define TICKET_SHIFT 16
- #define __SPIN_LOCK_UNLOCKED (spinlock_t) { 0, 0 }
- #define SPIN_LOCK_INIT(x) x = __SPIN_LOCK_UNLOCKED
- #define SPIN_LOCK_INITIALIZER __SPIN_LOCK_UNLOCKED

- #define DEFINE_SPIN_LOCK(x) spinlock_t SPIN_LOCK_INIT(x)

Functions

- bool spin_lock_check (spinlock_t *lock)
- bool spin_trylock (spinlock_t *lock)
- void spin_lock (spinlock_t *lock)
- void spin_unlock (spinlock_t *lock)

30.44.1 Macro Definition Documentation

30.44.1.1 __SPIN_LOCK_UNLOCKED

```
#define __SPIN_LOCK_UNLOCKED  (spinlock_t) { 0, 0 }
```

30.44.1.2 DEFINE_SPIN_LOCK

```
#define DEFINE_SPIN_LOCK(
            x )  spinlock_t SPIN_LOCK_INIT(x)
```

30.44.1.3 SPIN_LOCK_INIT

```
#define SPIN_LOCK_INIT(
            x )  x = __SPIN_LOCK_UNLOCKED
```

30.44.1.4 SPIN_LOCK_INITIALIZER

```
#define SPIN_LOCK_INITIALIZER  __SPIN_LOCK_UNLOCKED
```

30.44.1.5 TICKET_SHIFT

```
#define TICKET_SHIFT 16
```

30.44.2 Function Documentation

30.44.2.1 spin_lock()

```
void spin_lock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

30.44.2.2 spin_lock_check()

```
bool spin_lock_check (
            spinlock_t * lock )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.44.2.3 spin_trylock()

```
bool spin_trylock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

30.44.2.4 spin_unlock()

```
void spin_unlock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

30.45 riscv_locks.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 * Copyright (c) 2021 Christoph Müllner <cmuellner@linux.com>
 */

```

```
#ifndef __RISCV_LOCKS_H__
#define __RISCV_LOCKS_H__

#include <sbi/sbi_types.h>

#define TICKET_SHIFT    16

typedef struct {
#if __BYTE_ORDER__ == __ORDER_BIG_ENDIAN__
        u16 next;
        u16 owner;
#else
        u16 owner;
        u16 next;
#endif
} __aligned(4) spinlock_t;

#define __SPIN_LOCK_UNLOCKED    \
    (spinlock_t) { 0, 0 }

#define SPIN_LOCK_INIT(x)   \
    x = __SPIN_LOCK_UNLOCKED

#define SPIN_LOCK_INITIALIZER   \
    __SPIN_LOCK_UNLOCKED

#define DEFINE_SPIN_LOCK(x) \
    spinlock_t SPIN_LOCK_INIT(x)

bool spin_lock_check(spinlock_t *lock);

bool spin_trylock(spinlock_t *lock);

void spin_lock(spinlock_t *lock);

void spin_unlock(spinlock_t *lock);

#endif
```

30.46 include/sbi/sbi_bitmap.h File Reference

`#include <sbi/sbi_bitops.h>`

Include dependency graph for sbi_bitmap.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define BITMAP_FIRST_WORD_MASK(start) (~0UL << ((start) % BITS_PER_LONG))
- #define BITMAP_LAST_WORD_MASK(nbits)
- #define small_const_nbits(nbits) (__builtin_constant_p(nbits) && (nbits) <= BITS_PER_LONG)
- #define DECLARE_BITMAP(name, nbits) unsigned long name[BITS_TO_LONGS(nbits)]
- #define DEFINE_BITMAP(name) extern unsigned long name[]

Functions

- static unsigned long bitmap_estimate_size (int nbits)
- void __bitmap_and (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- void __bitmap_or (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- void __bitmap_xor (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- static void bitmap_set (unsigned long *bmap, int start, int len)
- static void bitmap_clear (unsigned long *bmap, int start, int len)
- static void bitmap_zero (unsigned long *dst, int nbits)
- static int bitmap_test (unsigned long *bmap, int bit)
- static void bitmap_zero_except (unsigned long *dst, int exception, int nbits)
- static void bitmap_fill (unsigned long *dst, int nbits)
- static void bitmap_copy (unsigned long *dst, const unsigned long *src, int nbits)
- static void bitmap_and (unsigned long *dst, const unsigned long *src1, const unsigned long *src2, int nbits)
- static void bitmap_or (unsigned long *dst, const unsigned long *src1, const unsigned long *src2, int nbits)
- static void bitmap_xor (unsigned long *dst, const unsigned long *src1, const unsigned long *src2, int nbits)

30.46.1 Macro Definition Documentation

30.46.1.1 BITMAP_FIRST_WORD_MASK

```
#define BITMAP_FIRST_WORD_MASK(
            start ) (~0UL << ((start) % BITS_PER_LONG))
```

30.46.1.2 BITMAP_LAST_WORD_MASK

```
#define BITMAP_LAST_WORD_MASK(
            nbits )
```

Value:

```
(                                   \
    ((nbits) % BITS_PER_LONG) ?                 \
        ((1UL « ((nbits) % BITS_PER_LONG)) - 1) : ~0UL      \
)
```

30.46.1.3 DECLARE_BITMAP

```
#define DECLARE_BITMAP(
            name,
            nbits ) unsigned long name[BITS_TO_LONGS(nbits)]
```

30.46.1.4 DEFINE_BITMAP

```
#define DEFINE_BITMAP(
            name ) extern unsigned long name[]
```

30.46.1.5 small_const_nbits

```
#define small_const_nbits(
            nbits ) (__builtin_constant_p(nbits) && (nbits) <= BITS_PER_LONG)
```

30.46.2 Function Documentation

30.46.2.1 __bitmap_and()

```
void __bitmap_and (
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

Here is the caller graph for this function:

30.46.2.2 __bitmap_or()

```
void __bitmap_or (
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

Here is the caller graph for this function:

30.46.2.3 __bitmap_xor()

```
void __bitmap_xor (
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

Here is the caller graph for this function:

30.46.2.4 bitmap_and()

```
static void bitmap_and (
            unsigned long * dst,
            const unsigned long * src1,
            const unsigned long * src2,
            int nbits )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.46.2.5 bitmap_clear()

```
static void bitmap_clear (
            unsigned long * bmap,
            int start,
            int len )  [inline], [static]
```

Here is the caller graph for this function:

30.46.2.6 bitmap_copy()

```
static void bitmap_copy (
            unsigned long * dst,
            const unsigned long * src,
            int nbits )  [inline], [static]
```

Here is the caller graph for this function:

30.46.2.7 bitmap_estimate_size()

```
static unsigned long bitmap_estimate_size (
            int nbits )  [inline], [static]
```

30.46.2.8 bitmap_fill()

```
static void bitmap_fill (
            unsigned long * dst,
            int nbits )  [inline], [static]
```

Here is the caller graph for this function:

30.46.2.9 bitmap_or()

```
static void bitmap_or (
            unsigned long * dst,
            const unsigned long * src1,
            const unsigned long * src2,
            int nbits )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.46.2.10 bitmap_set()

```
static void bitmap_set (
            unsigned long * bmap,
            int start,
            int len )  [inline], [static]
```

Here is the caller graph for this function:

30.46.2.11 bitmap_test()

```
static int bitmap_test (
            unsigned long * bmap,
            int bit )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.46.2.12 bitmap_xor()

```
static void bitmap_xor (
            unsigned long * dst,
            const unsigned long * src1,
            const unsigned long * src2,
            int nbits )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.46.2.13 bitmap_zero()

```
static void bitmap_zero (
            unsigned long * dst,
            int nbits )  [inline], [static]
```

Here is the caller graph for this function:

30.46.2.14 bitmap_zero_except()

```
static void bitmap_zero_except (
            unsigned long * dst,
            int exception,
            int nbits )  [inline], [static]
```

Here is the call graph for this function:

30.47 sbi_bitmap.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_BITMAP_H__
#define __SBI_BITMAP_H__

#include <sbi/sbi_bitops.h>

#define BITMAP_FIRST_WORD_MASK(start) (~0UL << ((start) % BITS_PER_LONG))
#define BITMAP_LAST_WORD_MASK(nbits)                    \
(                                   \
    ((nbits) % BITS_PER_LONG) ?                 \
        ((1UL << ((nbits) % BITS_PER_LONG)) - 1) : ~0UL     \
)

#define small_const_nbits(nbits) \
    (__builtin_constant_p(nbits) && (nbits) <= BITS_PER_LONG)

#define DECLARE_BITMAP(name, nbits) unsigned long name[BITS_TO_LONGS(nbits)]
#define DEFINE_BITMAP(name)     extern unsigned long name[]

static inline unsigned long bitmap_estimate_size(int nbits)
{
    return (BITS_TO_LONGS(nbits) * sizeof(unsigned long));
}

void __bitmap_and(unsigned long *dst, const unsigned long *bitmap1,
          const unsigned long *bitmap2, int bits);
void __bitmap_or(unsigned long *dst, const unsigned long *bitmap1,
         const unsigned long *bitmap2, int bits);
void __bitmap_xor(unsigned long *dst, const unsigned long *bitmap1,
          const unsigned long *bitmap2, int bits);

static inline void bitmap_set(unsigned long *bmap, int start, int len)
{
    int bit;
    for (bit = start; bit < (start + len); bit++)
        bmap[BIT_WORD(bit)] |= (0x1UL << BIT_WORD_OFFSET(bit));
}

static inline void bitmap_clear(unsigned long *bmap, int start, int len)
{
    int bit;
    for (bit = start; bit < (start + len); bit++)
        bmap[BIT_WORD(bit)] &= ~(0x1UL << BIT_WORD_OFFSET(bit));
}

static inline void bitmap_zero(unsigned long *dst, int nbits)
{
    if (small_const_nbits(nbits))
        *dst = 0UL;
    else {
        size_t i, len = BITS_TO_LONGS(nbits);
        for (i = 0; i < len; i++)
            dst[i] = 0;
    }
}

static inline int bitmap_test(unsigned long *bmap, int bit)
{
    return __test_bit(bit, bmap);
}

static inline void bitmap_zero_except(unsigned long *dst,
                      int exception, int nbits)
{
    if (small_const_nbits(nbits))
        *dst = 0UL;
    else {
        size_t i, len = BITS_TO_LONGS(nbits);
        for (i = 0; i < len; i++)
            dst[i] = 0;
    }
    if (exception < nbits)
        __set_bit(exception, dst);
}

```

```
static inline void bitmap_fill(unsigned long *dst, int nbits)
{
    size_t i, nlongs = BITS_TO_LONGS(nbits);
    if (!small_const_nbits(nbits)) {
        for (i = 0; i < (nlongs - 1); i++)
            dst[i] = -1UL;
    }
    dst[nlongs - 1] = BITMAP_LAST_WORD_MASK(nbits);
}

static inline void bitmap_copy(unsigned long *dst,
                    const unsigned long *src, int nbits)
{
    if (small_const_nbits(nbits))
        *dst = *src;
    else {
        size_t i, len = BITS_TO_LONGS(nbits);
        for (i = 0; i < len; i++)
            dst[i] = src[i];
    }
}

static inline void bitmap_and(unsigned long *dst, const unsigned long *src1,
                  const unsigned long *src2, int nbits)
{
    if (small_const_nbits(nbits))
        *dst = *src1 & *src2;
    else
        __bitmap_and(dst, src1, src2, nbits);
}

static inline void bitmap_or(unsigned long *dst, const unsigned long *src1,
                 const unsigned long *src2, int nbits)
{
    if (small_const_nbits(nbits))
        *dst = *src1 | *src2;
    else
        __bitmap_or(dst, src1, src2, nbits);
}

static inline void bitmap_xor(unsigned long *dst, const unsigned long *src1,
                  const unsigned long *src2, int nbits)
{
    if (small_const_nbits(nbits))
        *dst = *src1 ^ *src2;
    else
        __bitmap_xor(dst, src1, src2, nbits);
}

#endif
```

30.48 include/sbi/sbi_bitops.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_bitops.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define BITS_PER_LONG (8 ∗ __SIZEOF_LONG__)
- #define BITS_PER_LONG_LONG 64
- #define EXTRACT_FIELD(val, which) (((val) & (which)) / ((which) & ∼((which)-1)))
- #define INSERT_FIELD(val, which, fieldval) (((val) & ∼(which)) | ((fieldval) ∗ ((which) & ∼((which)-1))))
- #define BITS_TO_LONGS(nbits)
- #define BIT(nr) (1UL << (nr))
- #define BIT_MASK(nr) (1UL << ((nr) % BITS_PER_LONG))
- #define BIT_WORD(bit) ((bit) / BITS_PER_LONG)
- #define BIT_WORD_OFFSET(bit) ((bit) & (BITS_PER_LONG - 1))
- #define BIT_ALIGN(bit, align) (((bit) + ((align) - 1)) & ∼((align) - 1))
- #define BIT_ULL(nr) (1ULL << (nr))
- #define GENMASK(h, l) (((∼0UL) - (1UL << (l)) + 1) & (∼0UL >> (BITS_PER_LONG - 1 - (h))))
- #define GENMASK_ULL(h, l) (((∼0ULL) - (1ULL << (l)) + 1) & (∼0ULL >> (BITS_PER_LONG_LONG - 1 - (h))))
- #define sbi_ffz(x) sbi_ffs(∼(x))
- #define for_each_set_bit(bit, addr, size)
- #define for_each_set_bit_from(bit, addr, size)
- #define for_each_clear_bit(bit, addr, size)
- #define for_each_clear_bit_from(bit, addr, size)

Functions

- static int sbi_ffs (unsigned long word)
- static unsigned long sbi_fls (unsigned long word)
- static unsigned long sbi_popcount (unsigned long word)
- unsigned long find_first_bit (const unsigned long ∗addr, unsigned long size)
- unsigned long find_first_zero_bit (const unsigned long ∗addr, unsigned long size)
- unsigned long find_last_bit (const unsigned long ∗addr, unsigned long size)
- unsigned long find_next_bit (const unsigned long ∗addr, unsigned long size, unsigned long offset)
- unsigned long find_next_zero_bit (const unsigned long ∗addr, unsigned long size, unsigned long offset)
- static void __set_bit (int nr, volatile unsigned long ∗addr)
- static void __clear_bit (int nr, volatile unsigned long ∗addr)
- static void __change_bit (int nr, volatile unsigned long ∗addr)
- static int __test_and_set_bit (int nr, volatile unsigned long ∗addr)
- static int __test_and_clear_bit (int nr, volatile unsigned long ∗addr)
- static int __test_bit (int nr, const volatile unsigned long ∗addr)

30.48.1 Macro Definition Documentation

30.48.1.1 BIT

```
#define BIT(
            nr ) (1UL << (nr))
```

30.48.1.2 BIT_ALIGN

```
#define BIT_ALIGN(
            bit,
            align ) (((bit) + ((align) - 1)) & ∼((align) - 1))
```

30.48.1.3 BIT_MASK

```
#define BIT_MASK(
            nr ) (1UL << ((nr) % BITS_PER_LONG))
```

30.48.1.4 BIT_ULL

```
#define BIT_ULL(
            nr ) (1ULL << (nr))
```

30.48.1.5 BIT_WORD

```
#define BIT_WORD(
            bit ) ((bit) / BITS_PER_LONG)
```

30.48.1.6 BIT_WORD_OFFSET

```
#define BIT_WORD_OFFSET(
            bit ) ((bit) & (BITS_PER_LONG - 1))
```

30.48.1.7 BITS_PER_LONG

```
#define BITS_PER_LONG (8 * __SIZEOF_LONG__)
```

30.48.1.8 BITS_PER_LONG_LONG

```
#define BITS_PER_LONG_LONG 64
```

30.48.1.9 BITS_TO_LONGS

```
#define BITS_TO_LONGS(
            nbits )
```

Value:

```
 (((nbits) + BITS_PER_LONG - 1) / \
 BITS_PER_LONG)
```

30.48.1.10 EXTRACT_FIELD

```
#define EXTRACT_FIELD(
            val,
            which )  (((val) & (which)) / ((which) & ~((which)-1)))
```

30.48.1.11 for_each_clear_bit

```
#define for_each_clear_bit(
            bit,
            addr,
            size )
```

Value:

```
for ((bit) = find_first_zero_bit((addr), (size));    \
     (bit) < (size);                                 \
     (bit) = find_next_zero_bit((addr), (size), (bit) + 1))
```

30.48.1.12 for_each_clear_bit_from

```
#define for_each_clear_bit_from(
            bit,
            addr,
            size )
```

Value:

```
for ((bit) = find_next_zero_bit((addr), (size), (bit));   \
     (bit) < (size);                                      \
     (bit) = find_next_zero_bit((addr), (size), (bit) + 1))
```

30.48.1.13 for_each_set_bit

```
#define for_each_set_bit(
            bit,
            addr,
            size )
```

Value:

```
for ((bit) = find_first_bit((addr), (size));      \
     (bit) < (size);                              \
     (bit) = find_next_bit((addr), (size), (bit) + 1))
```

30.48.1.14 for_each_set_bit_from

```
#define for_each_set_bit_from(
            bit,
            addr,
            size )
```

Value:

```
for ((bit) = find_next_bit((addr), (size), (bit)); \
     (bit) < (size);                               \
     (bit) = find_next_bit((addr), (size), (bit) + 1))
```

30.48.1.15 GENMASK

```
#define GENMASK(
            h,
            l )   (((~0UL) - (1UL << (l)) + 1) & (~0UL >> (BITS_PER_LONG - 1 - (h))))
```

30.48.1.16 GENMASK_ULL

```
#define GENMASK_ULL(
            h,
            l )   (((~0ULL) - (1ULL << (l)) + 1) & (~0ULL >> (BITS_PER_LONG_LONG - 1 -
(h))))
```

30.48.1.17 INSERT_FIELD

```
#define INSERT_FIELD(
            val,
            which,
            fieldval )   (((val) & ~(which)) | ((fieldval) * ((which) & ~((which)-1))))
```

30.48.1.18 sbi_ffz

```
#define sbi_ffz(
            x ) sbi_ffs(~(x))
```

30.48.2 Function Documentation

30.48.2.1 __change_bit()

```
static void __change_bit (
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

__change_bit - Toggle a bit in memory @nr: the bit to change @addr: the address to start counting from This function is non-atomic and may be reordered.

30.48.2.2 __clear_bit()

```
static void __clear_bit (
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

__clear_bit - Clear a bit in memory @nr: the bit to clear @addr: the address to start counting from
This function is non-atomic and may be reordered. Here is the caller graph for this function:

30.48.2.3 __set_bit()

```
static void __set_bit (
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

__set_bit - Set a bit in memory @nr: the bit to set @addr: the address to start counting from
This function is non-atomic and may be reordered. Here is the caller graph for this function:

30.48.2.4 __test_and_clear_bit()

```
static int __test_and_clear_bit (
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

__test_and_clear_bit - Clear a bit and return its old value @nr: Bit to clear @addr: Address to count from
This operation is non-atomic and can be reordered.

30.48.2.5 __test_and_set_bit()

```
static int __test_and_set_bit (
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

__test_and_set_bit - Set a bit and return its old value @nr: Bit to set @addr: Address to count from
This operation is non-atomic and can be reordered.

30.48.2.6 __test_bit()

```
static int __test_bit (
            int nr,
            const volatile unsigned long * addr )  [inline], [static]
```

__test_bit - Determine whether a bit is set @nr: bit number to test @addr: Address to start counting from This operation is non-atomic and can be reordered. Here is the caller graph for this function:

30.48.2.7 find_first_bit()

```
unsigned long find_first_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_first_bit - find the first set bit in a memory region @addr: The address to start the search at @size: The maximum size to search

Returns the bit number of the first set bit. Here is the call graph for this function:

30.48.2.8 find_first_zero_bit()

```
unsigned long find_first_zero_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_first_zero_bit - find the first cleared bit in a memory region @addr: The address to start the search at @size: The maximum size to search
Returns the bit number of the first cleared bit.

30.48.2.9 find_last_bit()

```
unsigned long find_last_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_last_bit - find the last set bit in a memory region @addr: The address to start the search at @size: The maximum size to search
Returns the bit number of the first set bit, or size. Here is the call graph for this function:

30.48.2.10 find_next_bit()

```
unsigned long find_next_bit (
            const unsigned long * addr,
            unsigned long size,
            unsigned long offset )
```

find_next_bit - find the next set bit in a memory region @addr: The address to base the search on @offset: The

bitnumber to start searching at @size: The bitmap size in bits Here is the call graph for this function:

30.48.2.11 find_next_zero_bit()

```
unsigned long find_next_zero_bit (
            const unsigned long * addr,
            unsigned long size,
            unsigned long offset )
```

find_next_zero_bit - find the next cleared bit in a memory region @addr: The address to base the search on @offset: The bitnumber to start searching at @size: The bitmap size in bits

30.48.2.12 sbi_ffs()

```
static int sbi_ffs (
            unsigned long word )  [inline], [static]
```

sbi_ffs - find first (less-significant) set bit in a long word. @word: The word to search
Undefined if no bit exists, so code should check against 0 first. Here is the caller graph for this function:

30.48.2.13 sbi_fls()

```
static unsigned long sbi_fls (
            unsigned long word )  [inline], [static]
```

sbi_fls - find last (most-significant) set bit in a long word @word: the word to search
Undefined if no set bit exists, so code should check against 0 first. Here is the caller graph for this function:

30.48.2.14 sbi_popcount()

```
static unsigned long sbi_popcount (
            unsigned long word ) [inline], [static]
```

sbi_popcount - find the number of set bit in a long word @word: the word to search Here is the caller graph for this function:

30.49 sbi_bitops.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Atish Patra <atish.patra@wdc.com>
 */

#ifndef __SBI_BITOPS_H__
#define __SBI_BITOPS_H__

#include <sbi/sbi_types.h>

#define BITS_PER_LONG       (8 * __SIZEOF_LONG__)

#define BITS_PER_LONG_LONG  64

#define EXTRACT_FIELD(val, which) \
    (((val) & (which)) / ((which) & ~((which)-1)))
#define INSERT_FIELD(val, which, fieldval) \
    (((val) & ~(which)) | ((fieldval) * ((which) & ~((which)-1))))

#define BITS_TO_LONGS(nbits)    (((nbits) + BITS_PER_LONG - 1) / \
                 BITS_PER_LONG)

#define BIT(nr)         (1UL << (nr))
#define BIT_MASK(nr)        (1UL << ((nr) % BITS_PER_LONG))
#define BIT_WORD(bit)       ((bit) / BITS_PER_LONG)
#define BIT_WORD_OFFSET(bit)    ((bit) & (BITS_PER_LONG - 1))
#define BIT_ALIGN(bit, align)   (((bit) + ((align) - 1)) & ~((align) - 1))

#define BIT_ULL(nr)     (1ULL << (nr))

#define GENMASK(h, l) \
```

```
    (((~0UL) - (1UL << (l)) + 1) & (~0UL >> (BITS_PER_LONG - 1 - (h))))

#define GENMASK_ULL(h, l) \
    (((~0ULL) - (1ULL << (l)) + 1) & (~0ULL >> (BITS_PER_LONG_LONG - 1 - (h))))
static inline int sbi_ffs(unsigned long word)
{
    int num = 0;

#if BITS_PER_LONG == 64
    if ((word & 0xffffffff) == 0) {
        num += 32;
        word >>= 32;
    }
#endif
    if ((word & 0xffff) == 0) {
        num += 16;
        word >>= 16;
    }
    if ((word & 0xff) == 0) {
        num += 8;
        word >>= 8;
    }
    if ((word & 0xf) == 0) {
        num += 4;
        word >>= 4;
    }
    if ((word & 0x3) == 0) {
        num += 2;
        word >>= 2;
    }
    if ((word & 0x1) == 0)
        num += 1;
    return num;
}

/*
 * sbi_ffz - find first zero in word.
 * @word: The word to search
 *
 * Undefined if no zero exists, so code should check against ~0UL first.
 */
#define sbi_ffz(x) sbi_ffs(~(x))

static inline unsigned long sbi_fls(unsigned long word)
{
    int num = BITS_PER_LONG - 1;

#if BITS_PER_LONG == 64
    if (!(word & (~0ul << 32))) {
        num -= 32;
        word <<= 32;
    }
#endif
    if (!(word & (~0ul << (BITS_PER_LONG-16)))) {
        num -= 16;
        word <<= 16;
    }
    if (!(word & (~0ul << (BITS_PER_LONG-8)))) {
        num -= 8;
        word <<= 8;
    }
    if (!(word & (~0ul << (BITS_PER_LONG-4)))) {
        num -= 4;
        word <<= 4;
    }
    if (!(word & (~0ul << (BITS_PER_LONG-2)))) {
        num -= 2;
        word <<= 2;
    }
    if (!(word & (~0ul << (BITS_PER_LONG-1))))
        num -= 1;
    return num;
}

static inline unsigned long sbi_popcount(unsigned long word)
{
    unsigned long count = 0;

    while (word) {
        word &= word - 1;
        count++;
    }

    return count;
}

#define for_each_set_bit(bit, addr, size) \
```

```
    for ((bit) = find_first_bit((addr), (size));        \
         (bit) < (size);                                \
         (bit) = find_next_bit((addr), (size), (bit) + 1))

/* same as for_each_set_bit() but use bit as value to start with */
#define for_each_set_bit_from(bit, addr, size) \
    for ((bit) = find_next_bit((addr), (size), (bit));  \
         (bit) < (size);                                \
         (bit) = find_next_bit((addr), (size), (bit) + 1))

#define for_each_clear_bit(bit, addr, size) \
    for ((bit) = find_first_zero_bit((addr), (size));   \
         (bit) < (size);                                \
         (bit) = find_next_zero_bit((addr), (size), (bit) + 1))

/* same as for_each_clear_bit() but use bit as value to start with */
#define for_each_clear_bit_from(bit, addr, size) \
    for ((bit) = find_next_zero_bit((addr), (size), (bit)); \
         (bit) < (size);                                \
         (bit) = find_next_zero_bit((addr), (size), (bit) + 1))

unsigned long find_first_bit(const unsigned long *addr,
                 unsigned long size);

unsigned long find_first_zero_bit(const unsigned long *addr,
                  unsigned long size);

unsigned long find_last_bit(const unsigned long *addr,
                unsigned long size);

unsigned long find_next_bit(const unsigned long *addr,
                unsigned long size, unsigned long offset);

unsigned long find_next_zero_bit(const unsigned long *addr,
                 unsigned long size,
                 unsigned long offset);

static inline void __set_bit(int nr, volatile unsigned long *addr)
{
    unsigned long mask = BIT_MASK(nr);
    unsigned long *p = ((unsigned long *)addr) + BIT_WORD(nr);

    *p |= mask;
}

static inline void __clear_bit(int nr, volatile unsigned long *addr)
{
    unsigned long mask = BIT_MASK(nr);
    unsigned long *p = ((unsigned long *)addr) + BIT_WORD(nr);

    *p &= ~mask;
}

static inline void __change_bit(int nr, volatile unsigned long *addr)
{
    unsigned long mask = BIT_MASK(nr);
    unsigned long *p = ((unsigned long *)addr) + BIT_WORD(nr);

    *p ^= mask;
}

static inline int __test_and_set_bit(int nr, volatile unsigned long *addr)
{
    unsigned long mask = BIT_MASK(nr);
    unsigned long *p = ((unsigned long *)addr) + BIT_WORD(nr);
    unsigned long old = *p;

    *p = old | mask;
    return (old & mask) != 0;
}

static inline int __test_and_clear_bit(int nr, volatile unsigned long *addr)
{
    unsigned long mask = BIT_MASK(nr);
    unsigned long *p = ((unsigned long *)addr) + BIT_WORD(nr);
    unsigned long old = *p;

    *p = old & ~mask;
    return (old & mask) != 0;
}

static inline int __test_bit(int nr, const volatile unsigned long *addr)
{
    return 1UL & (addr[BIT_WORD(nr)] >> (nr & (BITS_PER_LONG-1)));
}

#endif
```

30.50 include/sbi/sbi_byteorder.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_byteorder.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define _conv_cast(type, val) ((type)(val))
- #define BSWAP16(x)
- #define BSWAP32(x)
- #define BSWAP64(x)
- #define cpu_to_be16(x) _conv_cast(uint16_t, BSWAP16(x))
- #define cpu_to_be32(x) _conv_cast(uint32_t, BSWAP32(x))
- #define cpu_to_be64(x) _conv_cast(uint64_t, BSWAP64(x))
- #define be16_to_cpu(x) _conv_cast(uint16_t, BSWAP16(x))
- #define be32_to_cpu(x) _conv_cast(uint32_t, BSWAP32(x))
- #define be64_to_cpu(x) _conv_cast(uint64_t, BSWAP64(x))
- #define cpu_to_le16(x) _conv_cast(uint16_t, (x))
- #define cpu_to_le32(x) _conv_cast(uint32_t, (x))
- #define cpu_to_le64(x) _conv_cast(uint64_t, (x))
- #define le16_to_cpu(x) _conv_cast(uint16_t, (x))
- #define le32_to_cpu(x) _conv_cast(uint32_t, (x))
- #define le64_to_cpu(x) _conv_cast(uint64_t, (x))

30.50.1 Macro Definition Documentation

30.50.1.1 _conv_cast

```
#define _conv_cast(
            type,
            val ) ((type)(val))
```

30.50.1.2 be16_to_cpu

```
#define be16_to_cpu(
            x ) _conv_cast(uint16_t, BSWAP16(x))
```

30.50.1.3 be32_to_cpu

```
#define be32_to_cpu(
            x ) _conv_cast(uint32_t, BSWAP32(x))
```

30.50.1.4 be64_to_cpu

```
#define be64_to_cpu(
            x ) _conv_cast(uint64_t, BSWAP64(x))
```

30.50.1.5 BSWAP16

```
#define BSWAP16(
            x )
```

Value:

```
(((x) & 0x00ff) « 8) | \
(((x) & 0xff00) » 8))
```

30.50.1.6 BSWAP32

```
#define BSWAP32(
            x )
```

Value:

```
(((x) & 0x000000ff) « 24) | \
(((x) & 0x0000ff00) « 8) | \
(((x) & 0x00ff0000) » 8) | \
(((x) & 0xff000000) » 24))
```

30.50.1.7 BSWAP64

```
#define BSWAP64(
            x )
```

Value:

```
(((x) & 0x00000000000000ffULL) « 56) | \
(((x) & 0x000000000000ff00ULL) « 40) | \
(((x) & 0x0000000000ff0000ULL) « 24) | \
(((x) & 0x00000000ff000000ULL) « 8) | \
(((x) & 0x000000ff00000000ULL) » 8) | \
(((x) & 0x0000ff0000000000ULL) » 24) | \
(((x) & 0x00ff000000000000ULL) » 40) | \
(((x) & 0xff00000000000000ULL) » 56))
```

30.50.1.8 cpu_to_be16

```
#define cpu_to_be16(
            x ) _conv_cast(uint16_t, BSWAP16(x))
```

30.50.1.9 cpu_to_be32

```
#define cpu_to_be32(
            x ) _conv_cast(uint32_t, BSWAP32(x))
```

30.50.1.10 cpu_to_be64

```
#define cpu_to_be64(
            x ) _conv_cast(uint64_t, BSWAP64(x))
```

30.50.1.11 cpu_to_le16

```
#define cpu_to_le16(
            x ) _conv_cast(uint16_t, (x))
```

30.50.1.12 cpu_to_le32

```
#define cpu_to_le32(
            x ) _conv_cast(uint32_t, (x))
```

30.50.1.13 cpu_to_le64

```
#define cpu_to_le64(
            x ) _conv_cast(uint64_t, (x))
```

30.50.1.14 le16_to_cpu

```
#define le16_to_cpu(
            x ) _conv_cast(uint16_t, (x))
```

30.50.1.15 le32_to_cpu

```
#define le32_to_cpu(
            x ) _conv_cast(uint32_t, (x))
```

30.50.1.16 le64_to_cpu

```
#define le64_to_cpu(
            x ) _conv_cast(uint64_t, (x))
```

30.51 sbi_byteorder.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Ventana Micro Systems Inc.
 */

#ifndef __SBI_BYTEORDER_H__
#define __SBI_BYTEORDER_H__

#ifdef __ASSEMBLER__
# define _conv_cast(type, val) (val)
#else
# include <sbi/sbi_types.h>
# define _conv_cast(type, val) ((type)(val))
#endif

#define BSWAP16(x)  ((((x) & 0x00ff) << 8) | \
             (((x) & 0xff00) >> 8))
#define BSWAP32(x)  ((((x) & 0x000000ff) << 24) | \
             (((x) & 0x0000ff00) << 8) | \
             (((x) & 0x00ff0000) >> 8) | \
             (((x) & 0xff000000) >> 24))
#define BSWAP64(x)  ((((x) & 0x00000000000000ffULL) << 56) | \
             (((x) & 0x000000000000ff00ULL) << 40) | \
             (((x) & 0x0000000000ff0000ULL) << 24) | \
             (((x) & 0x00000000ff000000ULL) << 8) | \
             (((x) & 0x000000ff00000000ULL) >> 8) | \
             (((x) & 0x0000ff0000000000ULL) >> 24) | \
             (((x) & 0x00ff000000000000ULL) >> 40) | \
             (((x) & 0xff00000000000000ULL) >> 56))

#if __BYTE_ORDER__ == __ORDER_LITTLE_ENDIAN__  /* CPU(little-endian) */
#define cpu_to_be16(x)      _conv_cast(uint16_t, BSWAP16(x))
#define cpu_to_be32(x)      _conv_cast(uint32_t, BSWAP32(x))
#define cpu_to_be64(x)      _conv_cast(uint64_t, BSWAP64(x))

#define be16_to_cpu(x)      _conv_cast(uint16_t, BSWAP16(x))
#define be32_to_cpu(x)      _conv_cast(uint32_t, BSWAP32(x))
#define be64_to_cpu(x)      _conv_cast(uint64_t, BSWAP64(x))

#define cpu_to_le16(x)      _conv_cast(uint16_t, (x))
#define cpu_to_le32(x)      _conv_cast(uint32_t, (x))
#define cpu_to_le64(x)      _conv_cast(uint64_t, (x))

#define le16_to_cpu(x)      _conv_cast(uint16_t, (x))
#define le32_to_cpu(x)      _conv_cast(uint32_t, (x))
```

```
#define le64_to_cpu(x)      _conv_cast(uint64_t, (x))
#else /* CPU(big-endian) */
#define cpu_to_be16(x)      _conv_cast(uint16_t, (x))
#define cpu_to_be32(x)      _conv_cast(uint32_t, (x))
#define cpu_to_be64(x)      _conv_cast(uint64_t, (x))

#define be16_to_cpu(x)      _conv_cast(uint16_t, (x))
#define be32_to_cpu(x)      _conv_cast(uint32_t, (x))
#define be64_to_cpu(x)      _conv_cast(uint64_t, (x))

#define cpu_to_le16(x)      _conv_cast(uint16_t, BSWAP16(x))
#define cpu_to_le32(x)      _conv_cast(uint32_t, BSWAP32(x))
#define cpu_to_le64(x)      _conv_cast(uint64_t, BSWAP64(x))

#define le16_to_cpu(x)      _conv_cast(uint16_t, BSWAP16(x))
#define le32_to_cpu(x)      _conv_cast(uint32_t, BSWAP32(x))
#define le64_to_cpu(x)      _conv_cast(uint64_t, BSWAP64(x))
#endif

#if __riscv_xlen == 64
#define cpu_to_lle      cpu_to_le64
#define lle_to_cpu      le64_to_cpu
#elif __riscv_xlen == 32
#define cpu_to_lle      cpu_to_le32
#define lle_to_cpu      le32_to_cpu
#else
#error "Unknown __riscv_xlen"
#endif

#endif /* __SBI_BYTEORDER_H__ */
```

30.52 include/sbi/sbi_console.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_console.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_console_device

Macros

- #define __printf(a, b) __attribute__((format(printf, a, b)))
- #define SBI_ASSERT(cond, args)

Functions

- bool sbi_isprintable (char ch)

- int sbi_getc (void)
- void sbi_putc (char ch)
- void sbi_puts (const char ∗str)
- unsigned long sbi_nputs (const char ∗str, unsigned long len)
- void sbi_gets (char ∗s, int maxwidth, char endchar)
- unsigned long sbi_ngets (char ∗str, unsigned long len)
- int __printf (2, 3) sbi_sprintf(char ∗out
- int const char int __printf (3, 4) sbi_snprintf(char ∗out
- int const char int u32 const char int __printf (1, 2) sbi_printf(const char ∗format
- int const char int u32 const char int int void const struct sbi_console_device ∗ sbi_console_get_device (void)
- void sbi_console_set_device (const struct sbi_console_device ∗dev)

Variables

- int const char ∗ format
- int const char int u32 out_sz

30.52.1 Macro Definition Documentation

30.52.1.1 __printf

```
#define __printf(
            a,
            b ) __attribute__((format(printf, a, b)))
```

30.52.1.2 SBI_ASSERT

```
#define SBI_ASSERT(
            cond,
            args )
```

Value:

```
    do { \
    if (unlikely(!(cond))) \
        sbi_panic args; \
} while (0)
```

30.52.2 Function Documentation

30.52.2.1 __printf() [1/3]

```
int const char int u32 const char int int void __printf (
            1 ,
            2  ) const
```

30.52.2.2 __printf() [2/3]

```
int __printf (
            2 ,
            3  )
```

30.52.2.3 __printf() [3/3]

```
int const char int __printf (
            3 ,
            4  )
```

30.52.2.4 sbi_console_get_device()

```
int const char int u32 const char int int void const struct sbi_console_device * sbi_console_↩
get_device (
            void  )
```

Here is the caller graph for this function:

30.52.2.5 sbi_console_set_device()

```
void sbi_console_set_device (
            const struct sbi_console_device * dev )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.52.2.6 sbi_getc()

```
int sbi_getc (
            void  )
```

Here is the caller graph for this function:

30.52.2.7 sbi_gets()

```
void sbi_gets (
            char * s,
            int maxwidth,
            char endchar )
```

Here is the call graph for this function:

30.52.2.8 sbi_isprintable()

```
bool sbi_isprintable (
            char ch )
```

30.52.2.9 sbi_ngets()

```
unsigned long sbi_ngets (
            char * str,
            unsigned long len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.52.2.10 sbi_nputs()

```
unsigned long sbi_nputs (
            const char * str,
            unsigned long len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.52.2.11 sbi_putc()

```
void sbi_putc (
            char ch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.52.2.12 sbi_puts()

```
void sbi_puts (
            const char * str )
```

Here is the call graph for this function:

30.52.3 Variable Documentation

30.52.3.1 format

```
int const char int u32 const char * format
```

30.52.3.2 out_sz

```
int const char int u32 out_sz
```

30.53 sbi_console.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_CONSOLE_H__
#define __SBI_CONSOLE_H__

#include <sbi/sbi_types.h>

struct sbi_console_device {
    char name[32];

    void (*console_putc)(char ch);

    unsigned long (*console_puts)(const char *str, unsigned long len);

    int (*console_getc)(void);
};

#define __printf(a, b) __attribute__((format(printf, a, b)))

bool sbi_isprintable(char ch);

int sbi_getc(void);

void sbi_putc(char ch);

void sbi_puts(const char *str);

unsigned long sbi_nputs(const char *str, unsigned long len);

void sbi_gets(char *s, int maxwidth, char endchar);

unsigned long sbi_ngets(char *str, unsigned long len);

int __printf(2, 3) sbi_sprintf(char *out, const char *format, ...);

int __printf(3, 4) sbi_snprintf(char *out, u32 out_sz, const char *format, ...);

int __printf(1, 2) sbi_printf(const char *format, ...);

int __printf(1, 2) sbi_dprintf(const char *format, ...);

void __printf(1, 2) __attribute__((noreturn)) sbi_panic(const char *format, ...);

const struct sbi_console_device *sbi_console_get_device(void);

void sbi_console_set_device(const struct sbi_console_device *dev);

struct sbi_scratch;

#define SBI_ASSERT(cond, args) do { \
    if (unlikely(!(cond))) \
        sbi_panic args; \
} while (0)

#endif
```

30.54 include/sbi/sbi_const.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define __AC(X, Y) (X##Y)
- #define _AC(X, Y) __AC(X,Y)
- #define _AT(T, X) ((T)(X))
- #define _UL(x) (_AC(x, UL))
- #define _ULL(x) (_AC(x, ULL))
- #define _BITUL(x) (_UL(1) << (x))
- #define _BITULL(x) (_ULL(1) << (x))
- #define UL(x) (_UL(x))
- #define ULL(x) (_ULL(x))
- #define __STR(s) #s
- #define STRINGIFY(s) __STR(s)

30.54.1 Macro Definition Documentation

30.54.1.1 __AC

```
#define __AC(
            X,
            Y ) (X##Y)
```

30.54.1.2 __STR

```
#define __STR(
            s ) #s
```

30.54.1.3 _AC

```
#define _AC(
            X,
            Y ) __AC(X,Y)
```

30.54.1.4 _AT

```
#define _AT(
            T,
            X ) ((T)(X))
```

30.54.1.5 _BITUL

```
#define _BITUL(
            x ) (_UL(1) << (x))
```

30.54.1.6 _BITULL

```
#define _BITULL(
            x ) (_ULL(1) << (x))
```

30.54.1.7 _UL

```
#define _UL(
            x ) (_AC(x, UL))
```

30.54.1.8 _ULL

```
#define _ULL(
            x ) (_AC(x, ULL))
```

30.54.1.9 STRINGIFY

```
#define STRINGIFY(
            s ) __STR(s)
```

30.54.1.10 UL

```
#define UL(
            x ) (_UL(x))
```

30.54.1.11 ULL

```
#define ULL(
            x ) (_ULL(x))
```

30.55 sbi_const.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_CONST_H__
#define __SBI_CONST_H__

/*
 * Some constant macros are used in both assembler and
 * C code.  Therefore we cannot annotate them always with
 * 'UL' and other type specifiers unilaterally.  We
 * use the following macros to deal with this.
 *
 * Similarly, _AT() will cast an expression with a type in C, but
 * leave it unchanged in asm.
 */

/* clang-format off */

#ifdef __ASSEMBLER__
#define _AC(X,Y)    X
#define _AT(T,X)    X
#else
#define __AC(X,Y)   (X##Y)
#define _AC(X,Y)    __AC(X,Y)
#define _AT(T,X)    ((T)(X))
#endif

#define _UL(x)      (_AC(x, UL))
#define _ULL(x)     (_AC(x, ULL))

#define _BITUL(x)   (_UL(1) << (x))
#define _BITULL(x)  (_ULL(1) << (x))

#define UL(x)       (_UL(x))
#define ULL(x)      (_ULL(x))

#define __STR(s)    #s
#define STRINGIFY(s)    __STR(s)

/* clang-format on */
```

```
00047
00048 #endif
```

30.56 include/sbi/sbi_cppc.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for sbi_cppc.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_cppc_device

Functions

- int sbi_cppc_probe (unsigned long reg)
- int sbi_cppc_read (unsigned long reg, uint64_t *val)
- int sbi_cppc_write (unsigned long reg, uint64_t val)
- const struct sbi_cppc_device * sbi_cppc_get_device (void)
- void sbi_cppc_set_device (const struct sbi_cppc_device *dev)

30.56.1 Function Documentation

30.56.1.1 sbi_cppc_get_device()

```
const struct sbi_cppc_device * sbi_cppc_get_device (
            void  )
```

Here is the caller graph for this function:

30.56.1.2 sbi_cppc_probe()

```
int sbi_cppc_probe (
            unsigned long reg )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.56.1.3 sbi_cppc_read()

```
int sbi_cppc_read (
            unsigned long reg,
            uint64_t * val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.56.1.4 sbi_cppc_set_device()

```
void sbi_cppc_set_device (
            const struct sbi_cppc_device * dev )
```

Here is the caller graph for this function:

30.56.1.5 sbi_cppc_write()

```
int sbi_cppc_write (
            unsigned long reg,
            uint64_t val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.57 sbi_cppc.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Ventana Micro Systems Inc.
 *
 */

#ifndef __SBI_CPPC_H__
#define __SBI_CPPC_H__

#include <sbi/sbi_types.h>

struct sbi_cppc_device {
    char name[32];

    int (*cppc_probe)(unsigned long reg);

    int (*cppc_read)(unsigned long reg, uint64_t *val);

    int (*cppc_write)(unsigned long reg, uint64_t val);
};

int sbi_cppc_probe(unsigned long reg);
int sbi_cppc_read(unsigned long reg, uint64_t *val);
int sbi_cppc_write(unsigned long reg, uint64_t val);

const struct sbi_cppc_device *sbi_cppc_get_device(void);
void sbi_cppc_set_device(const struct sbi_cppc_device *dev);

#endif
```

30.58 include/sbi/sbi_csr_detect.h File Reference

```
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_csr_detect.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define csr_read_allowed(csr_num, trap)
- #define csr_write_allowed(csr_num, trap, value)

30.58.1 Macro Definition Documentation

30.58.1.1 csr_read_allowed

```
#define csr_read_allowed(
            csr_num,
            trap )
```

Value:

```
({                                  \
register ulong tinfo asm("a3") = (ulong)trap;          \
register ulong ttmp asm("a4");                  \
register ulong mtvec = sbi_hart_expected_trap_addr();        \
register ulong ret = 0;                         \
((struct sbi_trap_info *)(trap))->cause = 0;          \
asm volatile(                               \
    "add %[ttmp], %[tinfo], zero\n"             \
    "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"    \
    "csrr %[ret], %[csr]\n"                     \
    "csrw " STR(CSR_MTVEC) ", %[mtvec]"         \
    : [mtvec] "+&r"(mtvec), [tinfo] "+&r"(tinfo),       \
      [ttmp] "+&r"(ttmp), [ret] "=&r" (ret)         \
    : [csr] "i" (csr_num)                       \
    : "memory");                            \
ret;                                    \
})                                  \
```

30.58.1.2 csr_write_allowed

```
#define csr_write_allowed(
            csr_num,
            trap,
            value )
```

Value:

```
({                                  \
register ulong tinfo asm("a3") = (ulong)trap;          \
register ulong ttmp asm("a4");                  \
register ulong mtvec = sbi_hart_expected_trap_addr();        \
((struct sbi_trap_info *)(trap))->cause = 0;          \
asm volatile(                               \
    "add %[ttmp], %[tinfo], zero\n"             \
    "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"    \
    "csrw %[csr], %[val]\n"                     \
    "csrw " STR(CSR_MTVEC) ", %[mtvec]"         \
    : [mtvec] "+&r"(mtvec),                     \
      [tinfo] "+&r"(tinfo), [ttmp] "+&r"(ttmp)          \
    : [csr] "i" (csr_num), [val] "r" (value)            \
    : "memory");                            \
})                                  \
```

30.59 sbi_csr_detect.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Atish Patra <atish.patra@wdc.com>
 */

#ifndef __SBI_CSR_DETECT__H
#define __SBI_CSR_DETECT__H

#include <sbi/riscv_encoding.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_trap.h>

#define csr_read_allowed(csr_num, trap)                 \
    ({                                  \
    register ulong tinfo asm("a3") = (ulong)trap;           \
    register ulong ttmp asm("a4");                  \
    register ulong mtvec = sbi_hart_expected_trap_addr();        \
    register ulong ret = 0;                         \
    ((struct sbi_trap_info *)(trap))->cause = 0;            \
    asm volatile(                               \
        "add %[ttmp], %[tinfo], zero\n"             \
        "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"    \
        "csrr %[ret], %[csr]\n"                     \
        "csrw " STR(CSR_MTVEC) ", %[mtvec]"         \
        : [mtvec] "+&r"(mtvec), [tinfo] "+&r"(tinfo),       \
          [ttmp] "+&r"(ttmp), [ret] "=&r" (ret)         \
        : [csr] "i" (csr_num)                       \
        : "memory");                            \
    ret;                                    \
    })                                  \

```

```
#define csr_write_allowed(csr_num, trap, value)                    \
    ({                                                             \
    register ulong tinfo asm("a3") = (ulong)trap;                  \
    register ulong ttmp asm("a4");                                 \
    register ulong mtvec = sbi_hart_expected_trap_addr();          \
    ((struct sbi_trap_info *)(trap))->cause = 0;                   \
    asm volatile(                                                  \
        "add %[ttmp], %[tinfo], zero\n"                            \
        "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"           \
        "csrw %[csr], %[val]\n"                                    \
        "csrw " STR(CSR_MTVEC) ", %[mtvec]"                        \
        : [mtvec] "+&r"(mtvec),                                    \
          [tinfo] "+&r"(tinfo), [ttmp] "+&r"(ttmp)                 \
        : [csr] "i" (csr_num), [val] "r" (value)                   \
        : "memory");                                               \
    })                                                             \

#endif
```

30.60 include/sbi/sbi_dbtr.h File Reference

```
#include <sbi/riscv_dbtr.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_dbtr.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_dbtr_shmem
- struct sbi_dbtr_trigger
- struct sbi_dbtr_data_msg
- struct sbi_dbtr_id_msg
- struct sbi_dbtr_hart_triggers_state
- struct sbi_dbtr_shmem_entry

Macros

- #define TDATA1_GET_TYPE(_t1) EXTRACT_FIELD(_t1, RV_DBTR_BIT_MASK(TDATA1, TYPE))
- #define SET_TRIG_HW_INDEX(_state, _idx)
- #define SBI_DBTR_SHMEM_ALIGN_MASK ((__riscv_xlen / 8) - 1)

Enumerations

- enum { RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) }
- enum { RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) }

Functions

- int sbi_dbtr_init (struct sbi_scratch *scratch, bool coldboot)
- int sbi_dbtr_supported (void)
- int sbi_dbtr_setup_shmem (const struct sbi_domain *dom, unsigned long smode, unsigned long shmem_phys_lo, unsigned long shmem_phys_hi)
- int sbi_dbtr_num_trig (unsigned long trig_tdata1, unsigned long *out)
- int sbi_dbtr_read_trig (unsigned long smode, unsigned long trig_idx_base, unsigned long trig_count)
- int sbi_dbtr_install_trig (unsigned long smode, unsigned long trig_count, unsigned long *out)
- int sbi_dbtr_uninstall_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_enable_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_update_trig (unsigned long smode, unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_disable_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_get_total_triggers (void)

30.60.1 Macro Definition Documentation

30.60.1.1 SBI_DBTR_SHMEM_ALIGN_MASK

```
#define SBI_DBTR_SHMEM_ALIGN_MASK ((__riscv_xlen / 8) - 1)
```

30.60.1.2 SET_TRIG_HW_INDEX

```
#define SET_TRIG_HW_INDEX(
            _state,
            _idx )
```

Value:

```
do {                                    \
    _state &= ~RV_DBTR_BIT_MASK(TS, HW_IDX);     \
    _state |= (((unsigned long)_idx          \
            << RV_DBTR_BIT(TS, HW_IDX))      \
           & RV_DBTR_BIT_MASK(TS, HW_IDX)); \
}while (0);
```

30.60.1.3 TDATA1_GET_TYPE

```
#define TDATA1_GET_TYPE(
            _t1 )  EXTRACT_FIELD(_t1, RV_DBTR_BIT_MASK(TDATA1, TYPE))
```

30.60.2 Enumeration Type Documentation

30.60.2.1 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	

30.60.2.2 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

30.60.3 Function Documentation

30.60.3.1 sbi_dbtr_disable_trig()

```
int sbi_dbtr_disable_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.2 sbi_dbtr_enable_trig()

```
int sbi_dbtr_enable_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.3 sbi_dbtr_get_total_triggers()

```
int sbi_dbtr_get_total_triggers (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.4 sbi_dbtr_init()

```
int sbi_dbtr_init (
            struct sbi_scratch * scratch,
            bool coldboot )
```

Initialize debug triggers Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.5 sbi_dbtr_install_trig()

```
int sbi_dbtr_install_trig (
            unsigned long smode,
            unsigned long trig_count,
            unsigned long * out )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.6 sbi_dbtr_num_trig()

```
int sbi_dbtr_num_trig (
            unsigned long trig_tdata1,
            unsigned long * out )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.7 sbi_dbtr_read_trig()

```
int sbi_dbtr_read_trig (
            unsigned long smode,
            unsigned long trig_idx_base,
            unsigned long trig_count )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.8 sbi_dbtr_setup_shmem()

```
int sbi_dbtr_setup_shmem (
            const struct sbi_domain * dom,
            unsigned long smode,
            unsigned long shmem_phys_lo,
            unsigned long shmem_phys_hi )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.9 sbi_dbtr_supported()

```
int sbi_dbtr_supported (
            void )
```

SBI DBTR extension functions

30.60.3.10 sbi_dbtr_uninstall_trig()

```
int sbi_dbtr_uninstall_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.60.3.11 sbi_dbtr_update_trig()

```
int sbi_dbtr_update_trig (
            unsigned long smode,
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.61 sbi_dbtr.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Ventana Micro Systems, Inc.
 *
 * Authors:
 *   Himanshu Chauhan <hchauhan@ventanamicro.com>
 */

#ifndef __SBI_DBTR_H__
#define __SBI_DBTR_H__

#include <sbi/riscv_dbtr.h>
#include <sbi/sbi_types.h>

struct sbi_domain;

enum {
    RV_DBTR_DECLARE_BIT(TS, MAPPED, 0), /* trigger mapped to hw trigger */
    RV_DBTR_DECLARE_BIT(TS, U, 1),
    RV_DBTR_DECLARE_BIT(TS, S, 2),
    RV_DBTR_DECLARE_BIT(TS, VU, 3),
    RV_DBTR_DECLARE_BIT(TS, VS, 4),
    RV_DBTR_DECLARE_BIT(TS, HAVE_TRIG, 5), /* H/w dbtr details available */
    RV_DBTR_DECLARE_BIT(TS, HW_IDX, 8), /* Hardware index of trigger */
};

enum {
    RV_DBTR_DECLARE_BIT_MASK(TS, MAPPED, 1),
    RV_DBTR_DECLARE_BIT_MASK(TS, U, 1),
    RV_DBTR_DECLARE_BIT_MASK(TS, S, 1),
    RV_DBTR_DECLARE_BIT_MASK(TS, VU, 1),
    RV_DBTR_DECLARE_BIT_MASK(TS, VS, 1),
    RV_DBTR_DECLARE_BIT_MASK(TS, HAVE_TRIG, 1),
    RV_DBTR_DECLARE_BIT_MASK(TS, HW_IDX, (__riscv_xlen-9)),
};

#if __riscv_xlen == 64
#define SBI_DBTR_SHMEM_INVALID_ADDR 0xFFFFFFFFFFFFFFFFUL
#elif __riscv_xlen == 32
#define SBI_DBTR_SHMEM_INVALID_ADDR 0xFFFFFFFFUL
#else
#error "Unexpected __riscv_xlen"
#endif

struct sbi_dbtr_shmem {
    unsigned long phys_lo;
    unsigned long phys_hi;
};

struct sbi_dbtr_trigger {
    unsigned long index;
    unsigned long type_mask;
    unsigned long state;
    unsigned long tdata1;
    unsigned long tdata2;
    unsigned long tdata3;
};

struct sbi_dbtr_data_msg {
    unsigned long tstate;
    unsigned long tdata1;
    unsigned long tdata2;
    unsigned long tdata3;
};

struct sbi_dbtr_id_msg {
    unsigned long idx;
};

struct sbi_dbtr_hart_triggers_state {
    struct sbi_dbtr_trigger triggers[RV_MAX_TRIGGERS];
    struct sbi_dbtr_shmem shmem;
    u32 total_trigs;
    u32 available_trigs;
    u32 hartid;
    u32 probed;
};

#define TDATA1_GET_TYPE(_t1)                    \
    EXTRACT_FIELD(_t1, RV_DBTR_BIT_MASK(TDATA1, TYPE))

/* Set the hardware index of trigger in logical trigger state */
```

```
#define SET_TRIG_HW_INDEX(_state, _idx)                \
    do {                                               \
        _state &= ~RV_DBTR_BIT_MASK(TS, HW_IDX);       \
        _state |= (((unsigned long)_idx                \
                 << RV_DBTR_BIT(TS, HW_IDX))           \
                & RV_DBTR_BIT_MASK(TS, HW_IDX)); \
    }while (0);

struct sbi_dbtr_shmem_entry {
    struct sbi_dbtr_data_msg data;
    struct sbi_dbtr_id_msg id;
};

#define SBI_DBTR_SHMEM_ALIGN_MASK   ((__riscv_xlen / 8) - 1)

int sbi_dbtr_init(struct sbi_scratch *scratch, bool coldboot);

int sbi_dbtr_supported(void);
int sbi_dbtr_setup_shmem(const struct sbi_domain *dom, unsigned long smode,
             unsigned long shmem_phys_lo,
             unsigned long shmem_phys_hi);
int sbi_dbtr_num_trig(unsigned long trig_tdata1, unsigned long *out);
int sbi_dbtr_read_trig(unsigned long smode,
                unsigned long trig_idx_base, unsigned long trig_count);
int sbi_dbtr_install_trig(unsigned long smode,
                unsigned long trig_count, unsigned long *out);
int sbi_dbtr_uninstall_trig(unsigned long trig_idx_base,
                unsigned long trig_idx_mask);
int sbi_dbtr_enable_trig(unsigned long trig_idx_base,
             unsigned long trig_idx_mask);
int sbi_dbtr_update_trig(unsigned long smode,
             unsigned long trig_idx_base,
             unsigned long trig_idx_mask);
int sbi_dbtr_disable_trig(unsigned long trig_idx_base,
                unsigned long trig_idx_mask);

int sbi_dbtr_get_total_triggers(void);

#endif
```

30.62 include/sbi/sbi_domain.h File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_list.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_domain_context.h>
#include <sbi/sbi_domain_data.h>
```

Include dependency graph for sbi_domain.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_domain_memregion
- struct sbi_domain

Macros

- #define SBI_DOMAIN_MEMREGION_M_READABLE (1UL << 0)
- #define SBI_DOMAIN_MEMREGION_M_WRITABLE (1UL << 1)
- #define SBI_DOMAIN_MEMREGION_M_EXECUTABLE (1UL << 2)
- #define SBI_DOMAIN_MEMREGION_SU_READABLE (1UL << 3)
- #define SBI_DOMAIN_MEMREGION_SU_WRITABLE (1UL << 4)
- #define SBI_DOMAIN_MEMREGION_SU_EXECUTABLE (1UL << 5)
- #define SBI_DOMAIN_MEMREGION_ACCESS_MASK (0x3fUL)
- #define SBI_DOMAIN_MEMREGION_M_ACCESS_MASK (0x7UL)
- #define SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK (0x38UL)
- #define SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT (3)
- #define SBI_DOMAIN_MEMREGION_SHARED_RDONLY
- #define SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX
- #define SBI_DOMAIN_MEMREGION_SHARED_SUX_MX
- #define SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW
- #define SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW
- #define SBI_DOMAIN_MEMREGION_IS_SUR_MR(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SUX_MRX(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SUX_MX(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SURW_MRW(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SUR_MRW(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SHARED(_flags)
- #define SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS(__flags)
- #define SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS(__flags)
- #define SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS (1UL << 6)
- #define SBI_DOMAIN_MEMREGION_M_RWX
- #define SBI_DOMAIN_MEMREGION_SU_RWX
- #define SBI_DOMAIN_MEMREGION_READABLE
- #define SBI_DOMAIN_MEMREGION_WRITEABLE
- #define SBI_DOMAIN_MEMREGION_EXECUTABLE
- #define SBI_DOMAIN_MEMREGION_ENF_READABLE
- #define SBI_DOMAIN_MEMREGION_ENF_WRITABLE
- #define SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE
- #define SBI_DOMAIN_MEMREGION_MMIO (1UL << 31)
- #define sbi_domain_thishart_ptr() sbi_hartindex_to_domain(current_hartindex())
- #define sbi_domain_for_each(__d) sbi_list_for_each_entry(__d, &domain_list, node)
- #define sbi_domain_for_each_memregion(__d, __r) for ((__r) = (__d)->regions; (__r)->order; (__r)++)

Enumerations

- enum sbi_domain_access { SBI_DOMAIN_READ = (1UL << 0) , SBI_DOMAIN_WRITE = (1UL << 1) , SBI_DOMAIN_EXECUTE = (1UL << 2) , SBI_DOMAIN_MMIO = (1UL << 3) }

Functions

- struct sbi_domain * sbi_hartindex_to_domain (u32 hartindex)
- void sbi_update_hartindex_to_domain (u32 hartindex, struct sbi_domain *dom)
- bool sbi_domain_is_assigned_hart (const struct sbi_domain *dom, u32 hartindex)
- int sbi_domain_get_assigned_hartmask (const struct sbi_domain *dom, struct sbi_hartmask *mask)
- void sbi_domain_memregion_init (unsigned long addr, unsigned long size, unsigned long flags, struct sbi_domain_memregion *reg)
- bool sbi_domain_check_addr (const struct sbi_domain *dom, unsigned long addr, unsigned long mode, unsigned long access_flags)
- bool sbi_domain_check_addr_range (const struct sbi_domain *dom, unsigned long addr, unsigned long size, unsigned long mode, unsigned long access_flags)
- void sbi_domain_dump (const struct sbi_domain *dom, const char *suffix)
- void sbi_domain_dump_all (const char *suffix)
- int sbi_domain_register (struct sbi_domain *dom, const struct sbi_hartmask *assign_mask)
- int sbi_domain_root_add_memrange (unsigned long addr, unsigned long size, unsigned long align, unsigned long region_flags)
- int sbi_domain_finalize (struct sbi_scratch *scratch, u32 cold_hartid)
- int sbi_domain_init (struct sbi_scratch *scratch, u32 cold_hartid)

Variables

- struct sbi_domain root
- struct sbi_dlist domain_list

30.62.1 Macro Definition Documentation

30.62.1.1 sbi_domain_for_each

```
#define sbi_domain_for_each(
            __d )  sbi_list_for_each_entry(__d, &domain_list, node)
```

Iterate over each domain

30.62.1.2 sbi_domain_for_each_memregion

```
#define sbi_domain_for_each_memregion(
            __d,
            __r )  for ((__r) = (__d)->regions; (__r)->order; (__r)++)
```

Iterate over each memory region of a domain

30.62.1.3 SBI_DOMAIN_MEMREGION_ACCESS_MASK

```
#define SBI_DOMAIN_MEMREGION_ACCESS_MASK (0x3fUL)
```

30.62.1.4 SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE

```
#define SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_EXECUTABLE | \
 SBI_DOMAIN_MEMREGION_M_EXECUTABLE)
```

30.62.1.5 SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS

```
#define SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS (1UL << 6)
```

Bit to control if permissions are enforced on all modes

30.62.1.6 SBI_DOMAIN_MEMREGION_ENF_READABLE

```
#define SBI_DOMAIN_MEMREGION_ENF_READABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_READABLE | \
 SBI_DOMAIN_MEMREGION_M_READABLE)
```

30.62.1.7 SBI_DOMAIN_MEMREGION_ENF_WRITABLE

```
#define SBI_DOMAIN_MEMREGION_ENF_WRITABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
 SBI_DOMAIN_MEMREGION_M_WRITABLE)
```

30.62.1.8 SBI_DOMAIN_MEMREGION_EXECUTABLE

```
#define SBI_DOMAIN_MEMREGION_EXECUTABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_EXECUTABLE | \
 SBI_DOMAIN_MEMREGION_M_RWX)
```

30.62.1.9 SBI_DOMAIN_MEMREGION_IS_SHARED

```
#define SBI_DOMAIN_MEMREGION_IS_SHARED(
            _flags )
```

Value:

```
(SBI_DOMAIN_MEMREGION_IS_SUR_MR(_flags)  ||   \
 SBI_DOMAIN_MEMREGION_IS_SUX_MRX(_flags) || \
 SBI_DOMAIN_MEMREGION_IS_SUX_MX(_flags)  || \
 SBI_DOMAIN_MEMREGION_IS_SURW_MRW(_flags)|| \
 SBI_DOMAIN_MEMREGION_IS_SUR_MRW(_flags))
```

30.62.1.10 SBI_DOMAIN_MEMREGION_IS_SUR_MR

```
#define SBI_DOMAIN_MEMREGION_IS_SUR_MR(
            __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_RDONLY)
```

30.62.1.11 SBI_DOMAIN_MEMREGION_IS_SUR_MRW

```
#define SBI_DOMAIN_MEMREGION_IS_SUR_MRW(
            __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW)
```

30.62.1.12 SBI_DOMAIN_MEMREGION_IS_SURW_MRW

```
#define SBI_DOMAIN_MEMREGION_IS_SURW_MRW(
            __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW)
```

30.62.1.13 SBI_DOMAIN_MEMREGION_IS_SUX_MRX

```
#define SBI_DOMAIN_MEMREGION_IS_SUX_MRX(
            __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX)
```

30.62.1.14 SBI_DOMAIN_MEMREGION_IS_SUX_MX

```
#define SBI_DOMAIN_MEMREGION_IS_SUX_MX(
            __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_SUX_MX)
```

30.62.1.15 SBI_DOMAIN_MEMREGION_M_ACCESS_MASK

```
#define SBI_DOMAIN_MEMREGION_M_ACCESS_MASK (0x7UL)
```

30.62.1.16 SBI_DOMAIN_MEMREGION_M_EXECUTABLE

```
#define SBI_DOMAIN_MEMREGION_M_EXECUTABLE (1UL << 2)
```

30.62.1.17 SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS

```
#define SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS(
            __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_M_ACCESS_MASK) &&    \
 !(__flags & SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK))
```

30.62.1.18 SBI_DOMAIN_MEMREGION_M_READABLE

```
#define SBI_DOMAIN_MEMREGION_M_READABLE (1UL << 0)
```

Flags representing memory region attributes

30.62.1.19 SBI_DOMAIN_MEMREGION_M_RWX

```
#define SBI_DOMAIN_MEMREGION_M_RWX
```

Value:

```
(SBI_DOMAIN_MEMREGION_M_READABLE | \
 SBI_DOMAIN_MEMREGION_M_WRITABLE | \
 SBI_DOMAIN_MEMREGION_M_EXECUTABLE)
```

30.62.1.20 SBI_DOMAIN_MEMREGION_M_WRITABLE

```
#define SBI_DOMAIN_MEMREGION_M_WRITABLE (1UL << 1)
```

30.62.1.21 SBI_DOMAIN_MEMREGION_MMIO

```
#define SBI_DOMAIN_MEMREGION_MMIO (1UL << 31)
```

30.62.1.22 SBI_DOMAIN_MEMREGION_READABLE

```
#define SBI_DOMAIN_MEMREGION_READABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_READABLE | \
 SBI_DOMAIN_MEMREGION_M_RWX)
```

30.62.1.23 SBI_DOMAIN_MEMREGION_SHARED_RDONLY

```
#define SBI_DOMAIN_MEMREGION_SHARED_RDONLY
```

Value:

```
(SBI_DOMAIN_MEMREGION_M_READABLE |    \
 SBI_DOMAIN_MEMREGION_SU_READABLE)
```

30.62.1.24 SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW

```
#define SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW
```

Value:

```
(SBI_DOMAIN_MEMREGION_M_READABLE |    \
 SBI_DOMAIN_MEMREGION_M_WRITABLE |    \
 SBI_DOMAIN_MEMREGION_SU_READABLE)
```

30.62.1.25 SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW

```
#define SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW
```

Value:

```
(SBI_DOMAIN_MEMREGION_M_READABLE |    \
 SBI_DOMAIN_MEMREGION_M_WRITABLE |    \
```

```
SBI_DOMAIN_MEMREGION_SU_READABLE|  \
SBI_DOMAIN_MEMREGION_SU_WRITABLE)
```

30.62.1.26 SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX

```
#define SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX
```

Value:

```
(SBI_DOMAIN_MEMREGION_M_READABLE   | \
 SBI_DOMAIN_MEMREGION_M_EXECUTABLE |   \
 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)
```

30.62.1.27 SBI_DOMAIN_MEMREGION_SHARED_SUX_MX

```
#define SBI_DOMAIN_MEMREGION_SHARED_SUX_MX
```

Value:

```
(SBI_DOMAIN_MEMREGION_M_EXECUTABLE |   \
 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)
```

30.62.1.28 SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK

```
#define SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK (0x38UL)
```

30.62.1.29 SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT

```
#define SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT (3)
```

30.62.1.30 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE

```
#define SBI_DOMAIN_MEMREGION_SU_EXECUTABLE (1UL << 5)
```

30.62.1.31 SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS

```
#define SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS(
            __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK)  && \
 !(__flags & SBI_DOMAIN_MEMREGION_M_ACCESS_MASK))
```

30.62.1.32 SBI_DOMAIN_MEMREGION_SU_READABLE

```
#define SBI_DOMAIN_MEMREGION_SU_READABLE (1UL << 3)
```

30.62.1.33 SBI_DOMAIN_MEMREGION_SU_RWX

```
#define SBI_DOMAIN_MEMREGION_SU_RWX
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_READABLE | \
 SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)
```

30.62.1.34 SBI_DOMAIN_MEMREGION_SU_WRITABLE

```
#define SBI_DOMAIN_MEMREGION_SU_WRITABLE (1UL << 4)
```

30.62.1.35 SBI_DOMAIN_MEMREGION_WRITEABLE

```
#define SBI_DOMAIN_MEMREGION_WRITEABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
 SBI_DOMAIN_MEMREGION_M_RWX)
```

30.62.1.36 sbi_domain_thishart_ptr

```
#define sbi_domain_thishart_ptr( )  sbi_hartindex_to_domain(current_hartindex())
```

Get pointer to sbi_domain for current HART

30.62.2 Enumeration Type Documentation

30.62.2.1 sbi_domain_access

```
enum sbi_domain_access
```

Domain access types

Enumerator

SBI_DOMAIN_READ	
SBI_DOMAIN_WRITE	
SBI_DOMAIN_EXECUTE	
SBI_DOMAIN_MMIO	

30.62.3 Function Documentation

30.62.3.1 sbi_domain_check_addr()

```
bool sbi_domain_check_addr (
            const struct sbi_domain * dom,
            unsigned long addr,
            unsigned long mode,
            unsigned long access_flags )
```

Check whether we can access specified address for given mode and memory region flags under a domain

Parameters

dom	pointer to domain
addr	the address to be checked
mode	the privilege mode of access
access_flags	bitmask of domain access types (enum sbi_domain_access)

Returns

true if access allowed otherwise false

Here is the caller graph for this function:

30.62.3.2 sbi_domain_check_addr_range()

```
bool sbi_domain_check_addr_range (
            const struct sbi_domain * dom,
            unsigned long addr,
            unsigned long size,
            unsigned long mode,
            unsigned long access_flags )
```

Check whether we can access specified address range for given mode and memory region flags under a domain

Parameters

dom	pointer to domain
addr	the start of the address range to be checked
size	the size of the address range to be checked
mode	the privilege mode of access
access_flags	bitmask of domain access types (enum sbi_domain_access)

Returns

TRUE if access allowed otherwise FALSE

Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.3 sbi_domain_dump()

```
void sbi_domain_dump (
            const struct sbi_domain * dom,
            const char * suffix )
```

Dump domain details on the console Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.4 sbi_domain_dump_all()

```
void sbi_domain_dump_all (
            const char * suffix )
```

Dump all domain details on the console Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.5 sbi_domain_finalize()

```
int sbi_domain_finalize (
            struct sbi_scratch * scratch,
            u32 cold_hartid )
```

Finalize domain tables and startup non-root domains Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.6 sbi_domain_get_assigned_hartmask()

```
int sbi_domain_get_assigned_hartmask (
            const struct sbi_domain * dom,
            struct sbi_hartmask * mask )
```

Get the assigned HART mask for given domain

Parameters

dom	pointer to domain
mask	the output hartmask to fill

Returns

0 on success and SBI_Exxx ($<$ 0) on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.7 sbi_domain_init()

```
int sbi_domain_init (
            struct sbi_scratch * scratch,
            u32 cold_hartid )
```

Initialize domains Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.8 sbi_domain_is_assigned_hart()

```
bool sbi_domain_is_assigned_hart (
            const struct sbi_domain * dom,
            u32 hartindex )
```

Check whether given HART is assigned to specified domain

Parameters

dom	pointer to domain
hartindex	the HART index

Returns

true if HART is assigned to domain otherwise false

Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.9 sbi_domain_memregion_init()

```
void sbi_domain_memregion_init (
            unsigned long addr,
            unsigned long size,
            unsigned long flags,
            struct sbi_domain_memregion * reg )
```

Initialize a domain memory region based on it's physical address and size.

Parameters

addr	start physical address of memory region
size	physical size of memory region
flags	memory region flags
reg	pointer to memory region being initialized

Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.10 sbi_domain_register()

```
int sbi_domain_register (
            struct sbi_domain * dom,
            const struct sbi_hartmask * assign_mask )
```

Register a new domain

Parameters

dom	pointer to domain
assign_mask	pointer to HART mask of HARTs assigned to the domain

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.11 sbi_domain_root_add_memrange()

```
int sbi_domain_root_add_memrange (
            unsigned long addr,
            unsigned long size,
            unsigned long align,
            unsigned long region_flags )
```

Add a memory range with its flags to the root domain

Parameters

addr	start physical address of memory range
size	physical size of memory range
align	alignment of memory region
region_flags	memory range flags

Returns

0 on success

SBI_EALREADY if memory region conflicts with the existing one

SBI_EINVAL otherwise

Here is the call graph for this function:

Here is the caller graph for this function:

30.62.3.12 sbi_hartindex_to_domain()

```
struct sbi_domain * sbi_hartindex_to_domain (
            u32 hartindex )
```

Get pointer to sbi_domain from HART index Here is the caller graph for this function:

30.62.3.13 sbi_update_hartindex_to_domain()

```
void sbi_update_hartindex_to_domain (
            u32 hartindex,
            struct sbi_domain * dom )
```

Update HART local pointer to point to specified domain Here is the caller graph for this function:

30.62.4 Variable Documentation

30.62.4.1 domain_list

```
struct sbi_dlist domain_list  [extern]
```

Head of linked list of domains

30.62.4.2 root

```
struct sbi_domain root  [extern]
```

The root domain instance

30.63 sbi_domain.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2020 Western Digital Corporation or its affiliates.
00005  *
00006  * Authors:
00007  *   Anup Patel <anup.patel@wdc.com>
00008  */
00009
00010 #ifndef __SBI_DOMAIN_H__
00011 #define __SBI_DOMAIN_H__
00012
00013 #include <sbi/riscv_locks.h>
00014 #include <sbi/sbi_list.h>
00015 #include <sbi/sbi_types.h>
00016 #include <sbi/sbi_hartmask.h>
00017 #include <sbi/sbi_domain_context.h>
00018 #include <sbi/sbi_domain_data.h>
00019
00020 struct sbi_scratch;
00021
00023 enum sbi_domain_access {
```

```
    SBI_DOMAIN_READ = (1UL « 0),
    SBI_DOMAIN_WRITE = (1UL « 1),
    SBI_DOMAIN_EXECUTE = (1UL « 2),
    SBI_DOMAIN_MMIO = (1UL « 3)
};

struct sbi_domain_memregion {
    unsigned long order;
    unsigned long base;
#define SBI_DOMAIN_MEMREGION_M_READABLE     (1UL « 0)
#define SBI_DOMAIN_MEMREGION_M_WRITABLE     (1UL « 1)
#define SBI_DOMAIN_MEMREGION_M_EXECUTABLE   (1UL « 2)
#define SBI_DOMAIN_MEMREGION_SU_READABLE    (1UL « 3)
#define SBI_DOMAIN_MEMREGION_SU_WRITABLE    (1UL « 4)
#define SBI_DOMAIN_MEMREGION_SU_EXECUTABLE  (1UL « 5)

#define SBI_DOMAIN_MEMREGION_ACCESS_MASK    (0x3fUL)
#define SBI_DOMAIN_MEMREGION_M_ACCESS_MASK  (0x7UL)
#define SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK (0x38UL)

#define SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT    (3)

#define SBI_DOMAIN_MEMREGION_SHARED_RDONLY      \
        (SBI_DOMAIN_MEMREGION_M_READABLE |  \
         SBI_DOMAIN_MEMREGION_SU_READABLE)

#define SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX     \
        (SBI_DOMAIN_MEMREGION_M_READABLE   |    \
         SBI_DOMAIN_MEMREGION_M_EXECUTABLE |    \
         SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)

#define SBI_DOMAIN_MEMREGION_SHARED_SUX_MX      \
        (SBI_DOMAIN_MEMREGION_M_EXECUTABLE |    \
         SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)

#define SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW        \
        (SBI_DOMAIN_MEMREGION_M_READABLE |  \
         SBI_DOMAIN_MEMREGION_M_WRITABLE |  \
         SBI_DOMAIN_MEMREGION_SU_READABLE|  \
         SBI_DOMAIN_MEMREGION_SU_WRITABLE)

#define SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW     \
        (SBI_DOMAIN_MEMREGION_M_READABLE |  \
         SBI_DOMAIN_MEMREGION_M_WRITABLE |  \
         SBI_DOMAIN_MEMREGION_SU_READABLE)

    /* Shared read-only region between M and SU mode */
#define SBI_DOMAIN_MEMREGION_IS_SUR_MR(__flags)          \
        ((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
         SBI_DOMAIN_MEMREGION_SHARED_RDONLY)

    /* Shared region: SU execute-only and M read/execute */
#define SBI_DOMAIN_MEMREGION_IS_SUX_MRX(__flags)         \
        ((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
         SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX)

    /* Shared region: SU and M execute-only */
#define SBI_DOMAIN_MEMREGION_IS_SUX_MX(__flags)          \
        ((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
         SBI_DOMAIN_MEMREGION_SHARED_SUX_MX)

    /* Shared region: SU and M read/write */
#define SBI_DOMAIN_MEMREGION_IS_SURW_MRW(__flags)        \
        ((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
         SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW)

    /* Shared region: SU read-only and M read/write */
#define SBI_DOMAIN_MEMREGION_IS_SUR_MRW(__flags)         \
        ((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
         SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW)

    /*
     * Check if region flags match with any of the above
     * mentioned shared region type
     */
#define SBI_DOMAIN_MEMREGION_IS_SHARED(_flags)          \
        (SBI_DOMAIN_MEMREGION_IS_SUR_MR(_flags)  || \
         SBI_DOMAIN_MEMREGION_IS_SUX_MRX(_flags) || \
         SBI_DOMAIN_MEMREGION_IS_SUX_MX(_flags)  || \
         SBI_DOMAIN_MEMREGION_IS_SURW_MRW(_flags)|| \
         SBI_DOMAIN_MEMREGION_IS_SUR_MRW(_flags))

#define SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS(__flags)         \
        ((__flags & SBI_DOMAIN_MEMREGION_M_ACCESS_MASK) &&  \
         !(__flags & SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK))

#define SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS(__flags)            \
```

```
        ((__flags & SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK)  &&    \
         !(__flags & SBI_DOMAIN_MEMREGION_M_ACCESS_MASK))

#define SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS    (1UL « 6)

#define SBI_DOMAIN_MEMREGION_M_RWX      \
                (SBI_DOMAIN_MEMREGION_M_READABLE | \
                 SBI_DOMAIN_MEMREGION_M_WRITABLE | \
                 SBI_DOMAIN_MEMREGION_M_EXECUTABLE)

#define SBI_DOMAIN_MEMREGION_SU_RWX     \
                (SBI_DOMAIN_MEMREGION_SU_READABLE | \
                 SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
                 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)

/* Unrestricted M-mode accesses but enfoced on SU-mode */
#define SBI_DOMAIN_MEMREGION_READABLE       \
                (SBI_DOMAIN_MEMREGION_SU_READABLE | \
                 SBI_DOMAIN_MEMREGION_M_RWX)
#define SBI_DOMAIN_MEMREGION_WRITEABLE      \
                (SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
                 SBI_DOMAIN_MEMREGION_M_RWX)
#define SBI_DOMAIN_MEMREGION_EXECUTABLE     \
                (SBI_DOMAIN_MEMREGION_SU_EXECUTABLE | \
                 SBI_DOMAIN_MEMREGION_M_RWX)

/* Enforced accesses across all modes */
#define SBI_DOMAIN_MEMREGION_ENF_READABLE   \
                (SBI_DOMAIN_MEMREGION_SU_READABLE | \
                 SBI_DOMAIN_MEMREGION_M_READABLE)
#define SBI_DOMAIN_MEMREGION_ENF_WRITABLE   \
                (SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
                 SBI_DOMAIN_MEMREGION_M_WRITABLE)
#define SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE \
                (SBI_DOMAIN_MEMREGION_SU_EXECUTABLE | \
                 SBI_DOMAIN_MEMREGION_M_EXECUTABLE)

#define SBI_DOMAIN_MEMREGION_MMIO       (1UL « 31)
    unsigned long flags;
};

struct sbi_domain {
    struct sbi_dlist node;
    struct sbi_domain_data_priv data_priv;
    u32 index;
    struct sbi_hartmask assigned_harts;
    spinlock_t assigned_harts_lock;
    char name[64];
    const struct sbi_hartmask *possible_harts;
    struct sbi_domain_memregion *regions;
    u32 boot_hartid;
    unsigned long next_arg1;
    unsigned long next_addr;
    unsigned long next_mode;
    bool system_reset_allowed;
    bool system_suspend_allowed;
    bool fw_region_inited;
};

extern struct sbi_domain root;

struct sbi_domain *sbi_hartindex_to_domain(u32 hartindex);

void sbi_update_hartindex_to_domain(u32 hartindex, struct sbi_domain *dom);

#define sbi_domain_thishart_ptr() \
    sbi_hartindex_to_domain(current_hartindex())

extern struct sbi_dlist domain_list;

#define sbi_domain_for_each(__d) \
    sbi_list_for_each_entry(__d, &domain_list, node)

#define sbi_domain_for_each_memregion(__d, __r) \
    for ((__r) = (__d)->regions; (__r)->order; (__r)++)

bool sbi_domain_is_assigned_hart(const struct sbi_domain *dom, u32 hartindex);

int sbi_domain_get_assigned_hartmask(const struct sbi_domain *dom,
                     struct sbi_hartmask *mask);

void sbi_domain_memregion_init(unsigned long addr,
                unsigned long size,
                unsigned long flags,
                struct sbi_domain_memregion *reg);

bool sbi_domain_check_addr(const struct sbi_domain *dom,
```

```
                unsigned long addr, unsigned long mode,
                unsigned long access_flags);

bool sbi_domain_check_addr_range(const struct sbi_domain *dom,
                  unsigned long addr, unsigned long size,
                  unsigned long mode,
                  unsigned long access_flags);

void sbi_domain_dump(const struct sbi_domain *dom, const char *suffix);

void sbi_domain_dump_all(const char *suffix);

int sbi_domain_register(struct sbi_domain *dom,
             const struct sbi_hartmask *assign_mask);

int sbi_domain_root_add_memrange(unsigned long addr, unsigned long size,
                unsigned long align, unsigned long region_flags);

int sbi_domain_finalize(struct sbi_scratch *scratch, u32 cold_hartid);

int sbi_domain_init(struct sbi_scratch *scratch, u32 cold_hartid);

#endif
```

30.64 include/sbi/sbi_domain_context.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_domain_context.h:

This graph shows which files directly or indirectly include this file:

Functions

- int sbi_domain_context_enter (struct sbi_domain *dom)
- int sbi_domain_context_exit (void)
- int sbi_domain_context_init (void)
- void sbi_domain_context_deinit (void)

30.64.1 Function Documentation

30.64.1.1 sbi_domain_context_deinit()

```
void sbi_domain_context_deinit (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.64.1.2 sbi_domain_context_enter()

```
int sbi_domain_context_enter (
            struct sbi_domain * dom )
```

Enter a specific domain context synchronously

Parameters

dom	pointer to domain

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

30.64.1.3 sbi_domain_context_exit()

```
int sbi_domain_context_exit (
            void  )
```

Exit the current domain context, and then return to the caller of sbi_domain_context_enter or attempt to start the next domain context to be initialized

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

30.64.1.4 sbi_domain_context_init()

```
int sbi_domain_context_init (
            void  )
```

Initialize domain context support

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.65 sbi_domain_context.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) IPADS@SJTU 2023. All rights reserved.
 */

#ifndef __SBI_DOMAIN_CONTEXT_H__
#define __SBI_DOMAIN_CONTEXT_H__

#include <sbi/sbi_types.h>

struct sbi_domain;

int sbi_domain_context_enter(struct sbi_domain *dom);

int sbi_domain_context_exit(void);

int sbi_domain_context_init(void);

/* Deinitialize domain context support */
void sbi_domain_context_deinit(void);

#endif // __SBI_DOMAIN_CONTEXT_H__
```

30.66 include/sbi/sbi_domain_data.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for sbi_domain_data.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_domain_data_priv
- struct sbi_domain_data

Macros

- #define SBI_DOMAIN_MAX_DATA_PTRS 32

Functions

- void ∗ sbi_domain_data_ptr (struct sbi_domain ∗dom, struct sbi_domain_data ∗data)
- int sbi_domain_setup_data (struct sbi_domain ∗dom)
- void sbi_domain_cleanup_data (struct sbi_domain ∗dom)
- int sbi_domain_register_data (struct sbi_domain_data ∗data)
- void sbi_domain_unregister_data (struct sbi_domain_data ∗data)

30.66.1 Macro Definition Documentation

30.66.1.1 SBI_DOMAIN_MAX_DATA_PTRS

```
#define SBI_DOMAIN_MAX_DATA_PTRS 32
```

Maximum domain data per-domain

30.66.2 Function Documentation

30.66.2.1 sbi_domain_cleanup_data()

```
void sbi_domain_cleanup_data (
            struct sbi_domain * dom )
```

Cleanup all domain data for a domain

Parameters

dom	pointer to domain

Note: This function is used internally within domain framework. Here is the call graph for this function:

Here is the caller graph for this function:

30.66.2.2 sbi_domain_data_ptr()

```
void * sbi_domain_data_ptr (
            struct sbi_domain * dom,
            struct sbi_domain_data * data )
```

Get per-domain data pointer for a given domain

Parameters

dom	pointer to domain
data	pointer to domain data

Returns

per-domain data pointer

Here is the caller graph for this function:

30.66.2.3 sbi_domain_register_data()

```
int sbi_domain_register_data (
            struct sbi_domain_data * data )
```

Register a domain data

Parameters

hndl	pointer to domain data

Returns

0 on success and negative error code on failure

Note: This function must be used only in cold boot path. Here is the call graph for this function:

Here is the caller graph for this function:

30.66.2.4 sbi_domain_setup_data()

```
int sbi_domain_setup_data (
            struct sbi_domain * dom )
```

Setup all domain data for a domain

Parameters

dom	pointer to domain

Returns

0 on success and negative error code on failure

Note: This function is used internally within domain framework. Here is the call graph for this function:

Here is the caller graph for this function:

30.66.2.5 sbi_domain_unregister_data()

```
void sbi_domain_unregister_data (
            struct sbi_domain_data * data )
```

Unregister a domain data

Parameters

hndl	pointer to domain data

Note: This function must be used only in cold boot path. Here is the call graph for this function:

Here is the caller graph for this function:

30.67 sbi_domain_data.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2024 Ventana Micro Systems Inc.
 */

#ifndef __SBI_DOMAIN_DATA_H__
#define __SBI_DOMAIN_DATA_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>

struct sbi_domain;

#define SBI_DOMAIN_MAX_DATA_PTRS        32

struct sbi_domain_data_priv {
    void *idx_to_data_ptr[SBI_DOMAIN_MAX_DATA_PTRS];
};

struct sbi_domain_data {
    struct sbi_dlist head;
    unsigned long data_idx;
    unsigned long data_size;
    int (*data_setup)(struct sbi_domain *dom,
              struct sbi_domain_data *data, void *data_ptr);
    void (*data_cleanup)(struct sbi_domain *dom,
                 struct sbi_domain_data *data, void *data_ptr);
};

void *sbi_domain_data_ptr(struct sbi_domain *dom, struct sbi_domain_data *data);

int sbi_domain_setup_data(struct sbi_domain *dom);

void sbi_domain_cleanup_data(struct sbi_domain *dom);

int sbi_domain_register_data(struct sbi_domain_data *data);

void sbi_domain_unregister_data(struct sbi_domain_data *data);

#endif
```

30.68 include/sbi/sbi_ecall.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for sbi_ecall.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_ecall_return
- struct sbi_ecall_extension

Macros

- #define SBI_ECALL_VERSION_MAJOR 2
- #define SBI_ECALL_VERSION_MINOR 0
- #define SBI_OPENSBI_IMPID 1

Functions

- u16 sbi_ecall_version_major (void)
- u16 sbi_ecall_version_minor (void)
- unsigned long sbi_ecall_get_impid (void)
- void sbi_ecall_set_impid (unsigned long impid)
- struct sbi_ecall_extension * sbi_ecall_find_extension (unsigned long extid)
- void sbi_ecall_get_extensions_str (char *exts_str, int exts_str_size, bool experimental)
- int sbi_ecall_register_extension (struct sbi_ecall_extension *ext)
- void sbi_ecall_unregister_extension (struct sbi_ecall_extension *ext)
- int sbi_ecall_handler (struct sbi_trap_context *tcntx)
- int sbi_ecall_init (void)

30.68.1 Macro Definition Documentation

30.68.1.1 SBI_ECALL_VERSION_MAJOR

```
#define SBI_ECALL_VERSION_MAJOR 2
```

30.68.1.2 SBI_ECALL_VERSION_MINOR

```
#define SBI_ECALL_VERSION_MINOR 0
```

30.68.1.3 SBI_OPENSBI_IMPID

```
#define SBI_OPENSBI_IMPID 1
```

30.68.2 Function Documentation

30.68.2.1 sbi_ecall_find_extension()

```
struct sbi_ecall_extension * sbi_ecall_find_extension (
            unsigned long extid )
```

Here is the caller graph for this function:

30.68.2.2 sbi_ecall_get_extensions_str()

```
void sbi_ecall_get_extensions_str (
            char * exts_str,
            int exts_str_size,
            bool experimental )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.68.2.3 sbi_ecall_get_impid()

```
unsigned long sbi_ecall_get_impid (
            void )
```

Here is the caller graph for this function:

30.68.2.4 sbi_ecall_handler()

```
int sbi_ecall_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.68.2.5 sbi_ecall_init()

```
int sbi_ecall_init (
            void  )
```

Here is the caller graph for this function:

30.68.2.6 sbi_ecall_register_extension()

```
int sbi_ecall_register_extension (
            struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.68.2.7 sbi_ecall_set_impid()

```
void sbi_ecall_set_impid (
            unsigned long impid )
```

30.68.2.8 sbi_ecall_unregister_extension()

```
void sbi_ecall_unregister_extension (
```

```
            struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:

30.68.2.9 sbi_ecall_version_major()

```
u16 sbi_ecall_version_major (
            void  )
```

Here is the caller graph for this function:

30.68.2.10 sbi_ecall_version_minor()

```
u16 sbi_ecall_version_minor (
            void  )
```

Here is the caller graph for this function:

30.69 sbi_ecall.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_ECALL_H__
#define __SBI_ECALL_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>

#define SBI_ECALL_VERSION_MAJOR     2
#define SBI_ECALL_VERSION_MINOR     0
#define SBI_OPENSBI_IMPID       1

struct sbi_trap_regs;
struct sbi_trap_context;

struct sbi_ecall_return {
    /* Return flag to skip register update */
    bool skip_regs_update;
```

```
    /* Return value */
    unsigned long value;
};

struct sbi_ecall_extension {
    /* head is used by the extension list */
    struct sbi_dlist head;
    /* short name of the extension */
    char name[8];
    /*
     * extid_start and extid_end specify the range for this extension. As
     * the initial range may be wider than the valid runtime range, the
     * register_extensions callback is responsible for narrowing the range
     * before other callbacks may be invoked.
     */
    unsigned long extid_start;
    unsigned long extid_end;
    /* flag showing whether given extension is experimental or not */
    bool experimental;
    /*
     * register_extensions
     *
     * Calls sbi_ecall_register_extension() one or more times to register
     * extension ID range(s) which should be handled by this extension.
     * More than one sbi_ecall_extension struct and
     * sbi_ecall_register_extension() call is necessary when the supported
     * extension ID ranges have gaps. Additionally, extension availability
     * must be checked before registering, which means, when this callback
     * returns, only valid extension IDs from the initial range, which are
     * also available, have been registered.
     */
    int (* register_extensions)(void);
    /*
     * probe
     *
     * Implements the Base extension's probe function for the extension. As
     * the register_extensions callback ensures that no other extension
     * callbacks will be invoked when the extension is not available, then
     * probe can never fail. However, an extension may choose to set
     * out_val to a nonzero value other than one. In those cases, it should
     * implement this callback.
     */
    int (* probe)(unsigned long extid, unsigned long *out_val);
    /*
     * handle
     *
     * This is the extension handler. register_extensions ensures it is
     * never invoked with an invalid or unavailable extension ID.
     */
    int (* handle)(unsigned long extid, unsigned long funcid,
                   struct sbi_trap_regs *regs,
                   struct sbi_ecall_return *out);
};

u16 sbi_ecall_version_major(void);

u16 sbi_ecall_version_minor(void);

unsigned long sbi_ecall_get_impid(void);

void sbi_ecall_set_impid(unsigned long impid);

struct sbi_ecall_extension *sbi_ecall_find_extension(unsigned long extid);

void sbi_ecall_get_extensions_str(char *exts_str, int exts_str_size, bool experimental);

int sbi_ecall_register_extension(struct sbi_ecall_extension *ext);

void sbi_ecall_unregister_extension(struct sbi_ecall_extension *ext);

int sbi_ecall_handler(struct sbi_trap_context *tcntx);

int sbi_ecall_init(void);

#endif
```

30.70 include/sbi/sbi_ecall_interface.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_ecall_interface.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_pmu_event_info

Macros

- #define SBI_EXT_0_1_SET_TIMER 0x0
- #define SBI_EXT_0_1_CONSOLE_PUTCHAR 0x1
- #define SBI_EXT_0_1_CONSOLE_GETCHAR 0x2
- #define SBI_EXT_0_1_CLEAR_IPI 0x3
- #define SBI_EXT_0_1_SEND_IPI 0x4
- #define SBI_EXT_0_1_REMOTE_FENCE_I 0x5
- #define SBI_EXT_0_1_REMOTE_SFENCE_VMA 0x6
- #define SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID 0x7
- #define SBI_EXT_0_1_SHUTDOWN 0x8
- #define SBI_EXT_BASE 0x10
- #define SBI_EXT_TIME 0x54494D45
- #define SBI_EXT_IPI 0x735049
- #define SBI_EXT_RFENCE 0x52464E43
- #define SBI_EXT_HSM 0x48534D
- #define SBI_EXT_SRST 0x53525354
- #define SBI_EXT_PMU 0x504D55
- #define SBI_EXT_DBCN 0x4442434E
- #define SBI_EXT_SUSP 0x53555350
- #define SBI_EXT_CPPC 0x43505043
- #define SBI_EXT_DBTR 0x44425452
- #define SBI_EXT_SSE 0x535345
- #define SBI_EXT_FWFT 0x46574654
- #define SBI_EXT_MPXY 0x4D505859
- #define SBI_EXT_BASE_GET_SPEC_VERSION 0x0

- #define SBI_EXT_BASE_GET_IMP_ID 0x1
- #define SBI_EXT_BASE_GET_IMP_VERSION 0x2
- #define SBI_EXT_BASE_PROBE_EXT 0x3
- #define SBI_EXT_BASE_GET_MVENDORID 0x4
- #define SBI_EXT_BASE_GET_MARCHID 0x5
- #define SBI_EXT_BASE_GET_MIMPID 0x6
- #define SBI_EXT_TIME_SET_TIMER 0x0
- #define SBI_EXT_IPI_SEND_IPI 0x0
- #define SBI_EXT_RFENCE_REMOTE_FENCE_I 0x0
- #define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA 0x1
- #define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID 0x2
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID 0x3
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA 0x4
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID 0x5
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA 0x6
- #define SBI_EXT_HSM_HART_START 0x0
- #define SBI_EXT_HSM_HART_STOP 0x1
- #define SBI_EXT_HSM_HART_GET_STATUS 0x2
- #define SBI_EXT_HSM_HART_SUSPEND 0x3
- #define SBI_HSM_STATE_STARTED 0x0
- #define SBI_HSM_STATE_STOPPED 0x1
- #define SBI_HSM_STATE_START_PENDING 0x2
- #define SBI_HSM_STATE_STOP_PENDING 0x3
- #define SBI_HSM_STATE_SUSPENDED 0x4
- #define SBI_HSM_STATE_SUSPEND_PENDING 0x5
- #define SBI_HSM_STATE_RESUME_PENDING 0x6
- #define SBI_HSM_SUSP_BASE_MASK 0x7fffffff
- #define SBI_HSM_SUSP_NON_RET_BIT 0x80000000
- #define SBI_HSM_SUSP_PLAT_BASE 0x10000000
- #define SBI_HSM_SUSPEND_RET_DEFAULT 0x00000000
- #define SBI_HSM_SUSPEND_RET_PLATFORM SBI_HSM_SUSP_PLAT_BASE
- #define SBI_HSM_SUSPEND_RET_LAST SBI_HSM_SUSP_BASE_MASK
- #define SBI_HSM_SUSPEND_NON_RET_DEFAULT SBI_HSM_SUSP_NON_RET_BIT
- #define SBI_HSM_SUSPEND_NON_RET_PLATFORM
- #define SBI_HSM_SUSPEND_NON_RET_LAST
- #define SBI_EXT_SRST_RESET 0x0
- #define SBI_SRST_RESET_TYPE_SHUTDOWN 0x0
- #define SBI_SRST_RESET_TYPE_COLD_REBOOT 0x1
- #define SBI_SRST_RESET_TYPE_WARM_REBOOT 0x2
- #define SBI_SRST_RESET_TYPE_LAST SBI_SRST_RESET_TYPE_WARM_REBOOT
- #define SBI_SRST_RESET_REASON_NONE 0x0
- #define SBI_SRST_RESET_REASON_SYSFAIL 0x1
- #define SBI_EXT_PMU_NUM_COUNTERS 0x0
- #define SBI_EXT_PMU_COUNTER_GET_INFO 0x1
- #define SBI_EXT_PMU_COUNTER_CFG_MATCH 0x2
- #define SBI_EXT_PMU_COUNTER_START 0x3
- #define SBI_EXT_PMU_COUNTER_STOP 0x4
- #define SBI_EXT_PMU_COUNTER_FW_READ 0x5
- #define SBI_EXT_PMU_COUNTER_FW_READ_HI 0x6
- #define SBI_EXT_PMU_SNAPSHOT_SET_SHMEM 0x7
- #define SBI_EXT_PMU_EVENT_GET_INFO 0x8
- #define SBI_EXT_DBTR_NUM_TRIGGERS 0x0
- #define SBI_EXT_DBTR_SETUP_SHMEM 0x1
- #define SBI_EXT_DBTR_TRIGGER_READ 0x2
- #define SBI_EXT_DBTR_TRIGGER_INSTALL 0x3

- #define SBI_EXT_DBTR_TRIGGER_UPDATE 0x4
- #define SBI_EXT_DBTR_TRIGGER_UNINSTALL 0x5
- #define SBI_EXT_DBTR_TRIGGER_ENABLE 0x6
- #define SBI_EXT_DBTR_TRIGGER_DISABLE 0x7
- #define SBI_EXT_FWFT_SET 0x0
- #define SBI_EXT_FWFT_GET 0x1
- #define SBI_FWFT_GLOBAL_FEATURE_BIT (1 << 31)
- #define SBI_FWFT_PLATFORM_FEATURE_BIT (1 << 30)
- #define SBI_FWFT_SET_FLAG_LOCK (1 << 0)
- #define SBI_PMU_EVENT_IDX_MASK 0xFFFFF
- #define SBI_PMU_EVENT_IDX_TYPE_OFFSET 16
- #define SBI_PMU_EVENT_IDX_TYPE_MASK (0xF << SBI_PMU_EVENT_IDX_TYPE_OFFSET)
- #define SBI_PMU_EVENT_IDX_CODE_MASK 0xFFFF
- #define SBI_PMU_EVENT_RAW_IDX 0x20000
- #define SBI_PMU_EVENT_RAW_V2_IDX 0x30000
- #define SBI_PMU_EVENT_IDX_INVALID 0xFFFFFFFF
- #define SBI_PMU_EVENT_HW_CACHE_OPS_RESULT 0x1
- #define SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK 0x6
- #define SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET 1
- #define SBI_PMU_EVENT_HW_CACHE_ID_MASK 0xfff8
- #define SBI_PMU_EVENT_HW_CACHE_ID_OFFSET 3
- #define SBI_PMU_CFG_FLAG_SKIP_MATCH (1 << 0)
- #define SBI_PMU_CFG_FLAG_CLEAR_VALUE (1 << 1)
- #define SBI_PMU_CFG_FLAG_AUTO_START (1 << 2)
- #define SBI_PMU_CFG_FLAG_SET_VUINH (1 << 3)
- #define SBI_PMU_CFG_FLAG_SET_VSINH (1 << 4)
- #define SBI_PMU_CFG_FLAG_SET_UINH (1 << 5)
- #define SBI_PMU_CFG_FLAG_SET_SINH (1 << 6)
- #define SBI_PMU_CFG_FLAG_SET_MINH (1 << 7)
- #define SBI_PMU_START_FLAG_SET_INIT_VALUE (1 << 0)
- #define SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT (1 << 1)
- #define SBI_PMU_STOP_FLAG_RESET (1 << 0)
- #define SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT (1 << 1)
- #define SBI_EXT_DBCN_CONSOLE_WRITE 0x0
- #define SBI_EXT_DBCN_CONSOLE_READ 0x1
- #define SBI_EXT_DBCN_CONSOLE_WRITE_BYTE 0x2
- #define SBI_EXT_SUSP_SUSPEND 0x0
- #define SBI_SUSP_SLEEP_TYPE_SUSPEND 0x0
- #define SBI_SUSP_SLEEP_TYPE_LAST SBI_SUSP_SLEEP_TYPE_SUSPEND
- #define SBI_SUSP_PLATFORM_SLEEP_START 0x80000000
- #define SBI_EXT_CPPC_PROBE 0x0
- #define SBI_EXT_CPPC_READ 0x1
- #define SBI_EXT_CPPC_READ_HI 0x2
- #define SBI_EXT_CPPC_WRITE 0x3
- #define SBI_EXT_SSE_READ_ATTR 0x00000000
- #define SBI_EXT_SSE_WRITE_ATTR 0x00000001
- #define SBI_EXT_SSE_REGISTER 0x00000002
- #define SBI_EXT_SSE_UNREGISTER 0x00000003
- #define SBI_EXT_SSE_ENABLE 0x00000004
- #define SBI_EXT_SSE_DISABLE 0x00000005
- #define SBI_EXT_SSE_COMPLETE 0x00000006
- #define SBI_EXT_SSE_INJECT 0x00000007
- #define SBI_EXT_SSE_HART_UNMASK 0x00000008
- #define SBI_EXT_SSE_HART_MASK 0x00000009
- #define SBI_SSE_ATTR_STATUS_STATE_OFFSET 0

- #define SBI_SSE_ATTR_STATUS_STATE_MASK 0x3
- #define SBI_SSE_ATTR_STATUS_PENDING_OFFSET 2
- #define SBI_SSE_ATTR_STATUS_INJECT_OFFSET 3
- #define SBI_SSE_ATTR_CONFIG_ONESHOT (1 << 0)
- #define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP BIT(0)
- #define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE BIT(1)
- #define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV BIT(2)
- #define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP BIT(3)
- #define SBI_SSE_EVENT_LOCAL_RAS 0x00000000
- #define SBI_SSE_EVENT_LOCAL_DOUBLE_TRAP 0x00000001
- #define SBI_SSE_EVENT_LOCAL_PLAT_0_START 0x00004000
- #define SBI_SSE_EVENT_LOCAL_PLAT_0_END 0x00007fff
- #define SBI_SSE_EVENT_GLOBAL_RAS 0x00008000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_0_START 0x0000c000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_0_END 0x0000ffff
- #define SBI_SSE_EVENT_LOCAL_PMU 0x00010000
- #define SBI_SSE_EVENT_LOCAL_PLAT_1_START 0x00014000
- #define SBI_SSE_EVENT_LOCAL_PLAT_1_END 0x00017fff
- #define SBI_SSE_EVENT_GLOBAL_PLAT_1_START 0x0001c000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_1_END 0x0001ffff
- #define SBI_SSE_EVENT_LOCAL_PLAT_2_START 0x00024000
- #define SBI_SSE_EVENT_LOCAL_PLAT_2_END 0x00027fff
- #define SBI_SSE_EVENT_GLOBAL_PLAT_2_START 0x0002c000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_2_END 0x0002ffff
- #define SBI_SSE_EVENT_LOCAL_SOFTWARE 0xffff0000
- #define SBI_SSE_EVENT_LOCAL_PLAT_3_START 0xffff4000
- #define SBI_SSE_EVENT_LOCAL_PLAT_3_END 0xffff7fff
- #define SBI_SSE_EVENT_GLOBAL_SOFTWARE 0xffff8000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_3_START 0xffffc000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_3_END 0xffffffff
- #define SBI_SSE_EVENT_GLOBAL_BIT (1 << 15)
- #define SBI_SSE_EVENT_PLATFORM_BIT (1 << 14)
- #define SBI_EXT_MPXY_SET_SHMEM 0x0
- #define SBI_EXT_MPXY_GET_CHANNEL_IDS 0x1
- #define SBI_EXT_MPXY_READ_ATTRS 0x2
- #define SBI_EXT_MPXY_WRITE_ATTRS 0x3
- #define SBI_EXT_MPXY_SEND_MSG_WITH_RESP 0x4
- #define SBI_EXT_MPXY_SEND_MSG_NO_RESP 0x5
- #define SBI_EXT_MPXY_GET_NOTIFICATION_EVENTS 0x6
- #define SBI_SPEC_VERSION_MAJOR_OFFSET 24
- #define SBI_SPEC_VERSION_MAJOR_MASK 0x7f
- #define SBI_SPEC_VERSION_MINOR_MASK 0xffffff
- #define SBI_EXT_VENDOR_START 0x09000000
- #define SBI_EXT_VENDOR_END 0x09FFFFFF
- #define SBI_EXT_FIRMWARE_START 0x0A000000
- #define SBI_EXT_FIRMWARE_END 0x0AFFFFFF
- #define SBI_SUCCESS 0
- #define SBI_ERR_FAILED -1
- #define SBI_ERR_NOT_SUPPORTED -2
- #define SBI_ERR_INVALID_PARAM -3
- #define SBI_ERR_DENIED -4
- #define SBI_ERR_INVALID_ADDRESS -5
- #define SBI_ERR_ALREADY_AVAILABLE -6
- #define SBI_ERR_ALREADY_STARTED -7
- #define SBI_ERR_ALREADY_STOPPED -8

- #define SBI_ERR_NO_SHMEM -9
- #define SBI_ERR_INVALID_STATE -10
- #define SBI_ERR_BAD_RANGE -11
- #define SBI_ERR_TIMEOUT -12
- #define SBI_ERR_IO -13
- #define SBI_LAST_ERR SBI_ERR_BAD_RANGE

Enumerations

- enum sbi_fwft_feature_t { SBI_FWFT_MISALIGNED_EXC_DELEG = 0x0 , SBI_FWFT_LANDING_PAD = 0x1 , SBI_FWFT_SHADOW_STACK = 0x2 , SBI_FWFT_DOUBLE_TRAP = 0x3 , SBI_FWFT_PTE_AD_HW_UPDATING = 0x4 , SBI_FWFT_POINTER_MASKING_PMLEN = 0x5 , SBI_FWFT_LOCAL_RESERVED_START = 0x6 , SBI_FWFT_LOCAL_RESERVED_END = 0x3fffffff , SBI_FWFT_LOCAL_PLATFORM_START = 0x40000000 , SBI_FWFT_LOCAL_PLATFORM_END = 0x7fffffff , SBI_FWFT_GLOBAL_RESERVED_START = 0x80000000 , SBI_FWFT_GLOBAL_RESERVED_END = 0xbfffffff , SBI_FWFT_GLOBAL_PLATFORM_START = 0xc0000000 , SBI_FWFT_GLOBAL_PLATFORM_END = 0xffffffff }
- enum sbi_pmu_hw_generic_events_t { SBI_PMU_HW_NO_EVENT = 0 , SBI_PMU_HW_CPU_CYCLES = 1 , SBI_PMU_HW_INSTRUCTIONS = 2 , SBI_PMU_HW_CACHE_REFERENCES = 3 , SBI_PMU_HW_CACHE_MISSES = 4 , SBI_PMU_HW_BRANCH_INSTRUCTIONS = 5 , SBI_PMU_HW_BRANCH_MISSES = 6 , SBI_PMU_HW_BUS_CYCLES = 7 , SBI_PMU_HW_STALLED_CYCLES_FRONTEND = 8 , SBI_PMU_HW_STALLED_CYCLES_BACKEND = 9 , SBI_PMU_HW_REF_CPU_CYCLES = 10 , SBI_PMU_HW_GENERAL_MAX }
- enum sbi_pmu_hw_cache_id { SBI_PMU_HW_CACHE_L1D = 0 , SBI_PMU_HW_CACHE_L1I = 1 , SBI_PMU_HW_CACHE_LL = 2 , SBI_PMU_HW_CACHE_DTLB = 3 , SBI_PMU_HW_CACHE_ITLB = 4 , SBI_PMU_HW_CACHE_BPU = 5 , SBI_PMU_HW_CACHE_NODE = 6 , SBI_PMU_HW_CACHE_MAX }
- enum sbi_pmu_hw_cache_op_id { SBI_PMU_HW_CACHE_OP_READ = 0 , SBI_PMU_HW_CACHE_OP_WRITE = 1 , SBI_PMU_HW_CACHE_OP_PREFETCH = 2 , SBI_PMU_HW_CACHE_OP_MAX }
- enum sbi_pmu_hw_cache_op_result_id { SBI_PMU_HW_CACHE_RESULT_ACCESS = 0 , SBI_PMU_HW_CACHE_RESULT_ = 1 , SBI_PMU_HW_CACHE_RESULT_MAX }
- enum sbi_pmu_fw_event_code_id { SBI_PMU_FW_MISALIGNED_LOAD = 0 , SBI_PMU_FW_MISALIGNED_STORE = 1 , SBI_PMU_FW_ACCESS_LOAD = 2 , SBI_PMU_FW_ACCESS_STORE = 3 , SBI_PMU_FW_ILLEGAL_INSN = 4 , SBI_PMU_FW_SET_TIMER = 5 , SBI_PMU_FW_IPI_SENT = 6 , SBI_PMU_FW_IPI_RECVD = 7 , SBI_PMU_FW_FENCE_I_SENT = 8 , SBI_PMU_FW_FENCE_I_RECVD = 9 , SBI_PMU_FW_SFENCE_VMA_SENT = 10 , SBI_PMU_FW_SFENCE_VMA_RCVD = 11 , SBI_PMU_FW_SFENCE_VMA_ASID_SENT = 12 , SBI_PMU_FW_SFENCE_VMA_ASID_RCVD = 13 , SBI_PMU_FW_HFENCE_GVMA_SENT = 14 , SBI_PMU_FW_HFENCE_GVMA_RCVD = 15 , SBI_PMU_FW_HFENCE_GVMA_VMID_SENT = 16 , SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD = 17 , SBI_PMU_FW_HFENCE_VVMA_SENT = 18 , SBI_PMU_FW_HFENCE_VVMA_RCVD = 19 , SBI_PMU_FW_HFENCE_VVMA_ASID_SENT = 20 , SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD = 21 , SBI_PMU_FW_MAX , SBI_PMU_FW_RESERVED_MAX = 0xFFFE , SBI_PMU_FW_PLATFORM = 0xFFFF }
- enum sbi_pmu_event_type_id { SBI_PMU_EVENT_TYPE_HW = 0x0 , SBI_PMU_EVENT_TYPE_HW_CACHE = 0x1 , SBI_PMU_EVENT_TYPE_HW_RAW = 0x2 , SBI_PMU_EVENT_TYPE_HW_RAW_V2 = 0x3 , SBI_PMU_EVENT_TYPE_FW = 0xf , SBI_PMU_EVENT_TYPE_MAX }
- enum sbi_pmu_ctr_type { SBI_PMU_CTR_TYPE_HW = 0 , SBI_PMU_CTR_TYPE_FW }
- enum sbi_cppc_reg_id { SBI_CPPC_HIGHEST_PERF = 0x00000000 , SBI_CPPC_NOMINAL_PERF = 0x00000001 , SBI_CPPC_LOW_NON_LINEAR = 0x00000002 , SBI_CPPC_LOWEST_PERF = 0x00000003 , SBI_CPPC_GUARANTEED_PERF = 0x00000004 , SBI_CPPC_DESIRED_PERF = 0x00000005 ,

SBI_CPPC_MIN_PERF = 0x00000006 , SBI_CPPC_MAX_PERF = 0x00000007 , SBI_CPPC_PERF_REDUC_TOLERANCE = 0x00000008 , SBI_CPPC_TIME_WINDOW = 0x00000009 , SBI_CPPC_CTR_WRAP_TIME = 0x0000000A , SBI_CPPC_REFERENCE_CTR = 0x0000000B , SBI_CPPC_DELIVERED_CTR = 0x0000000C , SBI_CPPC_PERF_LIMITED = 0x0000000D , SBI_CPPC_ENABLE = 0x0000000E , SBI_CPPC_AUTO_SEL_ENABLE = 0x0000000F , SBI_CPPC_AUTO_ACT_WINDOW = 0x00000010 , SBI_CPPC_ENERGY_PERF_PREFERENCE = 0x00000011 , SBI_CPPC_REFERENCE_PERF = 0x00000012 , SBI_CPPC_LOWEST_FREQ = 0x00000013 , SBI_CPPC_NOMINAL_FREQ = 0x00000014 , SBI_CPPC_ACPI_LAST = SBI_CPPC_NOMINAL_FREQ , SBI_CPPC_TRANSITION_LATENCY = 0x80000000 , SBI_CPPC_NON_ACPI_LAST = SBI_CPPC_↩ TRANSITION_LATENCY }

- enum sbi_sse_attr_id { SBI_SSE_ATTR_STATUS = 0x00000000 , SBI_SSE_ATTR_PRIO = 0x00000001 , SBI_SSE_ATTR_CONFIG = 0x00000002 , SBI_SSE_ATTR_PREFERRED_HART = 0x00000003 , SBI_SSE_ATTR_ENTRY_PC = 0x00000004 , SBI_SSE_ATTR_ENTRY_ARG = 0x00000005 , SBI_SSE_ATTR_INTERRUPTED = 0x00000006 , SBI_SSE_ATTR_INTERRUPTED_FLAGS = 0x00000007 , SBI_SSE_ATTR_INTERRUPTED_A6 = 0x00000008 , SBI_SSE_ATTR_INTERRUPTED_A7 = 0x00000009 , SBI_SSE_ATTR_MAX = 0x0000000A }
- enum sbi_sse_state { SBI_SSE_STATE_UNUSED = 0 , SBI_SSE_STATE_REGISTERED = 1 , SBI_SSE_STATE_ENABLED = 2 , SBI_SSE_STATE_RUNNING = 3 }

30.70.1 Macro Definition Documentation

30.70.1.1 SBI_ERR_ALREADY_AVAILABLE

```
#define SBI_ERR_ALREADY_AVAILABLE -6
```

30.70.1.2 SBI_ERR_ALREADY_STARTED

```
#define SBI_ERR_ALREADY_STARTED -7
```

30.70.1.3 SBI_ERR_ALREADY_STOPPED

```
#define SBI_ERR_ALREADY_STOPPED -8
```

30.70.1.4 SBI_ERR_BAD_RANGE

```
#define SBI_ERR_BAD_RANGE -11
```

30.70.1.5 SBI_ERR_DENIED

```
#define SBI_ERR_DENIED -4
```

30.70.1.6 SBI_ERR_FAILED

```
#define SBI_ERR_FAILED -1
```

30.70.1.7 SBI_ERR_INVALID_ADDRESS

```
#define SBI_ERR_INVALID_ADDRESS -5
```

30.70.1.8 SBI_ERR_INVALID_PARAM

```
#define SBI_ERR_INVALID_PARAM -3
```

30.70.1.9 SBI_ERR_INVALID_STATE

```
#define SBI_ERR_INVALID_STATE -10
```

30.70.1.10 SBI_ERR_IO

```
#define SBI_ERR_IO -13
```

30.70.1.11 SBI_ERR_NO_SHMEM

```
#define SBI_ERR_NO_SHMEM -9
```

30.70.1.12 SBI_ERR_NOT_SUPPORTED

```
#define SBI_ERR_NOT_SUPPORTED -2
```

30.70.1.13 SBI_ERR_TIMEOUT

```
#define SBI_ERR_TIMEOUT -12
```

30.70.1.14 SBI_EXT_0_1_CLEAR_IPI

```
#define SBI_EXT_0_1_CLEAR_IPI 0x3
```

30.70.1.15 SBI_EXT_0_1_CONSOLE_GETCHAR

```
#define SBI_EXT_0_1_CONSOLE_GETCHAR 0x2
```

30.70.1.16 SBI_EXT_0_1_CONSOLE_PUTCHAR

```
#define SBI_EXT_0_1_CONSOLE_PUTCHAR 0x1
```

30.70.1.17 SBI_EXT_0_1_REMOTE_FENCE_I

```
#define SBI_EXT_0_1_REMOTE_FENCE_I 0x5
```

30.70.1.18 SBI_EXT_0_1_REMOTE_SFENCE_VMA

```
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA 0x6
```

30.70.1.19 SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID

```
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID 0x7
```

30.70.1.20 SBI_EXT_0_1_SEND_IPI

```
#define SBI_EXT_0_1_SEND_IPI 0x4
```

30.70.1.21 SBI_EXT_0_1_SET_TIMER

```
#define SBI_EXT_0_1_SET_TIMER 0x0
```

30.70.1.22 SBI_EXT_0_1_SHUTDOWN

```
#define SBI_EXT_0_1_SHUTDOWN 0x8
```

30.70.1.23 SBI_EXT_BASE

```
#define SBI_EXT_BASE 0x10
```

30.70.1.24 SBI_EXT_BASE_GET_IMP_ID

```
#define SBI_EXT_BASE_GET_IMP_ID 0x1
```

30.70.1.25 SBI_EXT_BASE_GET_IMP_VERSION

```
#define SBI_EXT_BASE_GET_IMP_VERSION 0x2
```

30.70.1.26 SBI_EXT_BASE_GET_MARCHID

```
#define SBI_EXT_BASE_GET_MARCHID 0x5
```

30.70.1.27 SBI_EXT_BASE_GET_MIMPID

```
#define SBI_EXT_BASE_GET_MIMPID 0x6
```

30.70.1.28 SBI_EXT_BASE_GET_MVENDORID

```
#define SBI_EXT_BASE_GET_MVENDORID 0x4
```

30.70.1.29 SBI_EXT_BASE_GET_SPEC_VERSION

```
#define SBI_EXT_BASE_GET_SPEC_VERSION 0x0
```

30.70.1.30 SBI_EXT_BASE_PROBE_EXT

```
#define SBI_EXT_BASE_PROBE_EXT 0x3
```

30.70.1.31 SBI_EXT_CPPC

```
#define SBI_EXT_CPPC 0x43505043
```

30.70.1.32 SBI_EXT_CPPC_PROBE

```
#define SBI_EXT_CPPC_PROBE 0x0
```

30.70.1.33 SBI_EXT_CPPC_READ

```
#define SBI_EXT_CPPC_READ 0x1
```

30.70.1.34 SBI_EXT_CPPC_READ_HI

```
#define SBI_EXT_CPPC_READ_HI 0x2
```

30.70.1.35 SBI_EXT_CPPC_WRITE

```
#define SBI_EXT_CPPC_WRITE 0x3
```

30.70.1.36 SBI_EXT_DBCN

```
#define SBI_EXT_DBCN 0x4442434E
```

30.70.1.37 SBI_EXT_DBCN_CONSOLE_READ

```
#define SBI_EXT_DBCN_CONSOLE_READ 0x1
```

30.70.1.38 SBI_EXT_DBCN_CONSOLE_WRITE

```
#define SBI_EXT_DBCN_CONSOLE_WRITE 0x0
```

30.70.1.39 SBI_EXT_DBCN_CONSOLE_WRITE_BYTE

```
#define SBI_EXT_DBCN_CONSOLE_WRITE_BYTE 0x2
```

30.70.1.40 SBI_EXT_DBTR

```
#define SBI_EXT_DBTR 0x44425452
```

30.70.1.41 SBI_EXT_DBTR_NUM_TRIGGERS

```
#define SBI_EXT_DBTR_NUM_TRIGGERS 0x0
```

30.70.1.42 SBI_EXT_DBTR_SETUP_SHMEM

```
#define SBI_EXT_DBTR_SETUP_SHMEM 0x1
```

30.70.1.43 SBI_EXT_DBTR_TRIGGER_DISABLE

```
#define SBI_EXT_DBTR_TRIGGER_DISABLE 0x7
```

30.70.1.44 SBI_EXT_DBTR_TRIGGER_ENABLE

```
#define SBI_EXT_DBTR_TRIGGER_ENABLE 0x6
```

30.70.1.45 SBI_EXT_DBTR_TRIGGER_INSTALL

```
#define SBI_EXT_DBTR_TRIGGER_INSTALL 0x3
```

30.70.1.46 SBI_EXT_DBTR_TRIGGER_READ

```
#define SBI_EXT_DBTR_TRIGGER_READ 0x2
```

30.70.1.47 SBI_EXT_DBTR_TRIGGER_UNINSTALL

```
#define SBI_EXT_DBTR_TRIGGER_UNINSTALL 0x5
```

30.70.1.48 SBI_EXT_DBTR_TRIGGER_UPDATE

```
#define SBI_EXT_DBTR_TRIGGER_UPDATE 0x4
```

30.70.1.49 SBI_EXT_FIRMWARE_END

```
#define SBI_EXT_FIRMWARE_END 0x0AFFFFFF
```

30.70.1.50 SBI_EXT_FIRMWARE_START

```
#define SBI_EXT_FIRMWARE_START 0x0A000000
```

30.70.1.51 SBI_EXT_FWFT

```
#define SBI_EXT_FWFT 0x46574654
```

30.70.1.52 SBI_EXT_FWFT_GET

```
#define SBI_EXT_FWFT_GET 0x1
```

30.70.1.53 SBI_EXT_FWFT_SET

```
#define SBI_EXT_FWFT_SET 0x0
```

30.70.1.54 SBI_EXT_HSM

```
#define SBI_EXT_HSM 0x48534D
```

30.70.1.55 SBI_EXT_HSM_HART_GET_STATUS

```
#define SBI_EXT_HSM_HART_GET_STATUS 0x2
```

30.70.1.56 SBI_EXT_HSM_HART_START

```
#define SBI_EXT_HSM_HART_START 0x0
```

30.70.1.57 SBI_EXT_HSM_HART_STOP

```
#define SBI_EXT_HSM_HART_STOP 0x1
```

30.70.1.58 SBI_EXT_HSM_HART_SUSPEND

```
#define SBI_EXT_HSM_HART_SUSPEND 0x3
```

30.70.1.59 SBI_EXT_IPI

```
#define SBI_EXT_IPI 0x735049
```

30.70.1.60 SBI_EXT_IPI_SEND_IPI

```
#define SBI_EXT_IPI_SEND_IPI 0x0
```

30.70.1.61 SBI_EXT_MPXY

```
#define SBI_EXT_MPXY 0x4D505859
```

30.70.1.62 SBI_EXT_MPXY_GET_CHANNEL_IDS

```
#define SBI_EXT_MPXY_GET_CHANNEL_IDS 0x1
```

30.70.1.63 SBI_EXT_MPXY_GET_NOTIFICATION_EVENTS

```
#define SBI_EXT_MPXY_GET_NOTIFICATION_EVENTS 0x6
```

30.70.1.64 SBI_EXT_MPXY_READ_ATTRS

```
#define SBI_EXT_MPXY_READ_ATTRS 0x2
```

30.70.1.65 SBI_EXT_MPXY_SEND_MSG_NO_RESP

```
#define SBI_EXT_MPXY_SEND_MSG_NO_RESP 0x5
```

30.70.1.66 SBI_EXT_MPXY_SEND_MSG_WITH_RESP

```
#define SBI_EXT_MPXY_SEND_MSG_WITH_RESP 0x4
```

30.70.1.67 SBI_EXT_MPXY_SET_SHMEM

```
#define SBI_EXT_MPXY_SET_SHMEM 0x0
```

30.70.1.68 SBI_EXT_MPXY_WRITE_ATTRS

```
#define SBI_EXT_MPXY_WRITE_ATTRS 0x3
```

30.70.1.69 SBI_EXT_PMU

```
#define SBI_EXT_PMU 0x504D55
```

30.70.1.70 SBI_EXT_PMU_COUNTER_CFG_MATCH

```
#define SBI_EXT_PMU_COUNTER_CFG_MATCH 0x2
```

30.70.1.71 SBI_EXT_PMU_COUNTER_FW_READ

```
#define SBI_EXT_PMU_COUNTER_FW_READ 0x5
```

30.70.1.72 SBI_EXT_PMU_COUNTER_FW_READ_HI

```
#define SBI_EXT_PMU_COUNTER_FW_READ_HI 0x6
```

30.70.1.73 SBI_EXT_PMU_COUNTER_GET_INFO

```
#define SBI_EXT_PMU_COUNTER_GET_INFO 0x1
```

30.70.1.74 SBI_EXT_PMU_COUNTER_START

```
#define SBI_EXT_PMU_COUNTER_START 0x3
```

30.70.1.75 SBI_EXT_PMU_COUNTER_STOP

```
#define SBI_EXT_PMU_COUNTER_STOP 0x4
```

30.70.1.76 SBI_EXT_PMU_EVENT_GET_INFO

```
#define SBI_EXT_PMU_EVENT_GET_INFO 0x8
```

30.70.1.77 SBI_EXT_PMU_NUM_COUNTERS

```
#define SBI_EXT_PMU_NUM_COUNTERS 0x0
```

30.70.1.78 SBI_EXT_PMU_SNAPSHOT_SET_SHMEM

```
#define SBI_EXT_PMU_SNAPSHOT_SET_SHMEM 0x7
```

30.70.1.79 SBI_EXT_RFENCE

```
#define SBI_EXT_RFENCE 0x52464E43
```

30.70.1.80 SBI_EXT_RFENCE_REMOTE_FENCE_I

```
#define SBI_EXT_RFENCE_REMOTE_FENCE_I 0x0
```

30.70.1.81 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA 0x4
```

30.70.1.82 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID 0x3
```

30.70.1.83 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA 0x6
```

30.70.1.84 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID 0x5
```

30.70.1.85 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA

```
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA 0x1
```

30.70.1.86 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID

```
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID 0x2
```

30.70.1.87 SBI_EXT_SRST

```
#define SBI_EXT_SRST 0x53525354
```

30.70.1.88 SBI_EXT_SRST_RESET

```
#define SBI_EXT_SRST_RESET 0x0
```

30.70.1.89 SBI_EXT_SSE

```
#define SBI_EXT_SSE 0x535345
```

30.70.1.90 SBI_EXT_SSE_COMPLETE

```
#define SBI_EXT_SSE_COMPLETE 0x00000006
```

30.70.1.91 SBI_EXT_SSE_DISABLE

```
#define SBI_EXT_SSE_DISABLE 0x00000005
```

30.70.1.92 SBI_EXT_SSE_ENABLE

```
#define SBI_EXT_SSE_ENABLE 0x00000004
```

30.70.1.93 SBI_EXT_SSE_HART_MASK

```
#define SBI_EXT_SSE_HART_MASK 0x00000009
```

30.70.1.94 SBI_EXT_SSE_HART_UNMASK

```
#define SBI_EXT_SSE_HART_UNMASK 0x00000008
```

30.70.1.95 SBI_EXT_SSE_INJECT

```
#define SBI_EXT_SSE_INJECT 0x00000007
```

30.70.1.96 SBI_EXT_SSE_READ_ATTR

```
#define SBI_EXT_SSE_READ_ATTR 0x00000000
```

30.70.1.97 SBI_EXT_SSE_REGISTER

```
#define SBI_EXT_SSE_REGISTER 0x00000002
```

30.70.1.98 SBI_EXT_SSE_UNREGISTER

```
#define SBI_EXT_SSE_UNREGISTER 0x00000003
```

30.70.1.99 SBI_EXT_SSE_WRITE_ATTR

```
#define SBI_EXT_SSE_WRITE_ATTR 0x00000001
```

30.70.1.100 SBI_EXT_SUSP

```
#define SBI_EXT_SUSP 0x53555350
```

30.70.1.101 SBI_EXT_SUSP_SUSPEND

```
#define SBI_EXT_SUSP_SUSPEND 0x0
```

30.70.1.102 SBI_EXT_TIME

```
#define SBI_EXT_TIME 0x54494D45
```

30.70.1.103 SBI_EXT_TIME_SET_TIMER

```
#define SBI_EXT_TIME_SET_TIMER 0x0
```

30.70.1.104 SBI_EXT_VENDOR_END

```
#define SBI_EXT_VENDOR_END 0x09FFFFFF
```

30.70.1.105 SBI_EXT_VENDOR_START

```
#define SBI_EXT_VENDOR_START 0x09000000
```

30.70.1.106 SBI_FWFT_GLOBAL_FEATURE_BIT

```
#define SBI_FWFT_GLOBAL_FEATURE_BIT (1 << 31)
```

30.70.1.107 SBI_FWFT_PLATFORM_FEATURE_BIT

```
#define SBI_FWFT_PLATFORM_FEATURE_BIT (1 << 30)
```

30.70.1.108 SBI_FWFT_SET_FLAG_LOCK

```
#define SBI_FWFT_SET_FLAG_LOCK (1 << 0)
```

30.70.1.109 SBI_HSM_STATE_RESUME_PENDING

```
#define SBI_HSM_STATE_RESUME_PENDING 0x6
```

30.70.1.110 SBI_HSM_STATE_START_PENDING

```
#define SBI_HSM_STATE_START_PENDING 0x2
```

30.70.1.111 SBI_HSM_STATE_STARTED

```
#define SBI_HSM_STATE_STARTED 0x0
```

30.70.1.112 SBI_HSM_STATE_STOP_PENDING

```
#define SBI_HSM_STATE_STOP_PENDING 0x3
```

30.70.1.113 SBI_HSM_STATE_STOPPED

```
#define SBI_HSM_STATE_STOPPED 0x1
```

30.70.1.114 SBI_HSM_STATE_SUSPEND_PENDING

```
#define SBI_HSM_STATE_SUSPEND_PENDING 0x5
```

30.70.1.115 SBI_HSM_STATE_SUSPENDED

```
#define SBI_HSM_STATE_SUSPENDED 0x4
```

30.70.1.116 SBI_HSM_SUSP_BASE_MASK

```
#define SBI_HSM_SUSP_BASE_MASK 0x7fffffff
```

30.70.1.117 SBI_HSM_SUSP_NON_RET_BIT

```
#define SBI_HSM_SUSP_NON_RET_BIT 0x80000000
```

30.70.1.118 SBI_HSM_SUSP_PLAT_BASE

```
#define SBI_HSM_SUSP_PLAT_BASE 0x10000000
```

30.70.1.119 SBI_HSM_SUSPEND_NON_RET_DEFAULT

```
#define SBI_HSM_SUSPEND_NON_RET_DEFAULT SBI_HSM_SUSP_NON_RET_BIT
```

30.70.1.120 SBI_HSM_SUSPEND_NON_RET_LAST

```
#define SBI_HSM_SUSPEND_NON_RET_LAST
```

Value:

```
(SBI_HSM_SUSP_NON_RET_BIT | \
SBI_HSM_SUSP_BASE_MASK)
```

30.70.1.121 SBI_HSM_SUSPEND_NON_RET_PLATFORM

```
#define SBI_HSM_SUSPEND_NON_RET_PLATFORM
```

Value:

```
(SBI_HSM_SUSP_NON_RET_BIT | \
SBI_HSM_SUSP_PLAT_BASE)
```

30.70.1.122 SBI_HSM_SUSPEND_RET_DEFAULT

```
#define SBI_HSM_SUSPEND_RET_DEFAULT 0x00000000
```

30.70.1.123 SBI_HSM_SUSPEND_RET_LAST

```
#define SBI_HSM_SUSPEND_RET_LAST SBI_HSM_SUSP_BASE_MASK
```

30.70.1.124 SBI_HSM_SUSPEND_RET_PLATFORM

```
#define SBI_HSM_SUSPEND_RET_PLATFORM SBI_HSM_SUSP_PLAT_BASE
```

30.70.1.125 SBI_LAST_ERR

```
#define SBI_LAST_ERR SBI_ERR_BAD_RANGE
```

30.70.1.126 SBI_PMU_CFG_FLAG_AUTO_START

```
#define SBI_PMU_CFG_FLAG_AUTO_START (1 << 2)
```

30.70.1.127 SBI_PMU_CFG_FLAG_CLEAR_VALUE

```
#define SBI_PMU_CFG_FLAG_CLEAR_VALUE (1 << 1)
```

30.70.1.128 SBI_PMU_CFG_FLAG_SET_MINH

```
#define SBI_PMU_CFG_FLAG_SET_MINH (1 << 7)
```

30.70.1.129 SBI_PMU_CFG_FLAG_SET_SINH

```
#define SBI_PMU_CFG_FLAG_SET_SINH (1 << 6)
```

30.70.1.130 SBI_PMU_CFG_FLAG_SET_UINH

```
#define SBI_PMU_CFG_FLAG_SET_UINH (1 << 5)
```

30.70.1.131 SBI_PMU_CFG_FLAG_SET_VSINH

```
#define SBI_PMU_CFG_FLAG_SET_VSINH (1 << 4)
```

30.70.1.132 SBI_PMU_CFG_FLAG_SET_VUINH

```
#define SBI_PMU_CFG_FLAG_SET_VUINH (1 << 3)
```

30.70.1.133 SBI_PMU_CFG_FLAG_SKIP_MATCH

```
#define SBI_PMU_CFG_FLAG_SKIP_MATCH (1 << 0)
```

30.70.1.134 SBI_PMU_EVENT_HW_CACHE_ID_MASK

```
#define SBI_PMU_EVENT_HW_CACHE_ID_MASK 0xfff8
```

30.70.1.135 SBI_PMU_EVENT_HW_CACHE_ID_OFFSET

```
#define SBI_PMU_EVENT_HW_CACHE_ID_OFFSET 3
```

30.70.1.136 SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK

```
#define SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK 0x6
```

30.70.1.137 SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET

```
#define SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET 1
```

30.70.1.138 SBI_PMU_EVENT_HW_CACHE_OPS_RESULT

```
#define SBI_PMU_EVENT_HW_CACHE_OPS_RESULT 0x1
```

30.70.1.139 SBI_PMU_EVENT_IDX_CODE_MASK

```
#define SBI_PMU_EVENT_IDX_CODE_MASK 0xFFFF
```

30.70.1.140 SBI_PMU_EVENT_IDX_INVALID

```
#define SBI_PMU_EVENT_IDX_INVALID 0xFFFFFFFF
```

30.70.1.141 SBI_PMU_EVENT_IDX_MASK

```
#define SBI_PMU_EVENT_IDX_MASK 0xFFFFF
```

30.70.1.142 SBI_PMU_EVENT_IDX_TYPE_MASK

```
#define SBI_PMU_EVENT_IDX_TYPE_MASK (0xF << SBI_PMU_EVENT_IDX_TYPE_OFFSET)
```

30.70.1.143 SBI_PMU_EVENT_IDX_TYPE_OFFSET

```
#define SBI_PMU_EVENT_IDX_TYPE_OFFSET 16
```

30.70.1.144 SBI_PMU_EVENT_RAW_IDX

```
#define SBI_PMU_EVENT_RAW_IDX 0x20000
```

30.70.1.145 SBI_PMU_EVENT_RAW_V2_IDX

```
#define SBI_PMU_EVENT_RAW_V2_IDX 0x30000
```

30.70.1.146 SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT

```
#define SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT (1 << 1)
```

30.70.1.147 SBI_PMU_START_FLAG_SET_INIT_VALUE

```
#define SBI_PMU_START_FLAG_SET_INIT_VALUE (1 << 0)
```

30.70.1.148 SBI_PMU_STOP_FLAG_RESET

```
#define SBI_PMU_STOP_FLAG_RESET (1 << 0)
```

30.70.1.149 SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT

```
#define SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT (1 << 1)
```

30.70.1.150 SBI_SPEC_VERSION_MAJOR_MASK

```
#define SBI_SPEC_VERSION_MAJOR_MASK 0x7f
```

30.70.1.151 SBI_SPEC_VERSION_MAJOR_OFFSET

```
#define SBI_SPEC_VERSION_MAJOR_OFFSET 24
```

30.70.1.152 SBI_SPEC_VERSION_MINOR_MASK

```
#define SBI_SPEC_VERSION_MINOR_MASK 0xffffff
```

30.70.1.153 SBI_SRST_RESET_REASON_NONE

```
#define SBI_SRST_RESET_REASON_NONE 0x0
```

30.70.1.154 SBI_SRST_RESET_REASON_SYSFAIL

```
#define SBI_SRST_RESET_REASON_SYSFAIL 0x1
```

30.70.1.155 SBI_SRST_RESET_TYPE_COLD_REBOOT

```
#define SBI_SRST_RESET_TYPE_COLD_REBOOT 0x1
```

30.70.1.156 SBI_SRST_RESET_TYPE_LAST

```
#define SBI_SRST_RESET_TYPE_LAST SBI_SRST_RESET_TYPE_WARM_REBOOT
```

30.70.1.157 SBI_SRST_RESET_TYPE_SHUTDOWN

```
#define SBI_SRST_RESET_TYPE_SHUTDOWN 0x0
```

30.70.1.158 SBI_SRST_RESET_TYPE_WARM_REBOOT

```
#define SBI_SRST_RESET_TYPE_WARM_REBOOT 0x2
```

30.70.1.159 SBI_SSE_ATTR_CONFIG_ONESHOT

```
#define SBI_SSE_ATTR_CONFIG_ONESHOT (1 << 0)
```

30.70.1.160 SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV

```
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV BIT(2)
```

30.70.1.161 SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP

```
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP BIT(3)
```

30.70.1.162 SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE

```
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE BIT(1)
```

30.70.1.163 SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP

```
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP BIT(0)
```

30.70.1.164 SBI_SSE_ATTR_STATUS_INJECT_OFFSET

```
#define SBI_SSE_ATTR_STATUS_INJECT_OFFSET 3
```

30.70.1.165 SBI_SSE_ATTR_STATUS_PENDING_OFFSET

```
#define SBI_SSE_ATTR_STATUS_PENDING_OFFSET 2
```

30.70.1.166 SBI_SSE_ATTR_STATUS_STATE_MASK

```
#define SBI_SSE_ATTR_STATUS_STATE_MASK 0x3
```

30.70.1.167 SBI_SSE_ATTR_STATUS_STATE_OFFSET

```
#define SBI_SSE_ATTR_STATUS_STATE_OFFSET 0
```

30.70.1.168 SBI_SSE_EVENT_GLOBAL_BIT

```
#define SBI_SSE_EVENT_GLOBAL_BIT (1 << 15)
```

30.70.1.169 SBI_SSE_EVENT_GLOBAL_PLAT_0_END

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_0_END 0x0000ffff
```

30.70.1.170 SBI_SSE_EVENT_GLOBAL_PLAT_0_START

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_0_START 0x0000c000
```

30.70.1.171 SBI_SSE_EVENT_GLOBAL_PLAT_1_END

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_1_END 0x0001ffff
```

30.70.1.172 SBI_SSE_EVENT_GLOBAL_PLAT_1_START

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_1_START 0x0001c000
```

30.70.1.173 SBI_SSE_EVENT_GLOBAL_PLAT_2_END

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_2_END 0x0002ffff
```

30.70.1.174 SBI_SSE_EVENT_GLOBAL_PLAT_2_START

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_2_START 0x0002c000
```

30.70.1.175 SBI_SSE_EVENT_GLOBAL_PLAT_3_END

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_3_END 0xffffffff
```

30.70.1.176 SBI_SSE_EVENT_GLOBAL_PLAT_3_START

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_3_START 0xffffc000
```

30.70.1.177 SBI_SSE_EVENT_GLOBAL_RAS

```
#define SBI_SSE_EVENT_GLOBAL_RAS 0x00008000
```

30.70.1.178 SBI_SSE_EVENT_GLOBAL_SOFTWARE

```
#define SBI_SSE_EVENT_GLOBAL_SOFTWARE 0xffff8000
```

30.70.1.179 SBI_SSE_EVENT_LOCAL_DOUBLE_TRAP

```
#define SBI_SSE_EVENT_LOCAL_DOUBLE_TRAP 0x00000001
```

30.70.1.180 SBI_SSE_EVENT_LOCAL_PLAT_0_END

```
#define SBI_SSE_EVENT_LOCAL_PLAT_0_END 0x00007fff
```

30.70.1.181 SBI_SSE_EVENT_LOCAL_PLAT_0_START

```
#define SBI_SSE_EVENT_LOCAL_PLAT_0_START 0x00004000
```

30.70.1.182 SBI_SSE_EVENT_LOCAL_PLAT_1_END

```
#define SBI_SSE_EVENT_LOCAL_PLAT_1_END 0x00017fff
```

30.70.1.183 SBI_SSE_EVENT_LOCAL_PLAT_1_START

```
#define SBI_SSE_EVENT_LOCAL_PLAT_1_START 0x00014000
```

30.70.1.184 SBI_SSE_EVENT_LOCAL_PLAT_2_END

```
#define SBI_SSE_EVENT_LOCAL_PLAT_2_END 0x00027fff
```

30.70.1.185 SBI_SSE_EVENT_LOCAL_PLAT_2_START

```
#define SBI_SSE_EVENT_LOCAL_PLAT_2_START 0x00024000
```

30.70.1.186 SBI_SSE_EVENT_LOCAL_PLAT_3_END

```
#define SBI_SSE_EVENT_LOCAL_PLAT_3_END 0xffff7fff
```

30.70.1.187 SBI_SSE_EVENT_LOCAL_PLAT_3_START

```
#define SBI_SSE_EVENT_LOCAL_PLAT_3_START 0xffff4000
```

30.70.1.188 SBI_SSE_EVENT_LOCAL_PMU

```
#define SBI_SSE_EVENT_LOCAL_PMU 0x00010000
```

30.70.1.189 SBI_SSE_EVENT_LOCAL_RAS

```
#define SBI_SSE_EVENT_LOCAL_RAS 0x00000000
```

30.70.1.190 SBI_SSE_EVENT_LOCAL_SOFTWARE

```
#define SBI_SSE_EVENT_LOCAL_SOFTWARE 0xffff0000
```

30.70.1.191 SBI_SSE_EVENT_PLATFORM_BIT

```
#define SBI_SSE_EVENT_PLATFORM_BIT (1 << 14)
```

30.70.1.192 SBI_SUCCESS

```
#define SBI_SUCCESS 0
```

30.70.1.193 SBI_SUSP_PLATFORM_SLEEP_START

```
#define SBI_SUSP_PLATFORM_SLEEP_START 0x80000000
```

30.70.1.194 SBI_SUSP_SLEEP_TYPE_LAST

```
#define SBI_SUSP_SLEEP_TYPE_LAST SBI_SUSP_SLEEP_TYPE_SUSPEND
```

30.70.1.195 SBI_SUSP_SLEEP_TYPE_SUSPEND

```
#define SBI_SUSP_SLEEP_TYPE_SUSPEND 0x0
```

30.70.2 Enumeration Type Documentation

30.70.2.1 sbi_cppc_reg_id

```
enum sbi_cppc_reg_id
```

Enumerator

SBI_CPPC_HIGHEST_PERF	
SBI_CPPC_NOMINAL_PERF	
SBI_CPPC_LOW_NON_LINEAR_PERF	
SBI_CPPC_LOWEST_PERF	
SBI_CPPC_GUARANTEED_PERF	
SBI_CPPC_DESIRED_PERF	
SBI_CPPC_MIN_PERF	
SBI_CPPC_MAX_PERF	
SBI_CPPC_PERF_REDUC_TOLERANCE	
SBI_CPPC_TIME_WINDOW	
SBI_CPPC_CTR_WRAP_TIME	
SBI_CPPC_REFERENCE_CTR	
SBI_CPPC_DELIVERED_CTR	
SBI_CPPC_PERF_LIMITED	
SBI_CPPC_ENABLE	
SBI_CPPC_AUTO_SEL_ENABLE	
SBI_CPPC_AUTO_ACT_WINDOW	
SBI_CPPC_ENERGY_PERF_PREFERENCE	
SBI_CPPC_REFERENCE_PERF	
SBI_CPPC_LOWEST_FREQ	
SBI_CPPC_NOMINAL_FREQ	

Enumerator

SBI_CPPC_ACPI_LAST	
SBI_CPPC_TRANSITION_LATENCY	
SBI_CPPC_NON_ACPI_LAST	

30.70.2.2 sbi_fwft_feature_t

`enum sbi_fwft_feature_t`

Enumerator

SBI_FWFT_MISALIGNED_EXC_DELEG	
SBI_FWFT_LANDING_PAD	
SBI_FWFT_SHADOW_STACK	
SBI_FWFT_DOUBLE_TRAP	
SBI_FWFT_PTE_AD_HW_UPDATING	
SBI_FWFT_POINTER_MASKING_PMLEN	
SBI_FWFT_LOCAL_RESERVED_START	
SBI_FWFT_LOCAL_RESERVED_END	
SBI_FWFT_LOCAL_PLATFORM_START	
SBI_FWFT_LOCAL_PLATFORM_END	
SBI_FWFT_GLOBAL_RESERVED_START	
SBI_FWFT_GLOBAL_RESERVED_END	
SBI_FWFT_GLOBAL_PLATFORM_START	
SBI_FWFT_GLOBAL_PLATFORM_END	

30.70.2.3 sbi_pmu_ctr_type

`enum sbi_pmu_ctr_type`
SBI PMU counter type

Enumerator

SBI_PMU_CTR_TYPE_HW	
SBI_PMU_CTR_TYPE_FW	

30.70.2.4 sbi_pmu_event_type_id

`enum sbi_pmu_event_type_id`
SBI PMU event idx type

Enumerator

SBI_PMU_EVENT_TYPE_HW	
SBI_PMU_EVENT_TYPE_HW_CACHE	
SBI_PMU_EVENT_TYPE_HW_RAW	
SBI_PMU_EVENT_TYPE_HW_RAW_V2	
SBI_PMU_EVENT_TYPE_FW	
SBI_PMU_EVENT_TYPE_MAX	

30.70.2.5 sbi_pmu_fw_event_code_id

enum sbi_pmu_fw_event_code_id

Special "firmware" events provided by the OpenSBI, even if the hardware does not support performance events. These events are encoded as a raw event type in Linux kernel perf framework.

Enumerator

SBI_PMU_FW_MISALIGNED_LOAD	
SBI_PMU_FW_MISALIGNED_STORE	
SBI_PMU_FW_ACCESS_LOAD	
SBI_PMU_FW_ACCESS_STORE	
SBI_PMU_FW_ILLEGAL_INSN	
SBI_PMU_FW_SET_TIMER	
SBI_PMU_FW_IPI_SENT	
SBI_PMU_FW_IPI_RECVD	
SBI_PMU_FW_FENCE_I_SENT	
SBI_PMU_FW_FENCE_I_RECVD	
SBI_PMU_FW_SFENCE_VMA_SENT	
SBI_PMU_FW_SFENCE_VMA_RCVD	
SBI_PMU_FW_SFENCE_VMA_ASID_SENT	
SBI_PMU_FW_SFENCE_VMA_ASID_RCVD	
SBI_PMU_FW_HFENCE_GVMA_SENT	
SBI_PMU_FW_HFENCE_GVMA_RCVD	
SBI_PMU_FW_HFENCE_GVMA_VMID_SENT	
SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD	
SBI_PMU_FW_HFENCE_VVMA_SENT	
SBI_PMU_FW_HFENCE_VVMA_RCVD	
SBI_PMU_FW_HFENCE_VVMA_ASID_SENT	
SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD	
SBI_PMU_FW_MAX	
SBI_PMU_FW_RESERVED_MAX	
SBI_PMU_FW_PLATFORM	

30.70.2.6 sbi_pmu_hw_cache_id

enum sbi_pmu_hw_cache_id

Generalized hardware cache events:

```
{ L1-D, L1-I, LLC, ITLB, DTLB, BPU, NODE } x
{ read, write, prefetch } x
{ accesses, misses }
```

Enumerator

SBI_PMU_HW_CACHE_L1D	
SBI_PMU_HW_CACHE_L1I	
SBI_PMU_HW_CACHE_LL	
SBI_PMU_HW_CACHE_DTLB	
SBI_PMU_HW_CACHE_ITLB	
SBI_PMU_HW_CACHE_BPU	
SBI_PMU_HW_CACHE_NODE	
SBI_PMU_HW_CACHE_MAX	

30.70.2.7 sbi_pmu_hw_cache_op_id

enum sbi_pmu_hw_cache_op_id

Enumerator

SBI_PMU_HW_CACHE_OP_READ	
SBI_PMU_HW_CACHE_OP_WRITE	
SBI_PMU_HW_CACHE_OP_PREFETCH	
SBI_PMU_HW_CACHE_OP_MAX	

30.70.2.8 sbi_pmu_hw_cache_op_result_id

enum sbi_pmu_hw_cache_op_result_id

Enumerator

SBI_PMU_HW_CACHE_RESULT_ACCESS	
SBI_PMU_HW_CACHE_RESULT_MISS	
SBI_PMU_HW_CACHE_RESULT_MAX	

30.70.2.9 sbi_pmu_hw_generic_events_t

enum sbi_pmu_hw_generic_events_t

General pmu event codes specified in SBI PMU extension

Enumerator

SBI_PMU_HW_NO_EVENT	
SBI_PMU_HW_CPU_CYCLES	
SBI_PMU_HW_INSTRUCTIONS	
SBI_PMU_HW_CACHE_REFERENCES	
SBI_PMU_HW_CACHE_MISSES	
SBI_PMU_HW_BRANCH_INSTRUCTIONS	
SBI_PMU_HW_BRANCH_MISSES	
SBI_PMU_HW_BUS_CYCLES	
SBI_PMU_HW_STALLED_CYCLES_FRONTEND	
SBI_PMU_HW_STALLED_CYCLES_BACKEND	
SBI_PMU_HW_REF_CPU_CYCLES	
SBI_PMU_HW_GENERAL_MAX	

30.70.2.10 sbi_sse_attr_id

enum sbi_sse_attr_id

Enumerator

SBI_SSE_ATTR_STATUS	
SBI_SSE_ATTR_PRIO	
SBI_SSE_ATTR_CONFIG	
SBI_SSE_ATTR_PREFERRED_HART	
SBI_SSE_ATTR_ENTRY_PC	
SBI_SSE_ATTR_ENTRY_ARG	

Enumerator

SBI_SSE_ATTR_INTERRUPTED_SEPC	
SBI_SSE_ATTR_INTERRUPTED_FLAGS	
SBI_SSE_ATTR_INTERRUPTED_A6	
SBI_SSE_ATTR_INTERRUPTED_A7	
SBI_SSE_ATTR_MAX	

30.70.2.11 sbi_sse_state

```
enum sbi_sse_state
```

Enumerator

SBI_SSE_STATE_UNUSED	
SBI_SSE_STATE_REGISTERED	
SBI_SSE_STATE_ENABLED	
SBI_SSE_STATE_RUNNING	

30.71 sbi_ecall_interface.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_ECALL_INTERFACE_H__
#define __SBI_ECALL_INTERFACE_H__

/* clang-format off */

#include <sbi/sbi_types.h>

/* SBI Extension IDs */
#define SBI_EXT_0_1_SET_TIMER           0x0
#define SBI_EXT_0_1_CONSOLE_PUTCHAR     0x1
#define SBI_EXT_0_1_CONSOLE_GETCHAR     0x2
#define SBI_EXT_0_1_CLEAR_IPI           0x3
#define SBI_EXT_0_1_SEND_IPI            0x4
#define SBI_EXT_0_1_REMOTE_FENCE_I      0x5
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA       0x6
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID  0x7
#define SBI_EXT_0_1_SHUTDOWN            0x8
#define SBI_EXT_BASE                0x10
#define SBI_EXT_TIME                0x54494D45
#define SBI_EXT_IPI             0x735049
#define SBI_EXT_RFENCE              0x52464E43
#define SBI_EXT_HSM             0x48534D
#define SBI_EXT_SRST                0x53525354
#define SBI_EXT_PMU             0x504D55
#define SBI_EXT_DBCN                0x4442434E
#define SBI_EXT_SUSP                0x53555350
#define SBI_EXT_CPPC                0x43505043
#define SBI_EXT_DBTR                0x44425452
#define SBI_EXT_SSE             0x535345
#define SBI_EXT_FWFT                0x46574654
#define SBI_EXT_MPXY                0x4D505859

/* SBI function IDs for BASE extension*/
#define SBI_EXT_BASE_GET_SPEC_VERSION       0x0
#define SBI_EXT_BASE_GET_IMP_ID         0x1
#define SBI_EXT_BASE_GET_IMP_VERSION        0x2
#define SBI_EXT_BASE_PROBE_EXT          0x3
#define SBI_EXT_BASE_GET_MVENDORID      0x4
#define SBI_EXT_BASE_GET_MARCHID        0x5
#define SBI_EXT_BASE_GET_MIMPID         0x6
```

```
/* SBI function IDs for TIME extension*/
#define SBI_EXT_TIME_SET_TIMER          0x0

/* SBI function IDs for IPI extension*/
#define SBI_EXT_IPI_SEND_IPI            0x0

/* SBI function IDs for RFENCE extension*/
#define SBI_EXT_RFENCE_REMOTE_FENCE_I       0x0
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA    0x1
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID   0x2
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID  0x3
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA   0x4
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID  0x5
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA   0x6

/* SBI function IDs for HSM extension */
#define SBI_EXT_HSM_HART_START          0x0
#define SBI_EXT_HSM_HART_STOP           0x1
#define SBI_EXT_HSM_HART_GET_STATUS     0x2
#define SBI_EXT_HSM_HART_SUSPEND        0x3

#define SBI_HSM_STATE_STARTED           0x0
#define SBI_HSM_STATE_STOPPED           0x1
#define SBI_HSM_STATE_START_PENDING     0x2
#define SBI_HSM_STATE_STOP_PENDING      0x3
#define SBI_HSM_STATE_SUSPENDED         0x4
#define SBI_HSM_STATE_SUSPEND_PENDING       0x5
#define SBI_HSM_STATE_RESUME_PENDING        0x6

#define SBI_HSM_SUSP_BASE_MASK          0x7fffffff
#define SBI_HSM_SUSP_NON_RET_BIT        0x80000000
#define SBI_HSM_SUSP_PLAT_BASE          0x10000000

#define SBI_HSM_SUSPEND_RET_DEFAULT     0x00000000
#define SBI_HSM_SUSPEND_RET_PLATFORM        SBI_HSM_SUSP_PLAT_BASE
#define SBI_HSM_SUSPEND_RET_LAST        SBI_HSM_SUSP_BASE_MASK
#define SBI_HSM_SUSPEND_NON_RET_DEFAULT     SBI_HSM_SUSP_NON_RET_BIT
#define SBI_HSM_SUSPEND_NON_RET_PLATFORM    (SBI_HSM_SUSP_NON_RET_BIT | \
                          SBI_HSM_SUSP_PLAT_BASE)
#define SBI_HSM_SUSPEND_NON_RET_LAST        (SBI_HSM_SUSP_NON_RET_BIT | \
                          SBI_HSM_SUSP_BASE_MASK)

/* SBI function IDs for SRST extension */
#define SBI_EXT_SRST_RESET          0x0

#define SBI_SRST_RESET_TYPE_SHUTDOWN        0x0
#define SBI_SRST_RESET_TYPE_COLD_REBOOT 0x1
#define SBI_SRST_RESET_TYPE_WARM_REBOOT 0x2
#define SBI_SRST_RESET_TYPE_LAST    SBI_SRST_RESET_TYPE_WARM_REBOOT

#define SBI_SRST_RESET_REASON_NONE  0x0
#define SBI_SRST_RESET_REASON_SYSFAIL   0x1

/* SBI function IDs for PMU extension */
#define SBI_EXT_PMU_NUM_COUNTERS    0x0
#define SBI_EXT_PMU_COUNTER_GET_INFO    0x1
#define SBI_EXT_PMU_COUNTER_CFG_MATCH   0x2
#define SBI_EXT_PMU_COUNTER_START   0x3
#define SBI_EXT_PMU_COUNTER_STOP    0x4
#define SBI_EXT_PMU_COUNTER_FW_READ 0x5
#define SBI_EXT_PMU_COUNTER_FW_READ_HI  0x6
#define SBI_EXT_PMU_SNAPSHOT_SET_SHMEM  0x7
#define SBI_EXT_PMU_EVENT_GET_INFO      0x8

/* SBI function IDs for DBTR extension */
#define SBI_EXT_DBTR_NUM_TRIGGERS   0x0
#define SBI_EXT_DBTR_SETUP_SHMEM    0x1
#define SBI_EXT_DBTR_TRIGGER_READ   0x2
#define SBI_EXT_DBTR_TRIGGER_INSTALL    0x3
#define SBI_EXT_DBTR_TRIGGER_UPDATE 0x4
#define SBI_EXT_DBTR_TRIGGER_UNINSTALL  0x5
#define SBI_EXT_DBTR_TRIGGER_ENABLE 0x6
#define SBI_EXT_DBTR_TRIGGER_DISABLE    0x7

/* SBI function IDs for FW feature extension */
#define SBI_EXT_FWFT_SET        0x0
#define SBI_EXT_FWFT_GET        0x1

enum sbi_fwft_feature_t {
    SBI_FWFT_MISALIGNED_EXC_DELEG      = 0x0,
    SBI_FWFT_LANDING_PAD            = 0x1,
    SBI_FWFT_SHADOW_STACK          = 0x2,
    SBI_FWFT_DOUBLE_TRAP            = 0x3,
    SBI_FWFT_PTE_AD_HW_UPDATING      = 0x4,
    SBI_FWFT_POINTER_MASKING_PMLEN        = 0x5,
    SBI_FWFT_LOCAL_RESERVED_START      = 0x6,
```

```
    SBI_FWFT_LOCAL_RESERVED_END      = 0x3fffffff,
    SBI_FWFT_LOCAL_PLATFORM_START      = 0x40000000,
    SBI_FWFT_LOCAL_PLATFORM_END      = 0x7fffffff,

    SBI_FWFT_GLOBAL_RESERVED_START        = 0x80000000,
    SBI_FWFT_GLOBAL_RESERVED_END        = 0xbfffffff,
    SBI_FWFT_GLOBAL_PLATFORM_START        = 0xc0000000,
    SBI_FWFT_GLOBAL_PLATFORM_END        = 0xffffffff,
};

#define SBI_FWFT_GLOBAL_FEATURE_BIT     (1 « 31)
#define SBI_FWFT_PLATFORM_FEATURE_BIT       (1 « 30)

#define SBI_FWFT_SET_FLAG_LOCK          (1 « 0)

enum sbi_pmu_hw_generic_events_t {
    SBI_PMU_HW_NO_EVENT          = 0,
    SBI_PMU_HW_CPU_CYCLES          = 1,
    SBI_PMU_HW_INSTRUCTIONS          = 2,
    SBI_PMU_HW_CACHE_REFERENCES      = 3,
    SBI_PMU_HW_CACHE_MISSES          = 4,
    SBI_PMU_HW_BRANCH_INSTRUCTIONS        = 5,
    SBI_PMU_HW_BRANCH_MISSES        = 6,
    SBI_PMU_HW_BUS_CYCLES          = 7,
    SBI_PMU_HW_STALLED_CYCLES_FRONTEND    = 8,
    SBI_PMU_HW_STALLED_CYCLES_BACKEND  = 9,
    SBI_PMU_HW_REF_CPU_CYCLES      = 10,

    SBI_PMU_HW_GENERAL_MAX,
};

enum sbi_pmu_hw_cache_id {
    SBI_PMU_HW_CACHE_L1D        = 0,
    SBI_PMU_HW_CACHE_L1I        = 1,
    SBI_PMU_HW_CACHE_LL      = 2,
    SBI_PMU_HW_CACHE_DTLB      = 3,
    SBI_PMU_HW_CACHE_ITLB      = 4,
    SBI_PMU_HW_CACHE_BPU        = 5,
    SBI_PMU_HW_CACHE_NODE      = 6,

    SBI_PMU_HW_CACHE_MAX,
};

enum sbi_pmu_hw_cache_op_id {
    SBI_PMU_HW_CACHE_OP_READ    = 0,
    SBI_PMU_HW_CACHE_OP_WRITE  = 1,
    SBI_PMU_HW_CACHE_OP_PREFETCH    = 2,

    SBI_PMU_HW_CACHE_OP_MAX,
};

enum sbi_pmu_hw_cache_op_result_id {
    SBI_PMU_HW_CACHE_RESULT_ACCESS    = 0,
    SBI_PMU_HW_CACHE_RESULT_MISS    = 1,

    SBI_PMU_HW_CACHE_RESULT_MAX,
};

enum sbi_pmu_fw_event_code_id {
    SBI_PMU_FW_MISALIGNED_LOAD    = 0,
    SBI_PMU_FW_MISALIGNED_STORE  = 1,
    SBI_PMU_FW_ACCESS_LOAD        = 2,
    SBI_PMU_FW_ACCESS_STORE      = 3,
    SBI_PMU_FW_ILLEGAL_INSN      = 4,
    SBI_PMU_FW_SET_TIMER        = 5,
    SBI_PMU_FW_IPI_SENT      = 6,
    SBI_PMU_FW_IPI_RECVD        = 7,
    SBI_PMU_FW_FENCE_I_SENT      = 8,
    SBI_PMU_FW_FENCE_I_RECVD    = 9,
    SBI_PMU_FW_SFENCE_VMA_SENT    = 10,
    SBI_PMU_FW_SFENCE_VMA_RCVD    = 11,
    SBI_PMU_FW_SFENCE_VMA_ASID_SENT  = 12,
    SBI_PMU_FW_SFENCE_VMA_ASID_RCVD  = 13,

    SBI_PMU_FW_HFENCE_GVMA_SENT  = 14,
    SBI_PMU_FW_HFENCE_GVMA_RCVD  = 15,
    SBI_PMU_FW_HFENCE_GVMA_VMID_SENT = 16,
    SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD = 17,

    SBI_PMU_FW_HFENCE_VVMA_SENT  = 18,
    SBI_PMU_FW_HFENCE_VVMA_RCVD  = 19,
    SBI_PMU_FW_HFENCE_VVMA_ASID_SENT = 20,
    SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD = 21,
    SBI_PMU_FW_MAX,
    /*
     * Event codes 22 to 255 are reserved for future use.
     * Event codes 256 to 65534 are reserved for SBI implementation
```

```
     * specific custom firmware events.
     */
    SBI_PMU_FW_RESERVED_MAX = 0xFFFE,
    /*
     * Event code 0xFFFF is used for platform specific firmware
     * events where the event data contains any event specific information.
     */
    SBI_PMU_FW_PLATFORM = 0xFFFF,
};

enum sbi_pmu_event_type_id {
    SBI_PMU_EVENT_TYPE_HW              = 0x0,
    SBI_PMU_EVENT_TYPE_HW_CACHE          = 0x1,
    SBI_PMU_EVENT_TYPE_HW_RAW          = 0x2,
    SBI_PMU_EVENT_TYPE_HW_RAW_V2             = 0x3,
    SBI_PMU_EVENT_TYPE_FW              = 0xf,
    SBI_PMU_EVENT_TYPE_MAX,
};

enum sbi_pmu_ctr_type {
    SBI_PMU_CTR_TYPE_HW = 0,
    SBI_PMU_CTR_TYPE_FW,
};

struct sbi_pmu_event_info {
    uint32_t event_idx;
    uint32_t output;
    uint64_t event_data;
};

/* Helper macros to decode event idx */
#define SBI_PMU_EVENT_IDX_MASK 0xFFFFF
#define SBI_PMU_EVENT_IDX_TYPE_OFFSET 16
#define SBI_PMU_EVENT_IDX_TYPE_MASK (0xF « SBI_PMU_EVENT_IDX_TYPE_OFFSET)
#define SBI_PMU_EVENT_IDX_CODE_MASK 0xFFFF
#define SBI_PMU_EVENT_RAW_IDX 0x20000
#define SBI_PMU_EVENT_RAW_V2_IDX 0x30000

#define SBI_PMU_EVENT_IDX_INVALID 0xFFFFFFFF

#define SBI_PMU_EVENT_HW_CACHE_OPS_RESULT   0x1
#define SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK  0x6
#define SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET    1
#define SBI_PMU_EVENT_HW_CACHE_ID_MASK      0xfff8
#define SBI_PMU_EVENT_HW_CACHE_ID_OFFSET    3

/* Flags defined for config matching function */
#define SBI_PMU_CFG_FLAG_SKIP_MATCH (1 « 0)
#define SBI_PMU_CFG_FLAG_CLEAR_VALUE    (1 « 1)
#define SBI_PMU_CFG_FLAG_AUTO_START (1 « 2)
#define SBI_PMU_CFG_FLAG_SET_VUINH  (1 « 3)
#define SBI_PMU_CFG_FLAG_SET_VSINH  (1 « 4)
#define SBI_PMU_CFG_FLAG_SET_UINH   (1 « 5)
#define SBI_PMU_CFG_FLAG_SET_SINH   (1 « 6)
#define SBI_PMU_CFG_FLAG_SET_MINH   (1 « 7)

/* Flags defined for counter start function */
#define SBI_PMU_START_FLAG_SET_INIT_VALUE (1 « 0)
#define SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT (1 « 1)

/* Flags defined for counter stop function */
#define SBI_PMU_STOP_FLAG_RESET (1 « 0)
#define SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT (1 « 1)

/* SBI function IDs for DBCN extension */
#define SBI_EXT_DBCN_CONSOLE_WRITE      0x0
#define SBI_EXT_DBCN_CONSOLE_READ       0x1
#define SBI_EXT_DBCN_CONSOLE_WRITE_BYTE     0x2

/* SBI function IDs for SUSP extension */
#define SBI_EXT_SUSP_SUSPEND            0x0

#define SBI_SUSP_SLEEP_TYPE_SUSPEND     0x0
#define SBI_SUSP_SLEEP_TYPE_LAST        SBI_SUSP_SLEEP_TYPE_SUSPEND
#define SBI_SUSP_PLATFORM_SLEEP_START       0x80000000

/* SBI function IDs for CPPC extension */
#define SBI_EXT_CPPC_PROBE          0x0
#define SBI_EXT_CPPC_READ           0x1
#define SBI_EXT_CPPC_READ_HI            0x2
#define SBI_EXT_CPPC_WRITE          0x3

enum sbi_cppc_reg_id {
    SBI_CPPC_HIGHEST_PERF      = 0x00000000,
    SBI_CPPC_NOMINAL_PERF      = 0x00000001,
    SBI_CPPC_LOW_NON_LINEAR_PERF    = 0x00000002,
    SBI_CPPC_LOWEST_PERF        = 0x00000003,
```

```
    SBI_CPPC_GUARANTEED_PERF    = 0x00000004,
    SBI_CPPC_DESIRED_PERF      = 0x00000005,
    SBI_CPPC_MIN_PERF      = 0x00000006,
    SBI_CPPC_MAX_PERF      = 0x00000007,
    SBI_CPPC_PERF_REDUC_TOLERANCE  = 0x00000008,
    SBI_CPPC_TIME_WINDOW        = 0x00000009,
    SBI_CPPC_CTR_WRAP_TIME        = 0x0000000A,
    SBI_CPPC_REFERENCE_CTR        = 0x0000000B,
    SBI_CPPC_DELIVERED_CTR        = 0x0000000C,
    SBI_CPPC_PERF_LIMITED      = 0x0000000D,
    SBI_CPPC_ENABLE          = 0x0000000E,
    SBI_CPPC_AUTO_SEL_ENABLE    = 0x0000000F,
    SBI_CPPC_AUTO_ACT_WINDOW    = 0x00000010,
    SBI_CPPC_ENERGY_PERF_PREFERENCE  = 0x00000011,
    SBI_CPPC_REFERENCE_PERF      = 0x00000012,
    SBI_CPPC_LOWEST_FREQ        = 0x00000013,
    SBI_CPPC_NOMINAL_FREQ      = 0x00000014,
    SBI_CPPC_ACPI_LAST        = SBI_CPPC_NOMINAL_FREQ,
    SBI_CPPC_TRANSITION_LATENCY  = 0x80000000,
    SBI_CPPC_NON_ACPI_LAST        = SBI_CPPC_TRANSITION_LATENCY,
};

/* SBI Function IDs for SSE extension */
#define SBI_EXT_SSE_READ_ATTR       0x00000000
#define SBI_EXT_SSE_WRITE_ATTR      0x00000001
#define SBI_EXT_SSE_REGISTER        0x00000002
#define SBI_EXT_SSE_UNREGISTER      0x00000003
#define SBI_EXT_SSE_ENABLE      0x00000004
#define SBI_EXT_SSE_DISABLE     0x00000005
#define SBI_EXT_SSE_COMPLETE        0x00000006
#define SBI_EXT_SSE_INJECT      0x00000007
#define SBI_EXT_SSE_HART_UNMASK     0x00000008
#define SBI_EXT_SSE_HART_MASK       0x00000009

/* SBI SSE Event Attributes. */
enum sbi_sse_attr_id {
    SBI_SSE_ATTR_STATUS      = 0x00000000,
    SBI_SSE_ATTR_PRIO      = 0x00000001,
    SBI_SSE_ATTR_CONFIG      = 0x00000002,
    SBI_SSE_ATTR_PREFERRED_HART  = 0x00000003,
    SBI_SSE_ATTR_ENTRY_PC      = 0x00000004,
    SBI_SSE_ATTR_ENTRY_ARG        = 0x00000005,
    SBI_SSE_ATTR_INTERRUPTED_SEPC  = 0x00000006,
    SBI_SSE_ATTR_INTERRUPTED_FLAGS    = 0x00000007,
    SBI_SSE_ATTR_INTERRUPTED_A6  = 0x00000008,
    SBI_SSE_ATTR_INTERRUPTED_A7  = 0x00000009,

    SBI_SSE_ATTR_MAX        = 0x0000000A
};

#define SBI_SSE_ATTR_STATUS_STATE_OFFSET    0
#define SBI_SSE_ATTR_STATUS_STATE_MASK      0x3
#define SBI_SSE_ATTR_STATUS_PENDING_OFFSET  2
#define SBI_SSE_ATTR_STATUS_INJECT_OFFSET   3

#define SBI_SSE_ATTR_CONFIG_ONESHOT (1 << 0)

#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP   BIT(0)
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE  BIT(1)
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV  BIT(2)
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP BIT(3)

enum sbi_sse_state {
    SBI_SSE_STATE_UNUSED        = 0,
    SBI_SSE_STATE_REGISTERED    = 1,
    SBI_SSE_STATE_ENABLED      = 2,
    SBI_SSE_STATE_RUNNING      = 3,
};

/* SBI SSE Event IDs. */
#define SBI_SSE_EVENT_LOCAL_RAS         0x00000000
#define SBI_SSE_EVENT_LOCAL_DOUBLE_TRAP     0x00000001
#define SBI_SSE_EVENT_LOCAL_PLAT_0_START    0x00004000
#define SBI_SSE_EVENT_LOCAL_PLAT_0_END      0x00007fff
#define SBI_SSE_EVENT_GLOBAL_RAS        0x00008000
#define SBI_SSE_EVENT_GLOBAL_PLAT_0_START   0x0000c000
#define SBI_SSE_EVENT_GLOBAL_PLAT_0_END     0x0000ffff

#define SBI_SSE_EVENT_LOCAL_PMU         0x00010000
#define SBI_SSE_EVENT_LOCAL_PLAT_1_START    0x00014000
#define SBI_SSE_EVENT_LOCAL_PLAT_1_END      0x00017fff
#define SBI_SSE_EVENT_GLOBAL_PLAT_1_START   0x0001c000
#define SBI_SSE_EVENT_GLOBAL_PLAT_1_END     0x0001ffff

#define SBI_SSE_EVENT_LOCAL_PLAT_2_START    0x00024000
#define SBI_SSE_EVENT_LOCAL_PLAT_2_END      0x00027fff
#define SBI_SSE_EVENT_GLOBAL_PLAT_2_START   0x0002c000
```

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_2_END     0x0002ffff

#define SBI_SSE_EVENT_LOCAL_SOFTWARE        0xffff0000
#define SBI_SSE_EVENT_LOCAL_PLAT_3_START    0xffff4000
#define SBI_SSE_EVENT_LOCAL_PLAT_3_END      0xffff7fff
#define SBI_SSE_EVENT_GLOBAL_SOFTWARE       0xffff8000
#define SBI_SSE_EVENT_GLOBAL_PLAT_3_START   0xffffc000
#define SBI_SSE_EVENT_GLOBAL_PLAT_3_END     0xffffffff

#define SBI_SSE_EVENT_GLOBAL_BIT        (1 « 15)
#define SBI_SSE_EVENT_PLATFORM_BIT      (1 « 14)

/* SBI function IDs for MPXY extension */
#define SBI_EXT_MPXY_SET_SHMEM          0x0
#define SBI_EXT_MPXY_GET_CHANNEL_IDS        0x1
#define SBI_EXT_MPXY_READ_ATTRS         0x2
#define SBI_EXT_MPXY_WRITE_ATTRS        0x3
#define SBI_EXT_MPXY_SEND_MSG_WITH_RESP     0x4
#define SBI_EXT_MPXY_SEND_MSG_NO_RESP       0x5
#define SBI_EXT_MPXY_GET_NOTIFICATION_EVENTS    0x6

/* SBI base specification related macros */
#define SBI_SPEC_VERSION_MAJOR_OFFSET       24
#define SBI_SPEC_VERSION_MAJOR_MASK     0x7f
#define SBI_SPEC_VERSION_MINOR_MASK     0xffffff
#define SBI_EXT_VENDOR_START            0x09000000
#define SBI_EXT_VENDOR_END          0x09FFFFFF
#define SBI_EXT_FIRMWARE_START          0x0A000000
#define SBI_EXT_FIRMWARE_END            0x0AFFFFFF

/* SBI return error codes */
#define SBI_SUCCESS             0
#define SBI_ERR_FAILED              -1
#define SBI_ERR_NOT_SUPPORTED           -2
#define SBI_ERR_INVALID_PARAM           -3
#define SBI_ERR_DENIED              -4
#define SBI_ERR_INVALID_ADDRESS         -5
#define SBI_ERR_ALREADY_AVAILABLE       -6
#define SBI_ERR_ALREADY_STARTED         -7
#define SBI_ERR_ALREADY_STOPPED         -8
#define SBI_ERR_NO_SHMEM            -9
#define SBI_ERR_INVALID_STATE           -10
#define SBI_ERR_BAD_RANGE           -11
#define SBI_ERR_TIMEOUT             -12
#define SBI_ERR_IO              -13

#define SBI_LAST_ERR                SBI_ERR_BAD_RANGE

/* clang-format on */

#endif
```

30.72 include/sbi/sbi_emulate_csr.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_emulate_csr.h:

This graph shows which files directly or indirectly include this file:

Functions

- int sbi_emulate_csr_read (int csr_num, struct sbi_trap_regs *regs, ulong *csr_val)
- int sbi_emulate_csr_write (int csr_num, struct sbi_trap_regs *regs, ulong csr_val)

30.72.1 Function Documentation

30.72.1.1 sbi_emulate_csr_read()

```
int sbi_emulate_csr_read (
            int csr_num,
            struct sbi_trap_regs * regs,
            ulong * csr_val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.72.1.2 sbi_emulate_csr_write()

```
int sbi_emulate_csr_write (
            int csr_num,
            struct sbi_trap_regs * regs,
            ulong csr_val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.73 sbi_emulate_csr.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2019 Western Digital Corporation or its affiliates.
00005  *
00006  * Authors:
00007  *   Anup Patel <anup.patel@wdc.com>
00008  */
00009
00010 #ifndef __SBI_EMULATE_CSR_H__
```

```
#define __SBI_EMULATE_CSR_H__

#include <sbi/sbi_types.h>

struct sbi_trap_regs;

int sbi_emulate_csr_read(int csr_num, struct sbi_trap_regs *regs,
             ulong *csr_val);

int sbi_emulate_csr_write(int csr_num, struct sbi_trap_regs *regs,
              ulong csr_val);

#endif
```

30.74 include/sbi/sbi_error.h File Reference

`#include <sbi/sbi_ecall_interface.h>`

Include dependency graph for sbi_error.h:

Macros

- #define SBI_OK 0
- #define SBI_EFAIL SBI_ERR_FAILED
- #define SBI_ENOTSUPP SBI_ERR_NOT_SUPPORTED
- #define SBI_EINVAL SBI_ERR_INVALID_PARAM
- #define SBI_EDENIED SBI_ERR_DENIED
- #define SBI_EINVALID_ADDR SBI_ERR_INVALID_ADDRESS
- #define SBI_EALREADY SBI_ERR_ALREADY_AVAILABLE
- #define SBI_EALREADY_STARTED SBI_ERR_ALREADY_STARTED
- #define SBI_EALREADY_STOPPED SBI_ERR_ALREADY_STOPPED
- #define SBI_ENO_SHMEM SBI_ERR_NO_SHMEM
- #define SBI_EINVALID_STATE SBI_ERR_INVALID_STATE
- #define SBI_EBAD_RANGE SBI_ERR_BAD_RANGE
- #define SBI_ETIMEOUT SBI_ERR_TIMEOUT
- #define SBI_ETIMEDOUT SBI_ERR_TIMEOUT
- #define SBI_EIO SBI_ERR_IO
- #define SBI_ENODEV -1000
- #define SBI_ENOSYS -1001
- #define SBI_EILL -1002
- #define SBI_ENOSPC -1003
- #define SBI_ENOMEM -1004

- #define SBI_EUNKNOWN -1005
- #define SBI_ENOENT -1006

30.74.1 Macro Definition Documentation

30.74.1.1 SBI_EALREADY

```
#define SBI_EALREADY SBI_ERR_ALREADY_AVAILABLE
```

30.74.1.2 SBI_EALREADY_STARTED

```
#define SBI_EALREADY_STARTED SBI_ERR_ALREADY_STARTED
```

30.74.1.3 SBI_EALREADY_STOPPED

```
#define SBI_EALREADY_STOPPED SBI_ERR_ALREADY_STOPPED
```

30.74.1.4 SBI_EBAD_RANGE

```
#define SBI_EBAD_RANGE SBI_ERR_BAD_RANGE
```

30.74.1.5 SBI_EDENIED

```
#define SBI_EDENIED SBI_ERR_DENIED
```

30.74.1.6 SBI_EFAIL

```
#define SBI_EFAIL SBI_ERR_FAILED
```

30.74.1.7 SBI_EILL

```
#define SBI_EILL -1002
```

30.74.1.8 SBI_EINVAL

```
#define SBI_EINVAL SBI_ERR_INVALID_PARAM
```

30.74.1.9 SBI_EINVALID_ADDR

```
#define SBI_EINVALID_ADDR SBI_ERR_INVALID_ADDRESS
```

30.74.1.10 SBI_EINVALID_STATE

```
#define SBI_EINVALID_STATE SBI_ERR_INVALID_STATE
```

30.74.1.11 SBI_EIO

```
#define SBI_EIO SBI_ERR_IO
```

30.74.1.12 SBI_ENO_SHMEM

```
#define SBI_ENO_SHMEM SBI_ERR_NO_SHMEM
```

30.74.1.13 SBI_ENODEV

```
#define SBI_ENODEV -1000
```

30.74.1.14 SBI_ENOENT

```
#define SBI_ENOENT -1006
```

30.74.1.15 SBI_ENOMEM

```
#define SBI_ENOMEM -1004
```

30.74.1.16 SBI_ENOSPC

```
#define SBI_ENOSPC -1003
```

30.74.1.17 SBI_ENOSYS

```
#define SBI_ENOSYS -1001
```

30.74.1.18 SBI_ENOTSUPP

```
#define SBI_ENOTSUPP SBI_ERR_NOT_SUPPORTED
```

30.74.1.19 SBI_ETIMEDOUT

```
#define SBI_ETIMEDOUT SBI_ERR_TIMEOUT
```

30.74.1.20 SBI_ETIMEOUT

```
#define SBI_ETIMEOUT SBI_ERR_TIMEOUT
```

30.74.1.21 SBI_EUNKNOWN

```
#define SBI_EUNKNOWN -1005
```

30.74.1.22 SBI_OK

```
#define SBI_OK 0
```

30.75 sbi_error.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_ERROR_H__
#define __SBI_ERROR_H__

#include <sbi/sbi_ecall_interface.h>

/* clang-format off */

#define SBI_OK          0
#define SBI_EFAIL       SBI_ERR_FAILED
#define SBI_ENOTSUPP        SBI_ERR_NOT_SUPPORTED
#define SBI_EINVAL      SBI_ERR_INVALID_PARAM
#define SBI_EDENIED     SBI_ERR_DENIED
#define SBI_EINVALID_ADDR   SBI_ERR_INVALID_ADDRESS
#define SBI_EALREADY        SBI_ERR_ALREADY_AVAILABLE
#define SBI_EALREADY_STARTED    SBI_ERR_ALREADY_STARTED
#define SBI_EALREADY_STOPPED    SBI_ERR_ALREADY_STOPPED
#define SBI_ENO_SHMEM       SBI_ERR_NO_SHMEM
#define SBI_EINVALID_STATE  SBI_ERR_INVALID_STATE
#define SBI_EBAD_RANGE      SBI_ERR_BAD_RANGE
#define SBI_ETIMEOUT        SBI_ERR_TIMEOUT
#define SBI_ETIMEDOUT       SBI_ERR_TIMEOUT
#define SBI_EIO         SBI_ERR_IO

#define SBI_ENODEV      -1000
#define SBI_ENOSYS      -1001
#define SBI_EILL        -1002
#define SBI_ENOSPC      -1003
```

```
#define SBI_ENOMEM      -1004
#define SBI_EUNKNOWN        -1005
#define SBI_ENOENT      -1006

/* clang-format on */

#endif
```

30.76 include/sbi/sbi_fifo.h File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_fifo.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_fifo

Macros

- #define SBI_FIFO_INITIALIZER(__queue_mem, __entries, __entry_size)
- #define SBI_FIFO_DEFINE(__name, __queue_mem, __entries, __entry_size) struct sbi_fifo __name = SBI_FIFO_INITIALIZER(__queue_mem, __entries, __entry_size)

Enumerations

- enum sbi_fifo_inplace_update_types { SBI_FIFO_SKIP , SBI_FIFO_UPDATED , SBI_FIFO_UNCHANGED }

Functions

- int sbi_fifo_dequeue (struct sbi_fifo *fifo, void *data)
- int sbi_fifo_enqueue (struct sbi_fifo *fifo, void *data, bool force)
- void sbi_fifo_init (struct sbi_fifo *fifo, void *queue_mem, u16 entries, u16 entry_size)
- int sbi_fifo_is_empty (struct sbi_fifo *fifo)
- int sbi_fifo_is_full (struct sbi_fifo *fifo)
- int sbi_fifo_inplace_update (struct sbi_fifo *fifo, void *in, int(*fptr)(void *in, void *data))
- u16 sbi_fifo_avail (struct sbi_fifo *fifo)

30.76.1 Macro Definition Documentation

30.76.1.1 SBI_FIFO_DEFINE

```
#define SBI_FIFO_DEFINE(
            __name,
            __queue_mem,
            __entries,
            __entry_size ) struct sbi_fifo __name = SBI_FIFO_INITIALIZER(__queue_mem, __↩
entries, __entry_size)
```

30.76.1.2 SBI_FIFO_INITIALIZER

```
#define SBI_FIFO_INITIALIZER(
            __queue_mem,
            __entries,
            __entry_size )
```

Value:

```
{   .queue = __queue_mem,                       \
    .qlock = SPIN_LOCK_INITIALIZER,                     \
    .num_entries = __entries,                   \
    .entry_size = __entry_size,                 \
    .avail = 0,                         \
    .tail = 0,                          \
}
```

30.76.2 Enumeration Type Documentation

30.76.2.1 sbi_fifo_inplace_update_types

```
enum sbi_fifo_inplace_update_types
```

Enumerator

SBI_FIFO_SKIP	
SBI_FIFO_UPDATED	
SBI_FIFO_UNCHANGED	

30.76.3 Function Documentation

30.76.3.1 sbi_fifo_avail()

```
u16 sbi_fifo_avail (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

30.76.3.2 sbi_fifo_dequeue()

```
int sbi_fifo_dequeue (
            struct sbi_fifo * fifo,
            void * data )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.76.3.3 sbi_fifo_enqueue()

```
int sbi_fifo_enqueue (
            struct sbi_fifo * fifo,
            void * data,
            bool force )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.76.3.4 sbi_fifo_init()

```
void sbi_fifo_init (
            struct sbi_fifo * fifo,
            void * queue_mem,
            u16 entries,
            u16 entry_size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.76.3.5 sbi_fifo_inplace_update()

```
int sbi_fifo_inplace_update (
            struct sbi_fifo * fifo,
            void * in,
            int(*)(void *in, void *data) fptr )
```

Provide a helper function to do inplace update to the fifo. Note: The callback function is called with lock being held.

Do not invoke any other fifo function from callback. Otherwise, it will lead to deadlock. Here is the call graph for this function:

Here is the caller graph for this function:

30.76.3.6 sbi_fifo_is_empty()

```
int sbi_fifo_is_empty (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

30.76.3.7 sbi_fifo_is_full()

```
int sbi_fifo_is_full (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

30.77 sbi_fifo.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Atish Patra<atish.patra@wdc.com>
 *
 */

#ifndef __SBI_FIFO_H__
#define __SBI_FIFO_H__

#include <sbi/riscv_locks.h>
#include <sbi/sbi_types.h>

struct sbi_fifo {
    void *queue;
    spinlock_t qlock;
    u16 entry_size;
    u16 num_entries;
    u16 avail;
    u16 tail;
};

#define SBI_FIFO_INITIALIZER(__queue_mem, __entries, __entry_size)  \
{   .queue = __queue_mem,                                           \
    .qlock = SPIN_LOCK_INITIALIZER,                                 \
    .num_entries = __entries,                                       \
    .entry_size = __entry_size,                                     \
    .avail = 0,                                                     \
    .tail = 0,                                                      \
}

#define SBI_FIFO_DEFINE(__name, __queue_mem, __entries, __entry_size)   \
struct sbi_fifo __name = SBI_FIFO_INITIALIZER(__queue_mem, __entries, __entry_size)

enum sbi_fifo_inplace_update_types {
    SBI_FIFO_SKIP,
    SBI_FIFO_UPDATED,
    SBI_FIFO_UNCHANGED,
};

int sbi_fifo_dequeue(struct sbi_fifo *fifo, void *data);
int sbi_fifo_enqueue(struct sbi_fifo *fifo, void *data, bool force);
void sbi_fifo_init(struct sbi_fifo *fifo, void *queue_mem, u16 entries,
           u16 entry_size);
int sbi_fifo_is_empty(struct sbi_fifo *fifo);
int sbi_fifo_is_full(struct sbi_fifo *fifo);
int sbi_fifo_inplace_update(struct sbi_fifo *fifo, void *in,
                int (*fptr)(void *in, void *data));
u16 sbi_fifo_avail(struct sbi_fifo *fifo);

#endif
```

30.78 include/sbi/sbi_fwft.h File Reference

```
#include <sbi/sbi_ecall_interface.h>
```

Include dependency graph for sbi_fwft.h:

This graph shows which files directly or indirectly include this file:

Functions

- int sbi_fwft_set (enum sbi_fwft_feature_t feature, unsigned long value, unsigned long flags)
- int sbi_fwft_get (enum sbi_fwft_feature_t feature, unsigned long *out_val)
- int sbi_fwft_init (struct sbi_scratch *scratch, bool cold_boot)

30.78.1 Function Documentation

30.78.1.1 sbi_fwft_get()

```
int sbi_fwft_get (
            enum sbi_fwft_feature_t feature,
            unsigned long * out_val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.78.1.2 sbi_fwft_init()

```
int sbi_fwft_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.78.1.3 sbi_fwft_set()

```
int sbi_fwft_set (
            enum sbi_fwft_feature_t feature,
```

```
            unsigned long value,
            unsigned long flags )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.79 sbi_fwft.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2024 Rivos Inc.
00005  *
00006  * Authors:
00007  *   Clément Léger <cleger@rivosinc.com>
00008  */
00009
00010 #ifndef __SBI_FW_FEATURE_H__
00011 #define __SBI_FW_FEATURE_H__
00012
00013 #include <sbi/sbi_ecall_interface.h>
00014
00015 struct sbi_scratch;
00016
00017 int sbi_fwft_set(enum sbi_fwft_feature_t feature, unsigned long value,
00018          unsigned long flags);
00019 int sbi_fwft_get(enum sbi_fwft_feature_t feature, unsigned long *out_val);
00020
00021 int sbi_fwft_init(struct sbi_scratch *scratch, bool cold_boot);
00022
00023 #endif
```

30.80 include/sbi/sbi_hart.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_bitops.h>
```

Include dependency graph for sbi_hart.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_hart_ext_data
- struct sbi_hart_features

Macros

- #define SBI_SMEPMP_RESV_ENTRY 0

Enumerations

- enum sbi_hart_priv_versions { SBI_HART_PRIV_VER_UNKNOWN = 0 , SBI_HART_PRIV_VER_1_10 = 1 , SBI_HART_PRIV_VER_1_11 = 2 , SBI_HART_PRIV_VER_1_12 = 3 }
- enum sbi_hart_extensions { SBI_HART_EXT_SMAIA = 0 , SBI_HART_EXT_SMEPMP , SBI_HART_EXT_SMSTATEEN , SBI_HART_EXT_SSCOFPMF , SBI_HART_EXT_SSTC , SBI_HART_EXT_ZICNTR , SBI_HART_EXT_ZIHPM , SBI_HART_EXT_ZKR , SBI_HART_EXT_SMCNTRPMF , SBI_HART_EXT_XANDESPMU , SBI_HART_EXT_ZICBOZ , SBI_HART_EXT_ZICBOM , SBI_HART_EXT_SVPBMT , SBI_HART_EXT_SDTRIG , SBI_HART_EXT_SMCSRIND , SBI_HART_EXT_SMCDELEG , SBI_HART_EXT_SSCSRIND , SBI_HART_EXT_SSCCFG , SBI_HART_EXT_SVADE , SBI_HART_EXT_SVADU , SBI_HART_EXT_SMNPM , SBI_HART_EXT_ZICFILP , SBI_HART_EXT_ZICFISS , SBI_HART_EXT_SSDBLTRP , SBI_HART_EXT_MAX }

Functions

- int sbi_hart_reinit (struct sbi_scratch *scratch)
- int sbi_hart_init (struct sbi_scratch *scratch, bool cold_boot)

- static ulong sbi_hart_expected_trap_addr (void)
- unsigned int sbi_hart_mhpm_mask (struct sbi_scratch *scratch)
- void sbi_hart_delegation_dump (struct sbi_scratch *scratch, const char *prefix, const char *suffix)
- unsigned int sbi_hart_pmp_count (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_log2gran (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_addrbits (struct sbi_scratch *scratch)
- unsigned int sbi_hart_mhpm_bits (struct sbi_scratch *scratch)
- int sbi_hart_pmp_configure (struct sbi_scratch *scratch)
- int sbi_hart_map_saddr (unsigned long base, unsigned long size)
- int sbi_hart_unmap_saddr (void)
- int sbi_hart_priv_version (struct sbi_scratch *scratch)
- void sbi_hart_get_priv_version_str (struct sbi_scratch *scratch, char *version_str, int nvstr)
- void sbi_hart_update_extension (struct sbi_scratch *scratch, enum sbi_hart_extensions ext, bool enable)
- bool sbi_hart_has_extension (struct sbi_scratch *scratch, enum sbi_hart_extensions ext)
- void sbi_hart_get_extensions_str (struct sbi_scratch *scratch, char *extension_str, int nestr)
- void __attribute__ ((noreturn)) sbi_hart_hang(void)

Variables

- const struct sbi_hart_ext_data sbi_hart_ext [ ]
- void(* sbi_hart_expected_trap)(void)
- void unsigned long arg1
- void unsigned long unsigned long next_addr
- void unsigned long unsigned long unsigned long next_mode
- void unsigned long unsigned long unsigned long bool next_virt

30.80.1 Macro Definition Documentation

30.80.1.1 SBI_SMEPMP_RESV_ENTRY

```
#define SBI_SMEPMP_RESV_ENTRY 0
```

30.80.2 Enumeration Type Documentation

30.80.2.1 sbi_hart_extensions

```
enum sbi_hart_extensions
```

Possible ISA extensions of a hart

Enumerator

SBI_HART_EXT_SMAIA	HART has AIA M-mode CSRs
SBI_HART_EXT_SMEPMP	HART has Smepmp
SBI_HART_EXT_SMSTATEEN	HART has Smstateen CSR
SBI_HART_EXT_SSCOFPMF	Hart has Sscofpmt extension
SBI_HART_EXT_SSTC	HART has Sstc extension
SBI_HART_EXT_ZICNTR	HART has Zicntr extension (i.e. HW cycle, time & instret CSRs)
SBI_HART_EXT_ZIHPM	HART has Zihpm extension
SBI_HART_EXT_ZKR	HART has Zkr extension
SBI_HART_EXT_SMCNTRPMF	Hart has Smcntrpmf extension
SBI_HART_EXT_XANDESPMU	Hart has Xandespmu extension
SBI_HART_EXT_ZICBOZ	Hart has Zicboz extension
SBI_HART_EXT_ZICBOM	Hart has Zicbom extension
SBI_HART_EXT_SVPBMT	Hart has Svpbmt extension
SBI_HART_EXT_SDTRIG	Hart has debug trigger extension
SBI_HART_EXT_SMCSRIND	Hart has Smcsrind extension

Enumerator

SBI_HART_EXT_SMCDELEG	Hart has Smcdeleg extension
SBI_HART_EXT_SSCSRIND	Hart has Sscsrind extension
SBI_HART_EXT_SSCCFG	Hart has Ssccfg extension
SBI_HART_EXT_SVADE	Hart has Svade extension
SBI_HART_EXT_SVADU	Hart has Svadu extension
SBI_HART_EXT_SMNPM	Hart has Smnpm extension
SBI_HART_EXT_ZICFILP	HART has zicfilp extension
SBI_HART_EXT_ZICFISS	HART has zicfiss extension
SBI_HART_EXT_SSDBLTRP	Hart has Ssdbltrp extension
SBI_HART_EXT_MAX	Maximum index of Hart extension

30.80.2.2 sbi_hart_priv_versions

```
enum sbi_hart_priv_versions
```

Possible privileged specification versions of a hart

Enumerator

SBI_HART_PRIV_VER_UNKNOWN	Unknown privileged specification
SBI_HART_PRIV_VER_1_10	Privileged specification v1.10
SBI_HART_PRIV_VER_1_11	Privileged specification v1.11
SBI_HART_PRIV_VER_1_12	Privileged specification v1.12

30.80.3 Function Documentation

30.80.3.1 __attribute__()

```
void __attribute__ (
            (noreturn)  )
```

Here is the caller graph for this function:

30.80.3.2 sbi_hart_delegation_dump()

```
void sbi_hart_delegation_dump (
            struct sbi_scratch * scratch,
            const char * prefix,
            const char * suffix )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.3 sbi_hart_expected_trap_addr()

```
static ulong sbi_hart_expected_trap_addr (
            void ) [inline], [static]
```

Here is the caller graph for this function:

30.80.3.4 sbi_hart_get_extensions_str()

```
void sbi_hart_get_extensions_str (
            struct sbi_scratch * scratch,
            char * extensions_str,
            int nestr )
```

Get the hart extensions in string format

Parameters

scratch	pointer to the HART scratch space
extensions_str	pointer to a char array where the extensions string will be updated
nestr	length of the features_str. The feature string will be truncated if nestr is not long enough.

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.5 sbi_hart_get_priv_version_str()

```
void sbi_hart_get_priv_version_str (
            struct sbi_scratch * scratch,
            char * version_str,
            int nvstr )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.6 sbi_hart_has_extension()

```
bool sbi_hart_has_extension (
            struct sbi_scratch * scratch,
            enum sbi_hart_extensions ext )
```

Check whether a particular hart extension is available

Parameters

scratch	pointer to the HART scratch space
ext	the extension number to check

Returns

true (available) or false (not available)

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.7 sbi_hart_init()

```
int sbi_hart_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.8 sbi_hart_map_saddr()

```
int sbi_hart_map_saddr (
            unsigned long base,
            unsigned long size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.9 sbi_hart_mhpm_bits()

```
unsigned int sbi_hart_mhpm_bits (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.80.3.10 sbi_hart_mhpm_mask()

```
unsigned int sbi_hart_mhpm_mask (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.80.3.11 sbi_hart_pmp_addrbits()

```
unsigned int sbi_hart_pmp_addrbits (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.80.3.12 sbi_hart_pmp_configure()

```
int sbi_hart_pmp_configure (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.13 sbi_hart_pmp_count()

```
unsigned int sbi_hart_pmp_count (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.80.3.14 sbi_hart_pmp_log2gran()

```
unsigned int sbi_hart_pmp_log2gran (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.80.3.15 sbi_hart_priv_version()

```
int sbi_hart_priv_version (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.80.3.16 sbi_hart_reinit()

```
int sbi_hart_reinit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.17 sbi_hart_unmap_saddr()

```
int sbi_hart_unmap_saddr (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.80.3.18 sbi_hart_update_extension()

```
void sbi_hart_update_extension (
            struct sbi_scratch * scratch,
            enum sbi_hart_extensions ext,
            bool enable )
```

Enable/Disable a particular hart extension

Parameters

scratch	pointer to the HART scratch space
ext	the extension number to check
enable	new state of hart extension

Here is the call graph for this function:

30.80.4 Variable Documentation

30.80.4.1 arg1

```
void unsigned long arg1
```

30.80.4.2 next_addr

```
void unsigned long unsigned long next_addr
```

30.80.4.3 next_mode

```
void unsigned long unsigned long unsigned long next_mode
```

30.80.4.4 next_virt

```
void unsigned long unsigned long unsigned long bool next_virt
```

30.80.4.5 sbi_hart_expected_trap

```
void(* sbi_hart_expected_trap) (void) (
            void  )  [extern]
```

30.80.4.6 sbi_hart_ext

```
const struct sbi_hart_ext_data sbi_hart_ext[]  [extern]
```

30.81 sbi_hart.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2019 Western Digital Corporation or its affiliates.
00005  *
00006  * Authors:
00007  *   Anup Patel <anup.patel@wdc.com>
00008  */
00009
00010 #ifndef __SBI_HART_H__
00011 #define __SBI_HART_H__
00012
00013 #include <sbi/sbi_types.h>
00014 #include <sbi/sbi_bitops.h>
00015
00017 enum sbi_hart_priv_versions {
00019     SBI_HART_PRIV_VER_UNKNOWN = 0,
00021     SBI_HART_PRIV_VER_1_10 = 1,
00023     SBI_HART_PRIV_VER_1_11 = 2,
00025     SBI_HART_PRIV_VER_1_12 = 3,
00026 };
00027
00029 enum sbi_hart_extensions {
00031     SBI_HART_EXT_SMAIA = 0,
00033     SBI_HART_EXT_SMEPMP,
00035     SBI_HART_EXT_SMSTATEEN,
00037     SBI_HART_EXT_SSCOFPMF,
00039     SBI_HART_EXT_SSTC,
00041     SBI_HART_EXT_ZICNTR,
00043     SBI_HART_EXT_ZIHPM,
00045     SBI_HART_EXT_ZKR,
00047     SBI_HART_EXT_SMCNTRPMF,
00049     SBI_HART_EXT_XANDESPMU,
00051     SBI_HART_EXT_ZICBOZ,
00053     SBI_HART_EXT_ZICBOM,
00055     SBI_HART_EXT_SVPBMT,
00057     SBI_HART_EXT_SDTRIG,
00059     SBI_HART_EXT_SMCSRIND,
00061     SBI_HART_EXT_SMCDELEG,
00063     SBI_HART_EXT_SSCSRIND,
00065     SBI_HART_EXT_SSCCFG,
00067     SBI_HART_EXT_SVADE,
00069     SBI_HART_EXT_SVADU,
00071     SBI_HART_EXT_SMNPM,
00073     SBI_HART_EXT_ZICFILP,
00075     SBI_HART_EXT_ZICFISS,
00077     SBI_HART_EXT_SSDBLTRP,
00078
00080     SBI_HART_EXT_MAX,
00081 };
00082
00083 struct sbi_hart_ext_data {
```

```
    const unsigned int id;
    const char *name;
};

extern const struct sbi_hart_ext_data sbi_hart_ext[];

/*
 * Smepmp enforces access boundaries between M-mode and
 * S/U-mode. When it is enabled, the PMPs are programmed
 * such that M-mode doesn't have access to S/U-mode memory.
 *
 * To give M-mode R/W access to the shared memory between M and
 * S/U-mode, first entry is reserved. It is disabled at boot.
 * When shared memory access is required, the physical address
 * should be programmed into the first PMP entry with R/W
 * permissions to the M-mode. Once the work is done, it should be
 * unmapped. sbi_hart_map_saddr/sbi_hart_unmap_saddr function
 * pair should be used to map/unmap the shared memory.
 */
#define SBI_SMEPMP_RESV_ENTRY       0

struct sbi_hart_features {
    bool detected;
    int priv_version;
    unsigned long extensions[BITS_TO_LONGS(SBI_HART_EXT_MAX)];
    unsigned int pmp_count;
    unsigned int pmp_addr_bits;
    unsigned int pmp_log2gran;
    unsigned int mhpm_mask;
    unsigned int mhpm_bits;
};

struct sbi_scratch;

int sbi_hart_reinit(struct sbi_scratch *scratch);
int sbi_hart_init(struct sbi_scratch *scratch, bool cold_boot);

extern void (*sbi_hart_expected_trap)(void);
static inline ulong sbi_hart_expected_trap_addr(void)
{
    return (ulong)sbi_hart_expected_trap;
}

unsigned int sbi_hart_mhpm_mask(struct sbi_scratch *scratch);
void sbi_hart_delegation_dump(struct sbi_scratch *scratch,
                  const char *prefix, const char *suffix);
unsigned int sbi_hart_pmp_count(struct sbi_scratch *scratch);
unsigned int sbi_hart_pmp_log2gran(struct sbi_scratch *scratch);
unsigned int sbi_hart_pmp_addrbits(struct sbi_scratch *scratch);
unsigned int sbi_hart_mhpm_bits(struct sbi_scratch *scratch);
int sbi_hart_pmp_configure(struct sbi_scratch *scratch);
int sbi_hart_map_saddr(unsigned long base, unsigned long size);
int sbi_hart_unmap_saddr(void);
int sbi_hart_priv_version(struct sbi_scratch *scratch);
void sbi_hart_get_priv_version_str(struct sbi_scratch *scratch,
                   char *version_str, int nvstr);
void sbi_hart_update_extension(struct sbi_scratch *scratch,
                   enum sbi_hart_extensions ext,
                   bool enable);
bool sbi_hart_has_extension(struct sbi_scratch *scratch,
                enum sbi_hart_extensions ext);
void sbi_hart_get_extensions_str(struct sbi_scratch *scratch,
                 char *extension_str, int nestr);

void __attribute__((noreturn)) sbi_hart_hang(void);

void __attribute__((noreturn))
sbi_hart_switch_mode(unsigned long arg0, unsigned long arg1,
             unsigned long next_addr, unsigned long next_mode,
             bool next_virt);

#endif
```

30.82 include/sbi/sbi_hartmask.h File Reference

```
#include <sbi/sbi_bitmap.h>
#include <sbi/sbi_scratch.h>
```

Include dependency graph for sbi_hartmask.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_hartmask

Macros

- #define SBI_HARTMASK_MAX_BITS 128
- #define SBI_HARTMASK_INIT(__m) bitmap_zero(((__m)->bits), SBI_HARTMASK_MAX_BITS)
- #define SBI_HARTMASK_INIT_EXCEPT(__m, __h)
- #define sbi_hartmask_bits(__m) ((__m)->bits)
- #define sbi_hartmask_for_each_hartindex(__i, __m)

Functions

- static void sbi_hartmask_set_hartindex (u32 i, struct sbi_hartmask *m)
- static void sbi_hartmask_set_hartid (u32 h, struct sbi_hartmask *m)
- static void sbi_hartmask_clear_hartindex (u32 i, struct sbi_hartmask *m)
- static void sbi_hartmask_clear_hartid (u32 h, struct sbi_hartmask *m)
- static int sbi_hartmask_test_hartindex (u32 i, const struct sbi_hartmask *m)
- static int sbi_hartmask_test_hartid (u32 h, const struct sbi_hartmask *m)
- static void sbi_hartmask_set_all (struct sbi_hartmask *dstp)

- static void sbi_hartmask_clear_all (struct sbi_hartmask *dstp)
- static void sbi_hartmask_copy (struct sbi_hartmask *dstp, const struct sbi_hartmask *srcp)
- static void sbi_hartmask_and (struct sbi_hartmask *dstp, const struct sbi_hartmask *src1p, const struct sbi_hartmask *src2p)
- static void sbi_hartmask_or (struct sbi_hartmask *dstp, const struct sbi_hartmask *src1p, const struct sbi_hartmask *src2p)
- static void sbi_hartmask_xor (struct sbi_hartmask *dstp, const struct sbi_hartmask *src1p, const struct sbi_hartmask *src2p)

30.82.1 Macro Definition Documentation

30.82.1.1 sbi_hartmask_bits

```
#define sbi_hartmask_bits(
            __m ) ((__m)->bits)
```

Get underlying bitmap of hartmask

Parameters

m	the hartmask pointer

30.82.1.2 sbi_hartmask_for_each_hartindex

```
#define sbi_hartmask_for_each_hartindex(
            __i,
            __m )
```

Value:

```
for((__i) = find_first_bit((__m)->bits, SBI_HARTMASK_MAX_BITS); \
    (__i) < SBI_HARTMASK_MAX_BITS; \
    (__i) = find_next_bit((__m)->bits, SBI_HARTMASK_MAX_BITS, (__i) + 1))
```

Iterate over each HART index in hartmask __i hart index __m hartmask

30.82.1.3 SBI_HARTMASK_INIT

```
#define SBI_HARTMASK_INIT(
            __m )  bitmap_zero(((__m)->bits), SBI_HARTMASK_MAX_BITS)
```

Initialize hartmask to zero

30.82.1.4 SBI_HARTMASK_INIT_EXCEPT

```
#define SBI_HARTMASK_INIT_EXCEPT(
            __m,
            __h )
```

Value:

```
do { \
    u32 __i = sbi_hartid_to_hartindex(__h); \
    bitmap_zero_except(((__m)->bits), __i, SBI_HARTMASK_MAX_BITS); \
} while(0)
```

Initialize hartmask to zero except a particular HART id

30.82.1.5 SBI_HARTMASK_MAX_BITS

```
#define SBI_HARTMASK_MAX_BITS 128
```

Maximum number of bits in a hartmask

The hartmask is indexed using physical HART id so this define also represents the maximum number of HART ids generic OpenSBI can handle.

30.82.2 Function Documentation

30.82.2.1 sbi_hartmask_and()

```
static void sbi_hartmask_and (
            struct sbi_hartmask * dstp,
            const struct sbi_hartmask * src1p,
            const struct sbi_hartmask * src2p )  [inline], [static]
```

∗dstp = ∗src1p & ∗src2p

Parameters

dstp	the hartmask result
src1p	the first input
src2p	the second input

Here is the call graph for this function:

Here is the caller graph for this function:

30.82.2.2 sbi_hartmask_clear_all()

```
static void sbi_hartmask_clear_all (
            struct sbi_hartmask * dstp )  [inline], [static]
```

Clear all HARTs in a hartmask

Parameters

dstp	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:

30.82.2.3 sbi_hartmask_clear_hartid()

```
static void sbi_hartmask_clear_hartid (
            u32 h,
            struct sbi_hartmask * m )  [inline], [static]
```

Clear a HART id in hartmask

Parameters

h	HART id to clear
m	the hartmask pointer

Here is the call graph for this function:

30.82.2.4 sbi_hartmask_clear_hartindex()

```
static void sbi_hartmask_clear_hartindex (
            u32 i,
            struct sbi_hartmask * m )  [inline], [static]
```

Clear a HART index in hartmask

Parameters

i	HART index to clear
m	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:

30.82.2.5 sbi_hartmask_copy()

```
static void sbi_hartmask_copy (
            struct sbi_hartmask * dstp,
            const struct sbi_hartmask * srcp )  [inline], [static]
```

∗dstp = ∗srcp

Parameters

dstp	the hartmask destination
srcp	the hartmask source

Here is the call graph for this function:

Here is the caller graph for this function:

30.82.2.6 sbi_hartmask_or()

```
static void sbi_hartmask_or (
            struct sbi_hartmask * dstp,
            const struct sbi_hartmask * src1p,
            const struct sbi_hartmask * src2p )  [inline], [static]
```

$*$dstp = $*$src1p | $*$src2p

Parameters

dstp	the hartmask result
src1p	the first input
src2p	the second input

Here is the call graph for this function:

Here is the caller graph for this function:

30.82.2.7 sbi_hartmask_set_all()

```
static void sbi_hartmask_set_all (
            struct sbi_hartmask * dstp )  [inline], [static]
```

Set all HARTs in a hartmask

Parameters

dstp	the hartmask pointer

Here is the call graph for this function:

30.82.2.8 sbi_hartmask_set_hartid()

```
static void sbi_hartmask_set_hartid (
            u32 h,
            struct sbi_hartmask * m )  [inline], [static]
```

Set a HART id in hartmask

Parameters

h	HART id to set
m	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:

30.82.2.9 sbi_hartmask_set_hartindex()

```
static void sbi_hartmask_set_hartindex (
            u32 i,
            struct sbi_hartmask * m )  [inline], [static]
```

Set a HART index in hartmask

Parameters

i	HART index to set
m	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:

30.82.2.10 sbi_hartmask_test_hartid()

```
static int sbi_hartmask_test_hartid (
            u32 h,
            const struct sbi_hartmask * m )  [inline], [static]
```

Test a HART id in hartmask

Parameters

h	HART id to test
m	the hartmask pointer

Here is the call graph for this function:

30.82.2.11 sbi_hartmask_test_hartindex()

```
static int sbi_hartmask_test_hartindex (
```

```
u32 i,
const struct sbi_hartmask * m )  [inline], [static]
```

Test a HART index in hartmask

Parameters

i	HART index to test
m	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:

30.82.2.12 sbi_hartmask_xor()

```
static void sbi_hartmask_xor (
            struct sbi_hartmask * dstp,
            const struct sbi_hartmask * src1p,
            const struct sbi_hartmask * src2p )  [inline], [static]
```

∗dstp = ∗src1p ^ ∗src2p

Parameters

dstp	the hartmask result
src1p	the first input
src2p	the second input

Here is the call graph for this function:

30.83 sbi_hartmask.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_HARTMASK_H__
#define __SBI_HARTMASK_H__

#include <sbi/sbi_bitmap.h>
#include <sbi/sbi_scratch.h>

#define SBI_HARTMASK_MAX_BITS       128

struct sbi_hartmask {
    DECLARE_BITMAP(bits, SBI_HARTMASK_MAX_BITS);
};

#define SBI_HARTMASK_INIT(__m)      \
    bitmap_zero(((__m)->bits), SBI_HARTMASK_MAX_BITS)

#define SBI_HARTMASK_INIT_EXCEPT(__m, __h)  \
    do { \
        u32 __i = sbi_hartid_to_hartindex(__h); \
        bitmap_zero_except(((__m)->bits), __i, SBI_HARTMASK_MAX_BITS); \
    } while(0)

#define sbi_hartmask_bits(__m)      ((__m)->bits)

static inline void sbi_hartmask_set_hartindex(u32 i, struct sbi_hartmask *m)
{
    if (i < SBI_HARTMASK_MAX_BITS)
        __set_bit(i, m->bits);
}

static inline void sbi_hartmask_set_hartid(u32 h, struct sbi_hartmask *m)
{
    sbi_hartmask_set_hartindex(sbi_hartid_to_hartindex(h), m);
}

static inline void sbi_hartmask_clear_hartindex(u32 i, struct sbi_hartmask *m)
{
    if (i < SBI_HARTMASK_MAX_BITS)
        __clear_bit(i, m->bits);
}

static inline void sbi_hartmask_clear_hartid(u32 h, struct sbi_hartmask *m)
{
    sbi_hartmask_clear_hartindex(sbi_hartid_to_hartindex(h), m);
}

static inline int sbi_hartmask_test_hartindex(u32 i,
                          const struct sbi_hartmask *m)
{
    if (i < SBI_HARTMASK_MAX_BITS)
        return __test_bit(i, m->bits);
    return 0;
}

static inline int sbi_hartmask_test_hartid(u32 h, const struct sbi_hartmask *m)
{
    return sbi_hartmask_test_hartindex(sbi_hartid_to_hartindex(h), m);
```

```
}

static inline void sbi_hartmask_set_all(struct sbi_hartmask *dstp)
{
    bitmap_fill(sbi_hartmask_bits(dstp), SBI_HARTMASK_MAX_BITS);
}

static inline void sbi_hartmask_clear_all(struct sbi_hartmask *dstp)
{
    bitmap_zero(sbi_hartmask_bits(dstp), SBI_HARTMASK_MAX_BITS);
}

static inline void sbi_hartmask_copy(struct sbi_hartmask *dstp,
                     const struct sbi_hartmask *srcp)
{
    bitmap_copy(sbi_hartmask_bits(dstp), sbi_hartmask_bits(srcp),
            SBI_HARTMASK_MAX_BITS);
}

static inline void sbi_hartmask_and(struct sbi_hartmask *dstp,
                    const struct sbi_hartmask *src1p,
                    const struct sbi_hartmask *src2p)
{
    bitmap_and(sbi_hartmask_bits(dstp), sbi_hartmask_bits(src1p),
           sbi_hartmask_bits(src2p), SBI_HARTMASK_MAX_BITS);
}

static inline void sbi_hartmask_or(struct sbi_hartmask *dstp,
                   const struct sbi_hartmask *src1p,
                   const struct sbi_hartmask *src2p)
{
    bitmap_or(sbi_hartmask_bits(dstp), sbi_hartmask_bits(src1p),
          sbi_hartmask_bits(src2p), SBI_HARTMASK_MAX_BITS);
}

static inline void sbi_hartmask_xor(struct sbi_hartmask *dstp,
                    const struct sbi_hartmask *src1p,
                    const struct sbi_hartmask *src2p)
{
    bitmap_xor(sbi_hartmask_bits(dstp), sbi_hartmask_bits(src1p),
           sbi_hartmask_bits(src2p), SBI_HARTMASK_MAX_BITS);
}

#define sbi_hartmask_for_each_hartindex(__i, __m) \
    for((__i) = find_first_bit((__m)->bits, SBI_HARTMASK_MAX_BITS); \
        (__i) < SBI_HARTMASK_MAX_BITS; \
        (__i) = find_next_bit((__m)->bits, SBI_HARTMASK_MAX_BITS, (__i) + 1))

#endif
```

30.84 include/sbi/sbi_heap.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_heap.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define HEAP_BASE_ALIGN 1024

Functions

- void ∗ sbi_malloc_from (struct sbi_heap_control ∗hpctrl, size_t size)
- static void ∗ sbi_malloc (size_t size)
- void ∗ sbi_aligned_alloc_from (struct sbi_heap_control ∗hpctrl, size_t alignment, size_t size)
- static void ∗ sbi_aligned_alloc (size_t alignment, size_t size)
- void ∗ sbi_zalloc_from (struct sbi_heap_control ∗hpctrl, size_t size)
- static void ∗ sbi_zalloc (size_t size)
- static void ∗ sbi_calloc (size_t nitems, size_t size)
- static void ∗ sbi_calloc_from (struct sbi_heap_control ∗hpctrl, size_t nitems, size_t size)
- void sbi_free_from (struct sbi_heap_control ∗hpctrl, void ∗ptr)
- static void sbi_free (void ∗ptr)
- unsigned long sbi_heap_free_space_from (struct sbi_heap_control ∗hpctrl)
- static unsigned long sbi_heap_free_space (void)
- unsigned long sbi_heap_used_space_from (struct sbi_heap_control ∗hpctrl)
- static unsigned long sbi_heap_used_space (void)
- unsigned long sbi_heap_reserved_space_from (struct sbi_heap_control ∗hpctrl)
- static unsigned long sbi_heap_reserved_space (void)
- int sbi_heap_init (struct sbi_scratch ∗scratch)
- int sbi_heap_init_new (struct sbi_heap_control ∗hpctrl, unsigned long base, unsigned long size)
- int sbi_heap_alloc_new (struct sbi_heap_control ∗∗hpctrl)

Variables

- struct sbi_heap_control global_hpctrl

30.84.1 Macro Definition Documentation

30.84.1.1 HEAP_BASE_ALIGN

```
#define HEAP_BASE_ALIGN 1024
```

30.84.2 Function Documentation

30.84.2.1 sbi_aligned_alloc()

```
static void * sbi_aligned_alloc (
            size_t alignment,
            size_t size )  [inline], [static]
```

Here is the call graph for this function:

30.84.2.2 sbi_aligned_alloc_from()

```
void * sbi_aligned_alloc_from (
            struct sbi_heap_control * hpctrl,
            size_t alignment,
            size_t size )
```

Allocate aligned from heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.3 sbi_calloc()

```
static void * sbi_calloc (
            size_t nitems,
            size_t size )  [inline], [static]
```

Allocate array from heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.4 sbi_calloc_from()

```
static void * sbi_calloc_from (
            struct sbi_heap_control * hpctrl,
            size_t nitems,
            size_t size )  [inline], [static]
```

Here is the call graph for this function:

30.84.2.5 sbi_free()

```
static void sbi_free (
            void * ptr )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.6 sbi_free_from()

```
void sbi_free_from (
            struct sbi_heap_control * hpctrl,
            void * ptr )
```

Free-up to heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.7 sbi_heap_alloc_new()

```
int sbi_heap_alloc_new (
            struct sbi_heap_control ** hpctrl )
```

Here is the call graph for this function:

30.84.2.8 sbi_heap_free_space()

```
static unsigned long sbi_heap_free_space (
            void ) [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.9 sbi_heap_free_space_from()

```
unsigned long sbi_heap_free_space_from (
            struct sbi_heap_control * hpctrl )
```

Amount (in bytes) of free space in the heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.10 sbi_heap_init()

```
int sbi_heap_init (
            struct sbi_scratch * scratch )
```

Initialize heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.11 sbi_heap_init_new()

```
int sbi_heap_init_new (
            struct sbi_heap_control * hpctrl,
            unsigned long base,
            unsigned long size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.12 sbi_heap_reserved_space()

```
static unsigned long sbi_heap_reserved_space (
            void ) [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.13 sbi_heap_reserved_space_from()

```
unsigned long sbi_heap_reserved_space_from (
            struct sbi_heap_control * hpctrl )
```

Amount (in bytes) of reserved space in the heap area Here is the caller graph for this function:

30.84.2.14 sbi_heap_used_space()

```
static unsigned long sbi_heap_used_space (
            void ) [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.15 sbi_heap_used_space_from()

```
unsigned long sbi_heap_used_space_from (
            struct sbi_heap_control * hpctrl )
```

Amount (in bytes) of used space in the heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.16 sbi_malloc()

```
static void * sbi_malloc (
            size_t size )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.17 sbi_malloc_from()

```
void * sbi_malloc_from (
            struct sbi_heap_control * hpctrl,
            size_t size )
```

Allocate from heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.18 sbi_zalloc()

```
static void * sbi_zalloc (
            size_t size ) [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.84.2.19 sbi_zalloc_from()

```
void * sbi_zalloc_from (
            struct sbi_heap_control * hpctrl,
            size_t size )
```

Zero allocate from heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.84.3 Variable Documentation

30.84.3.1 global_hpctrl

```
struct sbi_heap_control global_hpctrl  [extern]
```

30.85 sbi_heap.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2023 Ventana Micro Systems Inc.
00005  *
00006  * Authors:
```

```
 *   Anup Patel<apatel@ventanamicro.com>
 */

#ifndef __SBI_HEAP_H__
#define __SBI_HEAP_H__

#include <sbi/sbi_types.h>

/* Opaque declaration of heap control struct */
struct sbi_heap_control;

/* Global heap control structure */
extern struct sbi_heap_control global_hpctrl;

/* Alignment of heap base address and size */
#define HEAP_BASE_ALIGN         1024

struct sbi_scratch;

void *sbi_malloc_from(struct sbi_heap_control *hpctrl, size_t size);

static inline void *sbi_malloc(size_t size)
{
    return sbi_malloc_from(&global_hpctrl, size);
}

void *sbi_aligned_alloc_from(struct sbi_heap_control *hpctrl,
                 size_t alignment,size_t size);

static inline void *sbi_aligned_alloc(size_t alignment, size_t size)
{
    return sbi_aligned_alloc_from(&global_hpctrl, alignment, size);
}

void *sbi_zalloc_from(struct sbi_heap_control *hpctrl, size_t size);

static inline void *sbi_zalloc(size_t size)
{
    return sbi_zalloc_from(&global_hpctrl, size);
}

static inline void *sbi_calloc(size_t nitems, size_t size)
{
    return sbi_zalloc(nitems * size);
}

static inline void *sbi_calloc_from(struct sbi_heap_control *hpctrl,
                    size_t nitems, size_t size)
{
    return sbi_zalloc_from(hpctrl, nitems * size);
}

void sbi_free_from(struct sbi_heap_control *hpctrl, void *ptr);

static inline void sbi_free(void *ptr)
{
    return sbi_free_from(&global_hpctrl, ptr);
}

unsigned long sbi_heap_free_space_from(struct sbi_heap_control *hpctrl);

static inline unsigned long sbi_heap_free_space(void)
{
    return sbi_heap_free_space_from(&global_hpctrl);
}

unsigned long sbi_heap_used_space_from(struct sbi_heap_control *hpctrl);

static inline unsigned long sbi_heap_used_space(void)
{
    return sbi_heap_used_space_from(&global_hpctrl);
}

unsigned long sbi_heap_reserved_space_from(struct sbi_heap_control *hpctrl);

static inline unsigned long sbi_heap_reserved_space(void)
{
    return sbi_heap_reserved_space_from(&global_hpctrl);
}

int sbi_heap_init(struct sbi_scratch *scratch);
int sbi_heap_init_new(struct sbi_heap_control *hpctrl, unsigned long base,
               unsigned long size);
int sbi_heap_alloc_new(struct sbi_heap_control **hpctrl);

#endif
```

30.86 include/sbi/sbi_hfence.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void __sbi_hfence_gvma_vmid_gpa (unsigned long gpa_divby_4, unsigned long vmid)
- void __sbi_hfence_gvma_vmid (unsigned long vmid)
- void __sbi_hfence_gvma_gpa (unsigned long gpa_divby_4)
- void __sbi_hfence_gvma_all (void)
- void __sbi_hfence_vvma_asid_va (unsigned long va, unsigned long asid)
- void __sbi_hfence_vvma_asid (unsigned long asid)
- void __sbi_hfence_vvma_va (unsigned long va)
- void __sbi_hfence_vvma_all (void)

30.86.1 Function Documentation

30.86.1.1 __sbi_hfence_gvma_all()

```
void __sbi_hfence_gvma_all (
            void  )
```

Invalidate all possible Stage2 TLBs Here is the caller graph for this function:

30.86.1.2 __sbi_hfence_gvma_gpa()

```
void __sbi_hfence_gvma_gpa (
            unsigned long gpa_divby_4 )
```

Invalidate Stage2 TLBs for given guest physical address Here is the caller graph for this function:

30.86.1.3 __sbi_hfence_gvma_vmid()

```
void __sbi_hfence_gvma_vmid (
            unsigned long vmid )
```

Invalidate Stage2 TLBs for given VMID Here is the caller graph for this function:

30.86.1.4 __sbi_hfence_gvma_vmid_gpa()

```
void __sbi_hfence_gvma_vmid_gpa (
            unsigned long gpa_divby_4,
            unsigned long vmid )
```

Invalidate Stage2 TLBs for given VMID and guest physical address Here is the caller graph for this function:

30.86.1.5 __sbi_hfence_vvma_all()

```
void __sbi_hfence_vvma_all (
            void )
```

Invalidate all possible Stage2 TLBs Here is the caller graph for this function:

30.86.1.6 __sbi_hfence_vvma_asid()

```
void __sbi_hfence_vvma_asid (
            unsigned long asid )
```

Invalidate unified TLB entries for given ASID for a guest Here is the caller graph for this function:

30.86.1.7 __sbi_hfence_vvma_asid_va()

```
void __sbi_hfence_vvma_asid_va (
            unsigned long va,
            unsigned long asid )
```

Invalidate unified TLB entries for given asid and guest virtual address Here is the caller graph for this function:

30.86.1.8 __sbi_hfence_vvma_va()

```
void __sbi_hfence_vvma_va (
            unsigned long va )
```

Invalidate unified TLB entries for a given guest virtual address Here is the caller graph for this function:

30.87 sbi_hfence.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Atish Patra <atish.patra@wdc.com>
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_FENCE_H__
#define __SBI_FENCE_H__

void __sbi_hfence_gvma_vmid_gpa(unsigned long gpa_divby_4,
                unsigned long vmid);

void __sbi_hfence_gvma_vmid(unsigned long vmid);

void __sbi_hfence_gvma_gpa(unsigned long gpa_divby_4);

void __sbi_hfence_gvma_all(void);

void __sbi_hfence_vvma_asid_va(unsigned long va, unsigned long asid);

void __sbi_hfence_vvma_asid(unsigned long asid);

void __sbi_hfence_vvma_va(unsigned long va);

void __sbi_hfence_vvma_all(void);

#endif
```

30.88 include/sbi/sbi_hsm.h File Reference

```
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_hsm.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_hsm_device

Functions

- const struct sbi_hsm_device ∗ sbi_hsm_get_device (void)
- void sbi_hsm_set_device (const struct sbi_hsm_device ∗dev)
- int sbi_hsm_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void __noreturn sbi_hsm_exit (struct sbi_scratch ∗scratch)
- int sbi_hsm_hart_start (struct sbi_scratch ∗scratch, const struct sbi_domain ∗dom, u32 hartid, ulong saddr, ulong smode, ulong arg1)
- int sbi_hsm_hart_stop (struct sbi_scratch ∗scratch, bool exitnow)
- void sbi_hsm_hart_resume_start (struct sbi_scratch ∗scratch)
- void __noreturn sbi_hsm_hart_resume_finish (struct sbi_scratch ∗scratch, u32 hartid)
- int sbi_hsm_hart_suspend (struct sbi_scratch ∗scratch, u32 suspend_type, ulong raddr, ulong rmode, ulong arg1)

- bool sbi_hsm_hart_change_state (struct sbi_scratch *scratch, long oldstate, long newstate)
- int __sbi_hsm_hart_get_state (u32 hartindex)
- int sbi_hsm_hart_get_state (const struct sbi_domain *dom, u32 hartid)
- int sbi_hsm_hart_interruptible_mask (const struct sbi_domain *dom, struct sbi_hartmask *mask)
- void __sbi_hsm_suspend_non_ret_save (struct sbi_scratch *scratch)
- void __noreturn sbi_hsm_hart_start_finish (struct sbi_scratch *scratch, u32 hartid)

30.88.1 Function Documentation

30.88.1.1 __sbi_hsm_hart_get_state()

```
int __sbi_hsm_hart_get_state (
            u32 hartindex )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.2 __sbi_hsm_suspend_non_ret_save()

```
void __sbi_hsm_suspend_non_ret_save (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.3 sbi_hsm_exit()

```
void __noreturn sbi_hsm_exit (
            struct sbi_scratch * scratch )
```

As platform is lacking support for hotplug, directly jump to warmboot and wait for interrupts in warmboot. We do it preemptively in order preserve the hart states and reuse the code path for hotplug.Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.4 sbi_hsm_get_device()

```
const struct sbi_hsm_device * sbi_hsm_get_device (
            void )
```

Here is the caller graph for this function:

30.88.1.5 sbi_hsm_hart_change_state()

```
bool sbi_hsm_hart_change_state (
            struct sbi_scratch * scratch,
            long oldstate,
            long newstate )
```

Here is the caller graph for this function:

30.88.1.6 sbi_hsm_hart_get_state()

```
int sbi_hsm_hart_get_state (
            const struct sbi_domain * dom,
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.7 sbi_hsm_hart_interruptible_mask()

```
int sbi_hsm_hart_interruptible_mask (
            const struct sbi_domain * dom,
            struct sbi_hartmask * mask )
```

Get the mask of harts which are valid IPI targets

Parameters

dom	the domain to be used for output HART mask
mask	the output hartmask to fill

Returns

0 on success and SBI_Exxx (< 0) on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.8 sbi_hsm_hart_resume_finish()

```
void __noreturn sbi_hsm_hart_resume_finish (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.9 sbi_hsm_hart_resume_start()

```
void sbi_hsm_hart_resume_start (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.10 sbi_hsm_hart_start()

```
int sbi_hsm_hart_start (
            struct sbi_scratch * scratch,
            const struct sbi_domain * dom,
            u32 hartid,
            ulong saddr,
            ulong smode,
            ulong arg1 )
```

if a hart is already transition to start or stop, another start call is considered as invalid request.Here is the call graph

for this function:

Here is the caller graph for this function:

30.88.1.11 sbi_hsm_hart_start_finish()

```
void __noreturn sbi_hsm_hart_start_finish (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.12 sbi_hsm_hart_stop()

```
int sbi_hsm_hart_stop (
            struct sbi_scratch * scratch,
            bool exitnow )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.13 sbi_hsm_hart_suspend()

```
int sbi_hsm_hart_suspend (
            struct sbi_scratch * scratch,
            u32 suspend_type,
            ulong raddr,
```

```
            ulong rmode,
            ulong arg1 )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.14 sbi_hsm_init()

```
int sbi_hsm_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.88.1.15 sbi_hsm_set_device()

```
void sbi_hsm_set_device (
            const struct sbi_hsm_device * dev )
```

Here is the caller graph for this function:

30.89 sbi_hsm.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Atish Patra <atish.patra@wdc.com>
 */

#ifndef __SBI_HSM_H__
#define __SBI_HSM_H__

#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_types.h>

struct sbi_hsm_device {
    char name[32];

    int (*hart_start)(u32 hartid, ulong saddr);

    int (*hart_stop)(void);

    int (*hart_suspend)(u32 suspend_type, ulong mmode_resume_addr);

    void (*hart_resume)(void);
};

struct sbi_domain;
struct sbi_scratch;

const struct sbi_hsm_device *sbi_hsm_get_device(void);

void sbi_hsm_set_device(const struct sbi_hsm_device *dev);

int sbi_hsm_init(struct sbi_scratch *scratch, bool cold_boot);
void __noreturn sbi_hsm_exit(struct sbi_scratch *scratch);

int sbi_hsm_hart_start(struct sbi_scratch *scratch,
                       const struct sbi_domain *dom,
```

```
                u32 hartid, ulong saddr, ulong smode, ulong arg1);
int sbi_hsm_hart_stop(struct sbi_scratch *scratch, bool exitnow);
void sbi_hsm_hart_resume_start(struct sbi_scratch *scratch);
void __noreturn sbi_hsm_hart_resume_finish(struct sbi_scratch *scratch,
                        u32 hartid);
int sbi_hsm_hart_suspend(struct sbi_scratch *scratch, u32 suspend_type,
              ulong raddr, ulong rmode, ulong arg1);
bool sbi_hsm_hart_change_state(struct sbi_scratch *scratch, long oldstate,
                    long newstate);
int __sbi_hsm_hart_get_state(u32 hartindex);
int sbi_hsm_hart_get_state(const struct sbi_domain *dom, u32 hartid);
int sbi_hsm_hart_interruptible_mask(const struct sbi_domain *dom,
                     struct sbi_hartmask *mask);
void __sbi_hsm_suspend_non_ret_save(struct sbi_scratch *scratch);
void __noreturn sbi_hsm_hart_start_finish(struct sbi_scratch *scratch,
                       u32 hartid);

#endif
```

30.90 include/sbi/sbi_illegal_insn.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_illegal_insn.h:

This graph shows which files directly or indirectly include this file:

Functions

- int sbi_illegal_insn_handler (struct sbi_trap_context *tcntx)

30.90.1 Function Documentation

30.90.1.1 sbi_illegal_insn_handler()

```
int sbi_illegal_insn_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.91 sbi_illegal_insn.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2019 Western Digital Corporation or its affiliates.
00005  *
00006  * Authors:
00007  *   Anup Patel <anup.patel@wdc.com>
00008  */
00009
00010 #ifndef __SBI_ILLEGAl_INSN_H__
00011 #define __SBI_ILLEGAl_INSN_H__
00012
00013 #include <sbi/sbi_types.h>
00014
00015 struct sbi_trap_context;
00016
00017 int sbi_illegal_insn_handler(struct sbi_trap_context *tcntx);
00018
00019 #endif
```

30.92 include/sbi/sbi_init.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_init.h:

This graph shows which files directly or indirectly include this file:

Functions

- void __noreturn sbi_init (struct sbi_scratch *scratch)
- unsigned long sbi_entry_count (u32 hartindex)
- unsigned long sbi_init_count (u32 hartindex)
- void __noreturn sbi_exit (struct sbi_scratch *scratch)

30.92.1 Function Documentation

30.92.1.1 sbi_entry_count()

```
unsigned long sbi_entry_count (
            u32 hartindex )
```

Here is the caller graph for this function:

30.92.1.2 sbi_exit()

```
void __noreturn sbi_exit (
            struct sbi_scratch * scratch )
```

Exit OpenSBI library for current HART and stop HART
The function expects following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. Stack pointer (SP) is setup for current HART

Parameters

scratch	pointer to sbi_scratch of current HART

Here is the call graph for this function:

Here is the caller graph for this function:

30.92.1.3 sbi_init()

```
void __noreturn sbi_init (
            struct sbi_scratch * scratch )
```

Initialize OpenSBI library for current HART and jump to next booting stage.
The function expects following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. Stack pointer (SP) is setup for current HART
3. Interrupts are disabled in MSTATUS CSR

4. All interrupts are disabled in MIE CSR

Parameters

scratch	pointer to sbi_scratch of current HART

Here is the call graph for this function:

30.92.1.4 sbi_init_count()

```
unsigned long sbi_init_count (
            u32 hartindex )
```

Here is the caller graph for this function:

30.93 sbi_init.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_INIT_H__
#define __SBI_INIT_H__

#include <sbi/sbi_types.h>

struct sbi_scratch;

void __noreturn sbi_init(struct sbi_scratch *scratch);

unsigned long sbi_entry_count(u32 hartindex);

unsigned long sbi_init_count(u32 hartindex);

void __noreturn sbi_exit(struct sbi_scratch *scratch);

#endif
```

30.94 include/sbi/sbi_ipi.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_ipi.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_ipi_device
- struct sbi_ipi_event_ops

Macros

- #define SBI_IPI_EVENT_MAX (8 ∗ __SIZEOF_LONG__)

Enumerations

- enum sbi_ipi_update_type { SBI_IPI_UPDATE_SUCCESS , SBI_IPI_UPDATE_BREAK , SBI_IPI_UPDATE_RETRY }

Functions

- int sbi_ipi_send_many (ulong hmask, ulong hbase, u32 event, void ∗data)
- int sbi_ipi_event_create (const struct sbi_ipi_event_ops ∗ops)
- void sbi_ipi_event_destroy (u32 event)
- int sbi_ipi_send_smode (ulong hmask, ulong hbase)
- void sbi_ipi_clear_smode (void)
- int sbi_ipi_send_halt (ulong hmask, ulong hbase)
- void sbi_ipi_process (void)
- int sbi_ipi_raw_send (u32 hartindex)
- void sbi_ipi_raw_clear (void)
- const struct sbi_ipi_device ∗ sbi_ipi_get_device (void)
- void sbi_ipi_set_device (const struct sbi_ipi_device ∗dev)
- int sbi_ipi_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_ipi_exit (struct sbi_scratch ∗scratch)

30.94.1 Macro Definition Documentation

30.94.1.1 SBI_IPI_EVENT_MAX

```
#define SBI_IPI_EVENT_MAX (8 * __SIZEOF_LONG__)
```

30.94.2 Enumeration Type Documentation

30.94.2.1 sbi_ipi_update_type

```
enum sbi_ipi_update_type
```

Enumerator

SBI_IPI_UPDATE_SUCCESS	
SBI_IPI_UPDATE_BREAK	
SBI_IPI_UPDATE_RETRY	

30.94.3 Function Documentation

30.94.3.1 sbi_ipi_clear_smode()

```
void sbi_ipi_clear_smode (
            void  )
```

Here is the caller graph for this function:

30.94.3.2 sbi_ipi_event_create()

```
int sbi_ipi_event_create (
            const struct sbi_ipi_event_ops * ops )
```

Here is the caller graph for this function:

30.94.3.3 sbi_ipi_event_destroy()

```
void sbi_ipi_event_destroy (
            u32 event )
```

30.94.3.4 sbi_ipi_exit()

```
void sbi_ipi_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.94.3.5 sbi_ipi_get_device()

```
const struct sbi_ipi_device * sbi_ipi_get_device (
            void )
```

Here is the caller graph for this function:

30.94.3.6 sbi_ipi_init()

```
int sbi_ipi_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.94.3.7 sbi_ipi_process()

```
void sbi_ipi_process (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.94.3.8 sbi_ipi_raw_clear()

```
void sbi_ipi_raw_clear (
            void  )
```

Here is the caller graph for this function:

30.94.3.9 sbi_ipi_raw_send()

```
int sbi_ipi_raw_send (
            u32 hartindex )
```

Here is the caller graph for this function:

30.94.3.10 sbi_ipi_send_halt()

```
int sbi_ipi_send_halt (
            ulong hmask,
            ulong hbase )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.94.3.11 sbi_ipi_send_many()

```
int sbi_ipi_send_many (
            ulong hmask,
            ulong hbase,
            u32 event,
            void * data )
```

As this this function only handlers scalar values of hart mask, it must be set to all online harts if the intention is to

send IPIs to all the harts. If hmask is zero, no IPIs will be sent. Here is the call graph for this function:

Here is the caller graph for this function:

30.94.3.12 sbi_ipi_send_smode()

```
int sbi_ipi_send_smode (
            ulong hmask,
            ulong hbase )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.94.3.13 sbi_ipi_set_device()

```
void sbi_ipi_set_device (
            const struct sbi_ipi_device * dev )
```

Here is the caller graph for this function:

30.95 sbi_ipi.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_IPI_H__
#define __SBI_IPI_H__

#include <sbi/sbi_types.h>

/* clang-format off */

#define SBI_IPI_EVENT_MAX           (8 * __SIZEOF_LONG__)

/* clang-format on */

struct sbi_ipi_device {
    char name[32];

    void (*ipi_send)(u32 hart_index);

    void (*ipi_clear)(void);
};

enum sbi_ipi_update_type {
    SBI_IPI_UPDATE_SUCCESS,
    SBI_IPI_UPDATE_BREAK,
    SBI_IPI_UPDATE_RETRY,
```

```
};

struct sbi_scratch;

struct sbi_ipi_event_ops {
    char name[32];

    int (* update)(struct sbi_scratch *scratch,
            struct sbi_scratch *remote_scratch,
            u32 remote_hartindex, void *data);

    void (* sync)(struct sbi_scratch *scratch);

    void (* process)(struct sbi_scratch *scratch);
};

int sbi_ipi_send_many(ulong hmask, ulong hbase, u32 event, void *data);

int sbi_ipi_event_create(const struct sbi_ipi_event_ops *ops);

void sbi_ipi_event_destroy(u32 event);

int sbi_ipi_send_smode(ulong hmask, ulong hbase);

void sbi_ipi_clear_smode(void);

int sbi_ipi_send_halt(ulong hmask, ulong hbase);

void sbi_ipi_process(void);

int sbi_ipi_raw_send(u32 hartindex);

void sbi_ipi_raw_clear(void);

const struct sbi_ipi_device *sbi_ipi_get_device(void);

void sbi_ipi_set_device(const struct sbi_ipi_device *dev);

int sbi_ipi_init(struct sbi_scratch *scratch, bool cold_boot);

void sbi_ipi_exit(struct sbi_scratch *scratch);

#endif
```

30.96 include/sbi/sbi_irqchip.h File Reference

```
#include <sbi/sbi_list.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_irqchip.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_irqchip_device

Functions

- int sbi_irqchip_process (void)
- void sbi_irqchip_add_device (struct sbi_irqchip_device ∗dev)
- int sbi_irqchip_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_irqchip_exit (struct sbi_scratch ∗scratch)

30.96.1 Function Documentation

30.96.1.1 sbi_irqchip_add_device()

```
void sbi_irqchip_add_device (
            struct sbi_irqchip_device * dev )
```

Register an irqchip device to receive callbacks Here is the call graph for this function:

Here is the caller graph for this function:

30.96.1.2 sbi_irqchip_exit()

```
void sbi_irqchip_exit (
            struct sbi_scratch * scratch )
```

Exit interrupt controllers Here is the call graph for this function:

Here is the caller graph for this function:

30.96.1.3 sbi_irqchip_init()

```
int sbi_irqchip_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Initialize interrupt controllers Here is the call graph for this function:

Here is the caller graph for this function:

30.96.1.4 sbi_irqchip_process()

```
int sbi_irqchip_process (
            void  )
```

Process external interrupts

This function is called by sbi_trap_handler() to handle external interrupts.

Parameters

regs	pointer for trap registers

Here is the caller graph for this function:

30.97 sbi_irqchip.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2022 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __SBI_IRQCHIP_H__
#define __SBI_IRQCHIP_H__

#include <sbi/sbi_list.h>
#include <sbi/sbi_types.h>

struct sbi_scratch;

struct sbi_irqchip_device {
    struct sbi_dlist node;

    int (*warm_init)(struct sbi_irqchip_device *dev);

    int (*irq_handle)(void);
};

int sbi_irqchip_process(void);

void sbi_irqchip_add_device(struct sbi_irqchip_device *dev);

int sbi_irqchip_init(struct sbi_scratch *scratch, bool cold_boot);

void sbi_irqchip_exit(struct sbi_scratch *scratch);

#endif
```

30.98 include/sbi/sbi_list.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_list.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_dlist

Macros

- #define SBI_LIST_POISON_PREV 0xDEADBEEF
- #define SBI_LIST_POISON_NEXT 0xFADEBABE
- #define SBI_LIST_HEAD_INIT(__lname) { &(__lname), &(__lname) }
- #define SBI_LIST_HEAD(_lname) struct sbi_dlist _lname = SBI_LIST_HEAD_INIT(_lname)
- #define SBI_INIT_LIST_HEAD(ptr)
- #define sbi_list_entry(ptr, type, member) container_of(ptr, type, member)
- #define sbi_list_first_entry(ptr, type, member) sbi_list_entry((ptr)->next, type, member)
- #define sbi_list_last_entry(ptr, type, member) sbi_list_entry((ptr)->prev, type, member)
- #define sbi_list_for_each(pos, head) for (pos = (head)->next; pos != (head); pos = pos->next)
- #define sbi_list_for_each_entry(pos, head, member)

Functions

- static void __sbi_list_add (struct sbi_dlist *new, struct sbi_dlist *prev, struct sbi_dlist *next)
- static bool sbi_list_empty (struct sbi_dlist *head)
- static void sbi_list_add (struct sbi_dlist *new, struct sbi_dlist *head)
- static void sbi_list_add_tail (struct sbi_dlist *new, struct sbi_dlist *tnode)
- static void __sbi_list_del (struct sbi_dlist *prev, struct sbi_dlist *next)
- static void __sbi_list_del_entry (struct sbi_dlist *entry)
- static void sbi_list_del (struct sbi_dlist *entry)
- static void sbi_list_del_init (struct sbi_dlist *entry)

30.98.1 Macro Definition Documentation

30.98.1.1 SBI_INIT_LIST_HEAD

```
#define SBI_INIT_LIST_HEAD(
            ptr )
```

Value:

```
do { \
    (ptr)->next = ptr; (ptr)->prev = ptr; \
} while (0)
```

30.98.1.2 sbi_list_entry

```
#define sbi_list_entry(
            ptr,
            type,
            member )  container_of(ptr, type, member)
```

Get the struct for this entry

Parameters

ptr	the &struct list_head pointer.
type	the type of the struct this is embedded in.
member	the name of the list_struct within the struct.

30.98.1.3 sbi_list_first_entry

```
#define sbi_list_first_entry(
            ptr,
            type,
            member )  sbi_list_entry((ptr)->next, type, member)
```

Get the first element from a list

Parameters

ptr	the list head to take the element from.
type	the type of the struct this is embedded in.
member	the name of the list_struct within the struct.

Note: that list is expected to be not empty.

30.98.1.4 sbi_list_for_each

```
#define sbi_list_for_each(
            pos,
            head )  for (pos = (head)->next; pos != (head); pos = pos->next)
```

Iterate over a list

Parameters

pos	the &struct list_head to use as a loop cursor.
head	the head for your list.

30.98.1.5 sbi_list_for_each_entry

```
#define sbi_list_for_each_entry(
            pos,
```

```
                head,
                member )
```

Value:

```
for (pos = sbi_list_entry((head)->next, typeof(*pos), member);     \
     &pos->member != (head);     \
     pos = sbi_list_entry(pos->member.next, typeof(*pos), member))
```

Iterate over list of given type

Parameters

pos	the type * to use as a loop cursor.
head	the head for your list.
member	the name of the list_struct within the struct.

30.98.1.6 SBI_LIST_HEAD

```
#define SBI_LIST_HEAD(
                _lname ) struct sbi_dlist _lname = SBI_LIST_HEAD_INIT(_lname)
```

30.98.1.7 SBI_LIST_HEAD_INIT

```
#define SBI_LIST_HEAD_INIT(
                __lname ) { &(__lname), &(__lname) }
```

30.98.1.8 sbi_list_last_entry

```
#define sbi_list_last_entry(
                ptr,
                type,
                member )  sbi_list_entry((ptr)->prev, type, member)
```

Get the last element from a list

Parameters

ptr	the list head to take the element from.
type	the type of the struct this is embedded in.
member	the name of the list_head within the struct.

Note: that list is expected to be not empty.

30.98.1.9 SBI_LIST_POISON_NEXT

```
#define SBI_LIST_POISON_NEXT 0xFADEBABE
```

30.98.1.10 SBI_LIST_POISON_PREV

```
#define SBI_LIST_POISON_PREV 0xDEADBEEF
```

30.98.2 Function Documentation

30.98.2.1 __sbi_list_add()

```
static void __sbi_list_add (
            struct sbi_dlist * new,
            struct sbi_dlist * prev,
            struct sbi_dlist * next )  [inline], [static]
```

Here is the caller graph for this function:

30.98.2.2 __sbi_list_del()

```
static void __sbi_list_del (
            struct sbi_dlist * prev,
            struct sbi_dlist * next )  [inline], [static]
```

Here is the caller graph for this function:

30.98.2.3 __sbi_list_del_entry()

```
static void __sbi_list_del_entry (
            struct sbi_dlist * entry )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.98.2.4 sbi_list_add()

```
static void sbi_list_add (
            struct sbi_dlist * new,
            struct sbi_dlist * head )  [inline], [static]
```

Adds the new node after the given head.

Parameters

new	New node that needs to be added to list.
head	List head after which the "new" node should be added. Note: the new node is added after the head.

Here is the call graph for this function:

Here is the caller graph for this function:

30.98.2.5 sbi_list_add_tail()

```
static void sbi_list_add_tail (
            struct sbi_dlist * new,
            struct sbi_dlist * tnode )  [inline], [static]
```

Adds a node at the tail where tnode points to tail node.

Parameters

new	The new node to be added before tail.
tnode	The current tail node. Note: the new node is added before tail node.

Here is the call graph for this function:

Here is the caller graph for this function:

30.98.2.6 sbi_list_del()

```
static void sbi_list_del (
            struct sbi_dlist * entry )  [inline], [static]
```

Deletes a given entry from list.

Parameters

node	Node to be deleted.

Here is the call graph for this function:

Here is the caller graph for this function:

30.98.2.7 sbi_list_del_init()

```
static void sbi_list_del_init (
            struct sbi_dlist * entry )  [inline], [static]
```

Deletes entry from list and reinitialize it.

Parameters

entry	the element to delete from the list.

Here is the call graph for this function:

Here is the caller graph for this function:

30.98.2.8 sbi_list_empty()

```
static bool sbi_list_empty (
            struct sbi_dlist * head )  [inline], [static]
```

Checks if the list is empty or not.

Parameters

head	List head

Returns true if list is empty, false otherwise. Here is the caller graph for this function:

30.99 sbi_list.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Simple doubly-linked list library.
 *
 * Adapted from Xvisor source file libs/include/libs/list.h
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
```

```
 */

#ifndef __SBI_LIST_H__
#define __SBI_LIST_H__

#include <sbi/sbi_types.h>

#define SBI_LIST_POISON_PREV    0xDEADBEEF
#define SBI_LIST_POISON_NEXT    0xFADEBABE

struct sbi_dlist {
    struct sbi_dlist *next, *prev;
};

#define SBI_LIST_HEAD_INIT(__lname) { &(__lname), &(__lname) }

#define SBI_LIST_HEAD(_lname)   \
struct sbi_dlist _lname = SBI_LIST_HEAD_INIT(_lname)

#define SBI_INIT_LIST_HEAD(ptr) \
do { \
    (ptr)->next = ptr; (ptr)->prev = ptr; \
} while (0)

static inline void __sbi_list_add(struct sbi_dlist *new,
                  struct sbi_dlist *prev,
                  struct sbi_dlist *next)
{
    new->prev = prev;
    new->next = next;
    prev->next = new;
    next->prev = new;
}

static inline bool sbi_list_empty(struct sbi_dlist *head)
{
    return head->next == head;
}

static inline void sbi_list_add(struct sbi_dlist *new, struct sbi_dlist *head)
{
    __sbi_list_add(new, head, head->next);
}

static inline void sbi_list_add_tail(struct sbi_dlist *new,
                     struct sbi_dlist *tnode)
{
    __sbi_list_add(new, tnode->prev, tnode);
}

static inline void __sbi_list_del(struct sbi_dlist *prev,
                  struct sbi_dlist *next)
{
    prev->next = next;
    next->prev = prev;
}

static inline void __sbi_list_del_entry(struct sbi_dlist *entry)
{
    __sbi_list_del(entry->prev, entry->next);
}

static inline void sbi_list_del(struct sbi_dlist *entry)
{
    __sbi_list_del(entry->prev, entry->next);
    entry->next = (void *)SBI_LIST_POISON_NEXT;
    entry->prev = (void *)SBI_LIST_POISON_PREV;
}

static inline void sbi_list_del_init(struct sbi_dlist *entry)
{
    __sbi_list_del_entry(entry);
    SBI_INIT_LIST_HEAD(entry);
}

#define sbi_list_entry(ptr, type, member) \
    container_of(ptr, type, member)

#define sbi_list_first_entry(ptr, type, member) \
    sbi_list_entry((ptr)->next, type, member)

#define sbi_list_last_entry(ptr, type, member) \
    sbi_list_entry((ptr)->prev, type, member)

#define sbi_list_for_each(pos, head) \
    for (pos = (head)->next; pos != (head); pos = pos->next)

```

```
#define sbi_list_for_each_entry(pos, head, member) \
    for (pos = sbi_list_entry((head)->next, typeof(*pos), member);  \
         &pos->member != (head);    \
         pos = sbi_list_entry(pos->member.next, typeof(*pos), member))

#endif
```

30.100 include/sbi/sbi_math.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- unsigned long log2roundup (unsigned long x)

30.100.1 Function Documentation

30.100.1.1 log2roundup()

```
unsigned long log2roundup (
            unsigned long x )
```

Here is the caller graph for this function:

30.101 sbi_math.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Atish Patra <atish.patra@wdc.com>
 */

#ifndef __SBI_MATH_H__
#define __SBI_MATH_H__

unsigned long log2roundup(unsigned long x);

#endif
```

30.102 include/sbi/sbi_mpxy.h File Reference

`#include <sbi/sbi_list.h>`
Include dependency graph for sbi_mpxy.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_mpxy_msi_info
- struct sbi_mpxy_channel_attrs
- struct sbi_mpxy_channel

Macros

- #define SBI_MPXY_MSGPROTO_VERSION(Major, Minor) ((Major << 16) | Minor)

Enumerations

- enum sbi_mpxy_attr_id {
  SBI_MPXY_ATTR_MSG_PROT_ID = 0x00000000 , SBI_MPXY_ATTR_MSG_PROT_VER = 0x00000001 , SBI_MPXY_ATTR_MSG_MAX_LEN = 0x00000002 , SBI_MPXY_ATTR_MSG_SEND_TIMEOUT = 0x00000003 ,
  SBI_MPXY_ATTR_MSG_COMPLETION_TIMEOUT = 0x00000004 , SBI_MPXY_ATTR_CHANNEL_CAPABILITY = 0x00000005 , SBI_MPXY_ATTR_SSE_EVENT_ID = 0x00000006 , SBI_MPXY_ATTR_MSI_CONTROL = 0x00000007 ,
  SBI_MPXY_ATTR_MSI_ADDR_LO = 0x00000008 , SBI_MPXY_ATTR_MSI_ADDR_HI = 0x00000009 , SBI_MPXY_ATTR_MSI_DATA = 0x0000000A , SBI_MPXY_ATTR_EVENTS_STATE_CONTROL = 0x0000000B ,

SBI_MPXY_ATTR_STD_ATTR_MAX_IDX , SBI_MPXY_ATTR_MSGPROTO_ATTR_START = 0x80000000 , SBI_MPXY_ATTR_MSGPROTO_ATTR_END = 0xffffffff }
- enum sbi_mpxy_msgproto_id { SBI_MPXY_MSGPROTO_RPMI_ID = 0x00000000 , SBI_MPXY_MSGPROTO_MAX_IDX , SBI_MPXY_MSGPROTO_VENDOR_START = 0x80000000 , SBI_MPXY_MSGPROTO_VENDOR_END = 0xffffffff }
- enum SBI_EXT_MPXY_SHMEM_FLAGS { SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE = 0b00 , SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE_RETURN = 0b01 , SBI_EXT_MPXY_SHMEM_FLAG_MAX_IDX }

Functions

- int sbi_mpxy_register_channel (struct sbi_mpxy_channel ∗channel)
- int sbi_mpxy_init (struct sbi_scratch ∗scratch)
- bool sbi_mpxy_channel_available (void)
- int sbi_mpxy_set_shmem (unsigned long shmem_size, unsigned long shmem_phys_lo, unsigned long shmem_phys_hi, unsigned long flags)
- int sbi_mpxy_get_channel_ids (u32 start_index)
- int sbi_mpxy_read_attrs (u32 channel_id, u32 base_attr_id, u32 attr_count)
- int sbi_mpxy_write_attrs (u32 channel_id, u32 base_attr_id, u32 attr_count)
- int sbi_mpxy_send_message (u32 channel_id, u8 msg_id, unsigned long msg_data_len, unsigned long ∗resp_data_len)
- int sbi_mpxy_get_notification_events (u32 channel_id, unsigned long ∗events_len)

30.102.1 Macro Definition Documentation

30.102.1.1 SBI_MPXY_MSGPROTO_VERSION

```
#define SBI_MPXY_MSGPROTO_VERSION(
            Major,
            Minor ) ((Major << 16) | Minor)
```

30.102.2 Enumeration Type Documentation

30.102.2.1 SBI_EXT_MPXY_SHMEM_FLAGS

```
enum SBI_EXT_MPXY_SHMEM_FLAGS
```

Enumerator

SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE	
SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE_RETURN	
SBI_EXT_MPXY_SHMEM_FLAG_MAX_IDX	

30.102.2.2 sbi_mpxy_attr_id

```
enum sbi_mpxy_attr_id
```

Enumerator

SBI_MPXY_ATTR_MSG_PROT_ID	
SBI_MPXY_ATTR_MSG_PROT_VER	
SBI_MPXY_ATTR_MSG_MAX_LEN	
SBI_MPXY_ATTR_MSG_SEND_TIMEOUT	
SBI_MPXY_ATTR_MSG_COMPLETION_TIMEOUT	
SBI_MPXY_ATTR_CHANNEL_CAPABILITY	
SBI_MPXY_ATTR_SSE_EVENT_ID	
SBI_MPXY_ATTR_MSI_CONTROL	

Enumerator

SBI_MPXY_ATTR_MSI_ADDR_LO	
SBI_MPXY_ATTR_MSI_ADDR_HI	
SBI_MPXY_ATTR_MSI_DATA	
SBI_MPXY_ATTR_EVENTS_STATE_CONTROL	
SBI_MPXY_ATTR_STD_ATTR_MAX_IDX	
SBI_MPXY_ATTR_MSGPROTO_ATTR_START	
SBI_MPXY_ATTR_MSGPROTO_ATTR_END	

30.102.2.3 sbi_mpxy_msgproto_id

`enum sbi_mpxy_msgproto_id`

SBI MPXY Message Protocol IDs

Enumerator

SBI_MPXY_MSGPROTO_RPMI_ID	
SBI_MPXY_MSGPROTO_MAX_IDX	
SBI_MPXY_MSGPROTO_VENDOR_START	Vendor specific message protocol IDs
SBI_MPXY_MSGPROTO_VENDOR_END	

30.102.3 Function Documentation

30.102.3.1 sbi_mpxy_channel_available()

```
bool sbi_mpxy_channel_available (
            void  )
```

Check if some Message proxy channel is available

Check if any channel is registered with mpxy framework Here is the call graph for this function:

Here is the caller graph for this function:

30.102.3.2 sbi_mpxy_get_channel_ids()

```
int sbi_mpxy_get_channel_ids (
            u32 start_index )
```

Get channel IDs list number of channel ids which can be stored in shmem adjusting for remaining and returned fieldsHere is the call graph for this function:

Here is the caller graph for this function:

30.102.3.3 sbi_mpxy_get_notification_events()

```
int sbi_mpxy_get_notification_events (
            u32 channel_id,
            unsigned long * events_len )
```

Get Message proxy notification events Here is the call graph for this function:

Here is the caller graph for this function:

30.102.3.4 sbi_mpxy_init()

```
int sbi_mpxy_init (
            struct sbi_scratch * scratch )
```

Initialize Message proxy subsystem TODO: Proper support for checking msi support from platform. Currently disable msi and sse and use pollingHere is the call graph for this function:

Here is the caller graph for this function:

30.102.3.5 sbi_mpxy_read_attrs()

```
int sbi_mpxy_read_attrs (
            u32 channel_id,
            u32 base_attr_id,
            u32 attr_count )
```

Read MPXY channel attributes Even if the message protocol driver does not provide read attribute callback, return bad range error instead of not supported to let client distinguish it from channel id not supported. Check the complate range supported for message protocol attributes. Actual supported attributes will be checked by the message protocol driver.

Function expected to return the SBI supported errors At this point both base attribute id and only the mpxy supported range been verified. Platform callback must check if the range requested is supported by message protocol driverHere is the call graph for this function:

Here is the caller graph for this function:

30.102.3.6 sbi_mpxy_register_channel()

```
int sbi_mpxy_register_channel (
            struct sbi_mpxy_channel * channel )
```

Register a Message proxy channel

Register a channel with MPXY framework. Called by message protocol drivers Here is the call graph for this function:

Here is the caller graph for this function:

30.102.3.7 sbi_mpxy_send_message()

```
int sbi_mpxy_send_message (
            u32 channel_id,
            u8 msg_id,
            unsigned long msg_data_len,
            unsigned long * resp_data_len )
```

Send a message over a MPXY channel. In case if response is not expected, resp_data_len will be NULL. Here is

the call graph for this function:

Here is the caller graph for this function:

30.102.3.8 sbi_mpxy_set_shmem()

```
int sbi_mpxy_set_shmem (
            unsigned long shmem_size,
            unsigned long shmem_phys_lo,
            unsigned long shmem_phys_hi,
            unsigned long flags )
```

Set Message proxy shared memory on the calling HART Disable shared memory if both hi and lo have all bit 1s
Check shared memory size and address aligned to 4K Page
Save the current shmem details in new shmem region
Setup the new shared memoryHere is the call graph for this function:

Here is the caller graph for this function:

30.102.3.9 sbi_mpxy_write_attrs()

```
int sbi_mpxy_write_attrs (
            u32 channel_id,
```

```
u32 base_attr_id,
u32 attr_count )
```

Write MPXY channel attributes Verify the attribute ids range and values

Message protocol specific attributes: If attributes belong to message protocol, they are simply passed to the message protocol driver callback after checking the valid range. Attributes contiguous range & permission & other checks are done by the mpxy and message protocol glue layer.

Even if the message protocol driver does not provide write attribute callback, return bad range error instead of not supported to let client distinguish it from channel id not supported.

Function expected to return the SBI supported errors At this point both base attribute id and only the mpxy supported range been verified. Platform callback must check if the range requested is supported by message protocol driverHere is the call graph for this function:

Here is the caller graph for this function:

30.103 sbi_mpxy.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2024 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Rahul Pathak <rpathak@ventanamicro.com>
 */

#ifndef __SBI_MPXY_H__
#define __SBI_MPXY_H__

#include <sbi/sbi_list.h>

struct sbi_scratch;

#define SBI_MPXY_MSGPROTO_VERSION(Major, Minor) ((Major « 16) | Minor)

enum sbi_mpxy_attr_id {
    /* Standard channel attributes managed by MPXY framework */
    SBI_MPXY_ATTR_MSG_PROT_ID      = 0x00000000,
    SBI_MPXY_ATTR_MSG_PROT_VER        = 0x00000001,
    SBI_MPXY_ATTR_MSG_MAX_LEN      = 0x00000002,
    SBI_MPXY_ATTR_MSG_SEND_TIMEOUT        = 0x00000003,
    SBI_MPXY_ATTR_MSG_COMPLETION_TIMEOUT    = 0x00000004,
    SBI_MPXY_ATTR_CHANNEL_CAPABILITY    = 0x00000005,
    SBI_MPXY_ATTR_SSE_EVENT_ID        = 0x00000006,
    SBI_MPXY_ATTR_MSI_CONTROL      = 0x00000007,
    SBI_MPXY_ATTR_MSI_ADDR_LO      = 0x00000008,
    SBI_MPXY_ATTR_MSI_ADDR_HI      = 0x00000009,
    SBI_MPXY_ATTR_MSI_DATA            = 0x0000000A,
    SBI_MPXY_ATTR_EVENTS_STATE_CONTROL    = 0x0000000B,
    SBI_MPXY_ATTR_STD_ATTR_MAX_IDX,
```

```
    /* Message protocol specific attributes, managed by
     * message protocol driver */
    SBI_MPXY_ATTR_MSGPROTO_ATTR_START  = 0x80000000,
    SBI_MPXY_ATTR_MSGPROTO_ATTR_END      = 0xffffffff
};

enum sbi_mpxy_msgproto_id {
    SBI_MPXY_MSGPROTO_RPMI_ID = 0x00000000,
    SBI_MPXY_MSGPROTO_MAX_IDX,
    SBI_MPXY_MSGPROTO_VENDOR_START    = 0x80000000,
    SBI_MPXY_MSGPROTO_VENDOR_END    = 0xffffffff
};

enum SBI_EXT_MPXY_SHMEM_FLAGS {
    SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE      = 0b00,
    SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE_RETURN    = 0b01,
    SBI_EXT_MPXY_SHMEM_FLAG_MAX_IDX
};

struct sbi_mpxy_msi_info {
    /* MSI target address low 32-bit */
    u32 msi_addr_lo;
    /* MSI target address high 32-bit */
    u32 msi_addr_hi;
    /* MSI data */
    u32 msi_data;
};

struct sbi_mpxy_channel_attrs {
    /* Message protocol ID */
    u32 msg_proto_id;
    /* Message protocol Version */
    u32 msg_proto_version;
    /* Message protocol maximum message data length(bytes) */
    u32 msg_data_maxlen;
    /* Message protocol message send timeout
     * in microseconds */
    u32 msg_send_timeout;
    /* Message protocol message response timeout in
     * microseconds. Its the aggregate of msg_send_timeout
     * and the timeout in receiving the response */
    u32 msg_completion_timeout;
    /* Bit array for channel capabilities */
    u32 capability;
    u32 sse_event_id;
    u32 msi_control;
    struct sbi_mpxy_msi_info msi_info;
    /* Events State Control */
    u32 eventsstate_ctrl;
};

struct sbi_mpxy_channel {
    struct sbi_dlist head;
    u32 channel_id;
    struct sbi_mpxy_channel_attrs attrs;

    int (*read_attributes)(struct sbi_mpxy_channel *channel,
                u32 *outmem,
                u32 base_attr_id,
                u32 attr_count);

    int (*write_attributes)(struct sbi_mpxy_channel *channel,
                u32 *inmem,
                u32 base_attr_id,
                u32 attr_count);
    int (*send_message_with_response)(struct sbi_mpxy_channel *channel,
                u32 msg_id, void *msgbuf, u32 msg_len,
                void *respbuf, u32 resp_max_len,
                unsigned long *resp_len);

    int (*send_message_without_response)(struct sbi_mpxy_channel *channel,
                u32 msg_id, void *msgbuf, u32 msg_len);

    int (*get_notification_events)(struct sbi_mpxy_channel *channel,
                    void *eventsbuf, u32 bufsize,
                    unsigned long *events_len);

    void (*switch_eventsstate)(u32 enable);
};

int sbi_mpxy_register_channel(struct sbi_mpxy_channel *channel);

int sbi_mpxy_init(struct sbi_scratch *scratch);

bool sbi_mpxy_channel_available(void);

int sbi_mpxy_set_shmem(unsigned long shmem_size,
```

```
            unsigned long shmem_phys_lo,
            unsigned long shmem_phys_hi,
            unsigned long flags);

int sbi_mpxy_get_channel_ids(u32 start_index);

int sbi_mpxy_read_attrs(u32 channel_id, u32 base_attr_id, u32 attr_count);

int sbi_mpxy_write_attrs(u32 channel_id, u32 base_attr_id, u32 attr_count);

int sbi_mpxy_send_message(u32 channel_id, u8 msg_id,
                    unsigned long msg_data_len,
                    unsigned long *resp_data_len);

int sbi_mpxy_get_notification_events(u32 channel_id,
                    unsigned long *events_len);

#endif
```

30.104 include/sbi/sbi_platform.h File Reference

```
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_version.h>
#include <sbi/sbi_trap_ldst.h>
```

Include dependency graph for sbi_platform.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_platform_operations
- struct sbi_platform

Macros

- #define SBI_PLATFORM_VERSION(Major, Minor) ((Major << 16) | Minor)
- #define SBI_PLATFORM_OPENSBI_VERSION_OFFSET (0x00)
- #define SBI_PLATFORM_VERSION_OFFSET (0x04)
- #define SBI_PLATFORM_NAME_OFFSET (0x08)
- #define SBI_PLATFORM_FEATURES_OFFSET (0x48)
- #define SBI_PLATFORM_HART_COUNT_OFFSET (0x50)
- #define SBI_PLATFORM_HART_STACK_SIZE_OFFSET (0x54)
- #define SBI_PLATFORM_HEAP_SIZE_OFFSET (0x58)
- #define SBI_PLATFORM_RESERVED_OFFSET (0x5c)
- #define SBI_PLATFORM_OPS_OFFSET (0x60)
- #define SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET (0x60 + __SIZEOF_POINTER__)
- #define SBI_PLATFORM_HART_INDEX2ID_OFFSET (0x60 + (__SIZEOF_POINTER__ * 2))
- #define SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT (1UL << 12)
- #define SBI_PLATFORM_DEFAULT_FEATURES (SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
- #define SBI_PLATFORM_DEFAULT_HART_STACK_SIZE 8192
- #define SBI_PLATFORM_DEFAULT_HEAP_SIZE(__num_hart) (0x8000 + 0x1000 * (__num_hart))
- #define sbi_platform_ptr(__s) ((const struct sbi_platform *)((__s)->platform_addr))
- #define sbi_platform_thishart_ptr()
- #define sbi_platform_ops(__p) ((const struct sbi_platform_operations *)(__p)->platform_ops_addr)
- #define sbi_platform_has_mfaults_delegation(__p) ((__p)->features & SBI_PLATFORM_HAS_MFAULTS_DELEGATION)

Enumerations

- enum sbi_platform_features { SBI_PLATFORM_HAS_MFAULTS_DELEGATION = (1 << 1) , SBI_PLATFORM_HAS_LAST_FE
  = SBI_PLATFORM_HAS_MFAULTS_DELEGATION }

Functions

- void sbi_platform_get_features_str (const struct sbi_platform *plat, char *features_str, int nfstr)
- static const char * sbi_platform_name (const struct sbi_platform *plat)
- static unsigned long sbi_platform_get_features (const struct sbi_platform *plat)
- static u64 sbi_platform_tlbr_flush_limit (const struct sbi_platform *plat)
- static u32 sbi_platform_tlb_fifo_num_entries (const struct sbi_platform *plat)
- static u32 sbi_platform_hart_count (const struct sbi_platform *plat)
- static u32 sbi_platform_hart_stack_size (const struct sbi_platform *plat)
- static bool sbi_platform_cold_boot_allowed (const struct sbi_platform *plat, u32 hartid)
- static int sbi_platform_nascent_init (const struct sbi_platform *plat)
- static int sbi_platform_early_init (const struct sbi_platform *plat, bool cold_boot)
- static int sbi_platform_final_init (const struct sbi_platform *plat, bool cold_boot)
- static void sbi_platform_early_exit (const struct sbi_platform *plat)
- static void sbi_platform_final_exit (const struct sbi_platform *plat)
- static int sbi_platform_misa_extension (const struct sbi_platform *plat, char ext)
- static int sbi_platform_misa_xlen (const struct sbi_platform *plat)
- static int sbi_platform_extensions_init (const struct sbi_platform *plat, struct sbi_hart_features *hfeatures)
- static int sbi_platform_domains_init (const struct sbi_platform *plat)
- static int sbi_platform_pmu_init (const struct sbi_platform *plat)
- static uint64_t sbi_platform_pmu_xlate_to_mhpmevent (const struct sbi_platform *plat, uint32_t event_idx, uint64_t data)
- static int sbi_platform_irqchip_init (const struct sbi_platform *plat)
- static int sbi_platform_ipi_init (const struct sbi_platform *plat)
- static int sbi_platform_timer_init (const struct sbi_platform *plat)
- static int sbi_platform_mpxy_init (const struct sbi_platform *plat)
- static bool sbi_platform_vendor_ext_check (const struct sbi_platform *plat)

- static int sbi_platform_vendor_ext_provider (const struct sbi_platform *plat, long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_platform_emulate_load (const struct sbi_platform *plat, int rlen, unsigned long addr, union sbi_ldst_data *out_val)
- static int sbi_platform_emulate_store (const struct sbi_platform *plat, int wlen, unsigned long addr, union sbi_ldst_data in_val)

30.104.1 Macro Definition Documentation

30.104.1.1 SBI_PLATFORM_DEFAULT_FEATURES

```
#define SBI_PLATFORM_DEFAULT_FEATURES  (SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
```

Default feature set for a platform

30.104.1.2 SBI_PLATFORM_DEFAULT_HART_STACK_SIZE

```
#define SBI_PLATFORM_DEFAULT_HART_STACK_SIZE 8192
```

Platform default per-HART stack size for exception/interrupt handling

30.104.1.3 SBI_PLATFORM_DEFAULT_HEAP_SIZE

```
#define SBI_PLATFORM_DEFAULT_HEAP_SIZE(
            __num_hart )  (0x8000 + 0x1000 * (__num_hart))
```

Platform default heap size

30.104.1.4 SBI_PLATFORM_FEATURES_OFFSET

```
#define SBI_PLATFORM_FEATURES_OFFSET (0x48)
```

Offset of features in struct sbi_platform

30.104.1.5 SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET

```
#define SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET (0x60 + __SIZEOF_POINTER__)
```

Offset of firmware_context in struct sbi_platform

30.104.1.6 SBI_PLATFORM_HART_COUNT_OFFSET

```
#define SBI_PLATFORM_HART_COUNT_OFFSET (0x50)
```

Offset of hart_count in struct sbi_platform

30.104.1.7 SBI_PLATFORM_HART_INDEX2ID_OFFSET

```
#define SBI_PLATFORM_HART_INDEX2ID_OFFSET (0x60 + (__SIZEOF_POINTER__ * 2))
```

Offset of hart_index2id in struct sbi_platform

30.104.1.8 SBI_PLATFORM_HART_STACK_SIZE_OFFSET

```
#define SBI_PLATFORM_HART_STACK_SIZE_OFFSET (0x54)
```

Offset of hart_stack_size in struct sbi_platform

30.104.1.9 sbi_platform_has_mfaults_delegation

```
#define sbi_platform_has_mfaults_delegation(
            __p )  ((__p)->features & SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
```

Check whether the platform supports fault delegation

30.104.1.10 SBI_PLATFORM_HEAP_SIZE_OFFSET

```
#define SBI_PLATFORM_HEAP_SIZE_OFFSET (0x58)
```

Offset of heap_size in struct sbi_platform

30.104.1.11 SBI_PLATFORM_NAME_OFFSET

```
#define SBI_PLATFORM_NAME_OFFSET (0x08)
```

Offset of name in struct sbi_platform

30.104.1.12 SBI_PLATFORM_OPENSBI_VERSION_OFFSET

```
#define SBI_PLATFORM_OPENSBI_VERSION_OFFSET (0x00)
```

Offset of opensbi_version in struct sbi_platform

30.104.1.13 sbi_platform_ops

```
#define sbi_platform_ops(
            __p )  ((const struct sbi_platform_operations *)(__p)->platform_ops_addr)
```

Get pointer to platform_ops_addr from platform pointer

30.104.1.14 SBI_PLATFORM_OPS_OFFSET

```
#define SBI_PLATFORM_OPS_OFFSET (0x60)
```

Offset of platform_ops_addr in struct sbi_platform

30.104.1.15 sbi_platform_ptr

```
#define sbi_platform_ptr(
            __s )  ((const struct sbi_platform *)((__s)->platform_addr))
```

Prevent modification of struct sbi_platform from affecting SBI_PLATFORM_xxx_OFFSET Get pointer to sbi_platform for sbi_scratch pointer

30.104.1.16 SBI_PLATFORM_RESERVED_OFFSET

```
#define SBI_PLATFORM_RESERVED_OFFSET (0x5c)
```

Offset of reserved in struct sbi_platform

30.104.1.17 sbi_platform_thishart_ptr

```
#define sbi_platform_thishart_ptr( )
```

Value:

```
((const struct sbi_platform *) \
(sbi_scratch_thishart_ptr()->platform_addr))
```

Get pointer to sbi_platform for current HART

30.104.1.18 SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT

```
#define SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT (1UL << 12)
```

30.104.1.19 SBI_PLATFORM_VERSION

```
#define SBI_PLATFORM_VERSION(
            Major,
            Minor ) ((Major << 16) | Minor)
```

OpenSBI 32-bit platform version with:

1. upper 16-bits as major number
2. lower 16-bits as minor number

30.104.1.20 SBI_PLATFORM_VERSION_OFFSET

```
#define SBI_PLATFORM_VERSION_OFFSET (0x04)
```

Offset of platform_version in struct sbi_platform

30.104.2 Enumeration Type Documentation

30.104.2.1 sbi_platform_features

```
enum sbi_platform_features
```

Possible feature flags of a platform

Enumerator

SBI_PLATFORM_HAS_MFAULTS_DELEGATION	Platform has fault delegation support
SBI_PLATFORM_HAS_LAST_FEATURE	Last index of Platform features

30.104.3 Function Documentation

30.104.3.1 sbi_platform_cold_boot_allowed()

```
static bool sbi_platform_cold_boot_allowed (
            const struct sbi_platform * plat,
            u32 hartid )  [inline], [static]
```

Check whether given HART is allowed to do cold boot

Parameters

plat	pointer to struct sbi_platform
hartid	HART ID

Returns

true if HART is allowed to do cold boot and false otherwise

Here is the caller graph for this function:

30.104.3.2 sbi_platform_domains_init()

```
static int sbi_platform_domains_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Initialize (or populate) domains for the platform

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.3 sbi_platform_early_exit()

```
static void sbi_platform_early_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Early exit for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:

30.104.3.4 sbi_platform_early_init()

```
static int sbi_platform_early_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Early initialization for current HART

Parameters

plat	pointer to struct sbi_platform
cold_boot	whether cold boot (true) or warm_boot (false)

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.5 sbi_platform_emulate_load()

```
static int sbi_platform_emulate_load (
            const struct sbi_platform * plat,
            int rlen,
            unsigned long addr,
            union sbi_ldst_data * out_val )  [inline], [static]
```

Ask platform to emulate the trapped load

Parameters

plat	pointer to struct sbi_platform
rlen	length of the load: 1/2/4/8...
addr	virtual address of the load. Platform needs to page-walk and find the physical address if necessary
out_val	value loaded

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.6 sbi_platform_emulate_store()

```
static int sbi_platform_emulate_store (
            const struct sbi_platform * plat,
            int wlen,
            unsigned long addr,
            union sbi_ldst_data in_val )  [inline], [static]
```

Ask platform to emulate the trapped store

Parameters

plat	pointer to struct sbi_platform
wlen	length of the store: 1/2/4/8...
addr	virtual address of the store. Platform needs to page-walk and find the physical address if necessary
in_val	value to store

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.7 sbi_platform_extensions_init()

```
static int sbi_platform_extensions_init (
            const struct sbi_platform * plat,
            struct sbi_hart_features * hfeatures )  [inline], [static]
```

Initialize (or populate) HART extensions for the platform

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.8 sbi_platform_final_exit()

```
static void sbi_platform_final_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Final exit for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:

30.104.3.9 sbi_platform_final_init()

```
static int sbi_platform_final_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Final initialization for current HART

Parameters

plat	pointer to struct sbi_platform
cold_boot	whether cold boot (true) or warm_boot (false)

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.10 sbi_platform_get_features()

```
static unsigned long sbi_platform_get_features (
            const struct sbi_platform * plat )  [inline], [static]
```

Get the platform features

Parameters

plat	pointer to struct sbi_platform

Returns

the features value currently set for the given platform

Here is the caller graph for this function:

30.104.3.11 sbi_platform_get_features_str()

```
void sbi_platform_get_features_str (
            const struct sbi_platform * plat,
            char * features_str,
            int nfstr )
```

Get the platform features in string format

Parameters

plat	pointer to struct sbi_platform
features_str	pointer to a char array where the features string will be updated
nfstr	length of the features_str. The feature string will be truncated if nfstr is not long enough.

Here is the call graph for this function:

Here is the caller graph for this function:

30.104.3.12 sbi_platform_hart_count()

```
static u32 sbi_platform_hart_count (
            const struct sbi_platform * plat )  [inline], [static]
```

Get total number of HARTs supported by the platform

Parameters

plat	pointer to struct sbi_platform

Returns

total number of HARTs

Here is the caller graph for this function:

30.104.3.13 sbi_platform_hart_stack_size()

```
static u32 sbi_platform_hart_stack_size (
            const struct sbi_platform * plat ) [inline], [static]
```

Get per-HART stack size for exception/interrupt handling

Parameters

plat	pointer to struct sbi_platform

Returns

stack size in bytes

30.104.3.14 sbi_platform_ipi_init()

```
static int sbi_platform_ipi_init (
            const struct sbi_platform * plat ) [inline], [static]
```

Initialize the platform IPI support during cold boot

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.15 sbi_platform_irqchip_init()

```
static int sbi_platform_irqchip_init (
            const struct sbi_platform * plat ) [inline], [static]
```

Initialize the platform interrupt controller during cold boot

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.16 sbi_platform_misa_extension()

```
static int sbi_platform_misa_extension (
            const struct sbi_platform * plat,
            char ext )  [inline], [static]
```

Check CPU extension in MISA

Parameters

plat	pointer to struct sbi_platform
ext	shorthand letter for CPU extensions

Returns

zero for not-supported and non-zero for supported

Here is the caller graph for this function:

30.104.3.17 sbi_platform_misa_xlen()

```
static int sbi_platform_misa_xlen (
            const struct sbi_platform * plat )  [inline], [static]
```

Get MXL field of MISA

Parameters

plat	pointer to struct sbi_platform

Returns

1/2/3 on success and error code on failure

Here is the caller graph for this function:

30.104.3.18 sbi_platform_mpxy_init()

```
static int sbi_platform_mpxy_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Initialize the platform Message Proxy drivers

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.19 sbi_platform_name()

```
static const char * sbi_platform_name (
            const struct sbi_platform * plat )  [inline], [static]
```

Get name of the platform

Parameters

plat	pointer to struct sbi_platform

Returns

pointer to platform name on success and "Unknown" on failure

Here is the caller graph for this function:

30.104.3.20 sbi_platform_nascent_init()

```
static int sbi_platform_nascent_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Nascent (very early) initialization for current HART
NOTE: This function can be used to do very early initialization of platform specific per-HART CSRs and devices.

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.21 sbi_platform_pmu_init()

```
static int sbi_platform_pmu_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Setup hw PMU events for the platform

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.22 sbi_platform_pmu_xlate_to_mhpmevent()

```
static uint64_t sbi_platform_pmu_xlate_to_mhpmevent (
            const struct sbi_platform * plat,
            uint32_t event_idx,
            uint64_t data )  [inline], [static]
```

Get the value to be written in mhpmeventx for event_idx

Parameters

plat	pointer to struct sbi_platform
event_idx	ID of the PMU event
data	Additional configuration data passed from supervisor software

Returns

expected value by the platform or 0 if platform doesn't know about the event

Here is the caller graph for this function:

30.104.3.23 sbi_platform_timer_init()

```
static int sbi_platform_timer_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Initialize the platform timer during cold boot

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.104.3.24 sbi_platform_tlb_fifo_num_entries()

```
static u32 sbi_platform_tlb_fifo_num_entries (
            const struct sbi_platform * plat )  [inline], [static]
```

Get platform specific tlb fifo num entries.

Parameters

plat	pointer to struct sbi_platform

Returns

number of tlb fifo entries

Here is the caller graph for this function:

30.104.3.25 sbi_platform_tlbr_flush_limit()

```
static u64 sbi_platform_tlbr_flush_limit (
            const struct sbi_platform * plat )  [inline], [static]
```

Get platform specific tlb range flush maximum value. Any request with size higher than this is upgraded to a full flush.

Parameters

plat	pointer to struct sbi_platform

Returns

tlb range flush limit value. Returns a default (page size) if not defined by platform.

Here is the caller graph for this function:

30.104.3.26 sbi_platform_vendor_ext_check()

```
static bool sbi_platform_vendor_ext_check (
            const struct sbi_platform * plat )  [inline], [static]
```

Check if SBI vendor extension is implemented or not.

Parameters

plat	pointer to struct sbi_platform

Returns

false if not implemented and true if implemented

Here is the caller graph for this function:

30.104.3.27 sbi_platform_vendor_ext_provider()

```
static int sbi_platform_vendor_ext_provider (
            const struct sbi_platform * plat,
            long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [inline], [static]
```

Invoke platform specific vendor SBI extension implementation.

Parameters

plat	pointer to struct sbi_platform
funcid	SBI function id within the extension id
regs	pointer to trap registers passed by the caller
out_value	output value that can be filled by the callee
out_trap	trap info that can be filled by the callee

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.105 sbi_platform.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
```

```
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_PLATFORM_H__
#define __SBI_PLATFORM_H__

#define SBI_PLATFORM_VERSION(Major, Minor) ((Major << 16) | Minor)

#define SBI_PLATFORM_OPENSBI_VERSION_OFFSET (0x00)
#define SBI_PLATFORM_VERSION_OFFSET (0x04)
#define SBI_PLATFORM_NAME_OFFSET (0x08)
#define SBI_PLATFORM_FEATURES_OFFSET (0x48)
#define SBI_PLATFORM_HART_COUNT_OFFSET (0x50)
#define SBI_PLATFORM_HART_STACK_SIZE_OFFSET (0x54)
#define SBI_PLATFORM_HEAP_SIZE_OFFSET (0x58)
#define SBI_PLATFORM_RESERVED_OFFSET (0x5c)
#define SBI_PLATFORM_OPS_OFFSET (0x60)
#define SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET (0x60 + __SIZEOF_POINTER__)
#define SBI_PLATFORM_HART_INDEX2ID_OFFSET (0x60 + (__SIZEOF_POINTER__ * 2))

#define SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT      (1UL << 12)

#ifndef __ASSEMBLER__

#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_version.h>
#include <sbi/sbi_trap_ldst.h>

struct sbi_domain_memregion;
struct sbi_ecall_return;
struct sbi_trap_regs;
struct sbi_hart_features;
union sbi_ldst_data;

enum sbi_platform_features {
    SBI_PLATFORM_HAS_MFAULTS_DELEGATION = (1 << 1),

    SBI_PLATFORM_HAS_LAST_FEATURE = SBI_PLATFORM_HAS_MFAULTS_DELEGATION,
};

#define SBI_PLATFORM_DEFAULT_FEATURES                                   \
    (SBI_PLATFORM_HAS_MFAULTS_DELEGATION)

struct sbi_platform_operations {
    /* Check if specified HART is allowed to do cold boot */
    bool (*cold_boot_allowed)(u32 hartid);

    /* Platform nascent initialization */
    int (*nascent_init)(void);

    int (*early_init)(bool cold_boot);
    int (*final_init)(bool cold_boot);

    void (*early_exit)(void);
    void (*final_exit)(void);

    int (*misa_check_extension)(char ext);

    int (*misa_get_xlen)(void);

    int (*extensions_init)(struct sbi_hart_features *hfeatures);

    int (*domains_init)(void);

    int (*pmu_init)(void);

    uint64_t (*pmu_xlate_to_mhpmevent)(uint32_t event_idx, uint64_t data);

    int (*irqchip_init)(void);

    int (*ipi_init)(void);

    u64 (*get_tlbr_flush_limit)(void);

    u32 (*get_tlb_num_entries)(void);

    int (*timer_init)(void);

    int (*mpxy_init)(void);

    bool (*vendor_ext_check)(void);
    int (*vendor_ext_provider)(long funcid,
                    struct sbi_trap_regs *regs,
                    struct sbi_ecall_return *out);

```

```
    int (*emulate_load)(int rlen, unsigned long addr,
                union sbi_ldst_data *out_val);
    int (*emulate_store)(int wlen, unsigned long addr,
                 union sbi_ldst_data in_val);
};

#define SBI_PLATFORM_DEFAULT_HART_STACK_SIZE    8192

#define SBI_PLATFORM_DEFAULT_HEAP_SIZE(__num_hart)  \
                    (0x8000 + 0x1000 * (__num_hart))

struct sbi_platform {
    u32 opensbi_version;
    u32 platform_version;
    char name[64];
    u64 features;
    u32 hart_count;
    u32 hart_stack_size;
    u32 heap_size;
    u32 reserved;
    unsigned long platform_ops_addr;
    unsigned long firmware_context;
    const u32 *hart_index2id;
};

_Static_assert(
    offsetof(struct sbi_platform, opensbi_version)
        == SBI_PLATFORM_OPENSBI_VERSION_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_OPENSBI_VERSION_OFFSET");
_Static_assert(
    offsetof(struct sbi_platform, platform_version)
        == SBI_PLATFORM_VERSION_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_VERSION_OFFSET");
_Static_assert(
    offsetof(struct sbi_platform, name)
        == SBI_PLATFORM_NAME_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_NAME_OFFSET");
_Static_assert(
    offsetof(struct sbi_platform, features)
        == SBI_PLATFORM_FEATURES_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_FEATURES_OFFSET");
_Static_assert(
    offsetof(struct sbi_platform, hart_count)
        == SBI_PLATFORM_HART_COUNT_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_HART_COUNT_OFFSET");
_Static_assert(
    offsetof(struct sbi_platform, hart_stack_size)
        == SBI_PLATFORM_HART_STACK_SIZE_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_HART_STACK_SIZE_OFFSET");
_Static_assert(
    offsetof(struct sbi_platform, platform_ops_addr)
        == SBI_PLATFORM_OPS_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_OPS_OFFSET");
_Static_assert(
    offsetof(struct sbi_platform, firmware_context)
        == SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET");
_Static_assert(
    offsetof(struct sbi_platform, hart_index2id)
        == SBI_PLATFORM_HART_INDEX2ID_OFFSET,
    "struct sbi_platform definition has changed, please redefine "
    "SBI_PLATFORM_HART_INDEX2ID_OFFSET");

#define sbi_platform_ptr(__s) \
    ((const struct sbi_platform *)((__s)->platform_addr))
#define sbi_platform_thishart_ptr() ((const struct sbi_platform *) \
    (sbi_scratch_thishart_ptr()->platform_addr))
#define sbi_platform_ops(__p) \
    ((const struct sbi_platform_operations *)(__p)->platform_ops_addr)

#define sbi_platform_has_mfaults_delegation(__p) \
    ((__p)->features & SBI_PLATFORM_HAS_MFAULTS_DELEGATION)

void sbi_platform_get_features_str(const struct sbi_platform *plat,
                   char *features_str, int nfstr);

static inline const char *sbi_platform_name(const struct sbi_platform *plat)
{
    if (plat)
```

```
        return plat->name;
    return "Unknown";
}

static inline unsigned long sbi_platform_get_features(
                        const struct sbi_platform *plat)
{
    if (plat)
        return plat->features;
    return 0;
}

static inline u64 sbi_platform_tlbr_flush_limit(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->get_tlbr_flush_limit)
        return sbi_platform_ops(plat)->get_tlbr_flush_limit();
    return SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT;
}

static inline u32 sbi_platform_tlb_fifo_num_entries(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->get_tlb_num_entries)
        return sbi_platform_ops(plat)->get_tlb_num_entries();
    return sbi_scratch_last_hartindex() + 1;
}

static inline u32 sbi_platform_hart_count(const struct sbi_platform *plat)
{
    if (plat)
        return plat->hart_count;
    return 0;
}

static inline u32 sbi_platform_hart_stack_size(const struct sbi_platform *plat)
{
    if (plat)
        return plat->hart_stack_size;
    return 0;
}

static inline bool sbi_platform_cold_boot_allowed(
                    const struct sbi_platform *plat,
                    u32 hartid)
{
    if (plat && sbi_platform_ops(plat)->cold_boot_allowed)
        return sbi_platform_ops(plat)->cold_boot_allowed(hartid);
    return true;
}

static inline int sbi_platform_nascent_init(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->nascent_init)
        return sbi_platform_ops(plat)->nascent_init();
    return 0;
}

static inline int sbi_platform_early_init(const struct sbi_platform *plat,
                      bool cold_boot)
{
    if (plat && sbi_platform_ops(plat)->early_init)
        return sbi_platform_ops(plat)->early_init(cold_boot);
    return 0;
}

static inline int sbi_platform_final_init(const struct sbi_platform *plat,
                      bool cold_boot)
{
    if (plat && sbi_platform_ops(plat)->final_init)
        return sbi_platform_ops(plat)->final_init(cold_boot);
    return 0;
}

static inline void sbi_platform_early_exit(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->early_exit)
        sbi_platform_ops(plat)->early_exit();
}

static inline void sbi_platform_final_exit(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->final_exit)
        sbi_platform_ops(plat)->final_exit();
}

static inline int sbi_platform_misa_extension(const struct sbi_platform *plat,
                          char ext)
{
```

```
    if (plat && sbi_platform_ops(plat)->misa_check_extension)
        return sbi_platform_ops(plat)->misa_check_extension(ext);
    return 0;
}

static inline int sbi_platform_misa_xlen(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->misa_get_xlen)
        return sbi_platform_ops(plat)->misa_get_xlen();
    return -1;
}

static inline int sbi_platform_extensions_init(
                    const struct sbi_platform *plat,
                    struct sbi_hart_features *hfeatures)
{
    if (plat && sbi_platform_ops(plat)->extensions_init)
        return sbi_platform_ops(plat)->extensions_init(hfeatures);
    return 0;
}

static inline int sbi_platform_domains_init(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->domains_init)
        return sbi_platform_ops(plat)->domains_init();
    return 0;
}

static inline int sbi_platform_pmu_init(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->pmu_init)
        return sbi_platform_ops(plat)->pmu_init();
    return 0;
}

static inline uint64_t sbi_platform_pmu_xlate_to_mhpmevent(const struct sbi_platform *plat,
                                uint32_t event_idx, uint64_t data)
{
    if (plat && sbi_platform_ops(plat)->pmu_xlate_to_mhpmevent)
        return sbi_platform_ops(plat)->pmu_xlate_to_mhpmevent(event_idx,
                                    data);
    return 0;
}

static inline int sbi_platform_irqchip_init(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->irqchip_init)
        return sbi_platform_ops(plat)->irqchip_init();
    return 0;
}

static inline int sbi_platform_ipi_init(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->ipi_init)
        return sbi_platform_ops(plat)->ipi_init();
    return 0;
}

static inline int sbi_platform_timer_init(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->timer_init)
        return sbi_platform_ops(plat)->timer_init();
    return 0;
}

static inline int sbi_platform_mpxy_init(const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->mpxy_init)
        return sbi_platform_ops(plat)->mpxy_init();
    return 0;
}

static inline bool sbi_platform_vendor_ext_check(
                    const struct sbi_platform *plat)
{
    if (plat && sbi_platform_ops(plat)->vendor_ext_check)
        return sbi_platform_ops(plat)->vendor_ext_check();

    return false;
}

static inline int sbi_platform_vendor_ext_provider(
                    const struct sbi_platform *plat,
                    long funcid,
                    struct sbi_trap_regs *regs,
                    struct sbi_ecall_return *out)
{
```

```
    if (plat && sbi_platform_ops(plat)->vendor_ext_provider)
        return sbi_platform_ops(plat)->vendor_ext_provider(funcid,
                                regs, out);

    return SBI_ENOTSUPP;
}

static inline int sbi_platform_emulate_load(const struct sbi_platform *plat,
                        int rlen, unsigned long addr,
                        union sbi_ldst_data *out_val)
{
    if (plat && sbi_platform_ops(plat)->emulate_load) {
        return sbi_platform_ops(plat)->emulate_load(rlen, addr,
                                out_val);
    }
    return SBI_ENOTSUPP;
}

static inline int sbi_platform_emulate_store(const struct sbi_platform *plat,
                         int wlen, unsigned long addr,
                         union sbi_ldst_data in_val)
{
    if (plat && sbi_platform_ops(plat)->emulate_store) {
        return sbi_platform_ops(plat)->emulate_store(wlen, addr,
                                 in_val);
    }
    return SBI_ENOTSUPP;
}

#endif

#endif
```

30.106 include/sbi/sbi_pmu.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_pmu.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_pmu_device

Macros

- #define SBI_PMU_HW_EVENT_MAX 256
- #define SBI_PMU_FW_CTR_MAX 16
- #define SBI_PMU_HW_CTR_MAX 32
- #define SBI_PMU_CTR_MAX (SBI_PMU_HW_CTR_MAX + SBI_PMU_FW_CTR_MAX)
- #define SBI_PMU_FIXED_CTR_MASK 0x07
- #define SBI_PMU_CY_IR_MASK 0x05

Functions

- const struct sbi_pmu_device * sbi_pmu_get_device (void)

- void sbi_pmu_set_device (const struct sbi_pmu_device *dev)
- int sbi_pmu_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_pmu_exit (struct sbi_scratch *scratch)
- int sbi_pmu_irq_bit (void)
- int sbi_pmu_add_hw_event_counter_map (u32 eidx_start, u32 eidx_end, u32 cmap)
- int sbi_pmu_add_raw_event_counter_map (uint64_t select, uint64_t select_mask, u32 cmap)
- int sbi_pmu_ctr_fw_read (uint32_t cidx, uint64_t *cval)
- int sbi_pmu_ctr_stop (unsigned long cidx_base, unsigned long cidx_mask, unsigned long flag)
- int sbi_pmu_ctr_start (unsigned long cidx_base, unsigned long cidx_mask, unsigned long flags, uint64_t ival)
- int sbi_pmu_ctr_get_info (uint32_t cidx, unsigned long *ctr_info)
- int sbi_pmu_event_get_info (unsigned long shmem_lo, unsigned long shmem_high, unsigned long num_events, unsigned long flags)
- unsigned long sbi_pmu_num_ctr (void)
- int sbi_pmu_ctr_cfg_match (unsigned long cidx_base, unsigned long cidx_mask, unsigned long flags, unsigned long event_idx, uint64_t event_data)
- int sbi_pmu_ctr_incr_fw (enum sbi_pmu_fw_event_code_id fw_id)
- void sbi_pmu_ovf_irq ()

30.106.1 Macro Definition Documentation

30.106.1.1 SBI_PMU_CTR_MAX

```
#define SBI_PMU_CTR_MAX (SBI_PMU_HW_CTR_MAX + SBI_PMU_FW_CTR_MAX)
```

30.106.1.2 SBI_PMU_CY_IR_MASK

```
#define SBI_PMU_CY_IR_MASK 0x05
```

30.106.1.3 SBI_PMU_FIXED_CTR_MASK

```
#define SBI_PMU_FIXED_CTR_MASK 0x07
```

30.106.1.4 SBI_PMU_FW_CTR_MAX

```
#define SBI_PMU_FW_CTR_MAX 16
```

30.106.1.5 SBI_PMU_HW_CTR_MAX

```
#define SBI_PMU_HW_CTR_MAX 32
```

30.106.1.6 SBI_PMU_HW_EVENT_MAX

```
#define SBI_PMU_HW_EVENT_MAX 256
```

30.106.2 Function Documentation

30.106.2.1 sbi_pmu_add_hw_event_counter_map()

```
int sbi_pmu_add_hw_event_counter_map (
            u32 eidx_start,
            u32 eidx_end,
            u32 cmap )
```

Add the hardware event to counter mapping information. This should be called from the platform code to update the mapping table.

Parameters

eidx_start	Start of the event idx range for supported counters
eidx_end	End of the event idx range for supported counters
cmap	A bitmap representing counters supporting the event range

Returns

0 on success, error otherwise.

Logical counter ids are assigned to hardware counters are assigned consecutively. E.g. counter0 must count MCYCLE where counter2 must count minstret. Similarly, counterX will mhpmcounterX. Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.2 sbi_pmu_add_raw_event_counter_map()

```
int sbi_pmu_add_raw_event_counter_map (
            uint64_t select,
            uint64_t select_mask,
            u32 cmap )
```

Add the raw hardware event selector and supported counter information. This should be called from the platform code to update the mapping table.

Parameters

info	a pointer to the hardware event info

Returns

0 on success, error otherwise.

Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.3 sbi_pmu_ctr_cfg_match()

```
int sbi_pmu_ctr_cfg_match (
            unsigned long cidx_base,
            unsigned long cidx_mask,
            unsigned long flags,
            unsigned long event_idx,
            uint64_t event_data )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.4 sbi_pmu_ctr_fw_read()

```
int sbi_pmu_ctr_fw_read (
            uint32_t cidx,
            uint64_t * cval )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.5 sbi_pmu_ctr_get_info()

```
int sbi_pmu_ctr_get_info (
            uint32_t cidx,
            unsigned long * ctr_info )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.6 sbi_pmu_ctr_incr_fw()

```
int sbi_pmu_ctr_incr_fw (
            enum sbi_pmu_fw_event_code_id fw_id )
```

Here is the caller graph for this function:

30.106.2.7 sbi_pmu_ctr_start()

```
int sbi_pmu_ctr_start (
            unsigned long cidx_base,
            unsigned long cidx_mask,
            unsigned long flags,
            uint64_t ival )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.8 sbi_pmu_ctr_stop()

```
int sbi_pmu_ctr_stop (
            unsigned long cidx_base,
            unsigned long cidx_mask,
            unsigned long flag )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.9 sbi_pmu_event_get_info()

```
int sbi_pmu_event_get_info (
            unsigned long shmem_lo,
            unsigned long shmem_high,
            unsigned long num_events,
            unsigned long flags )
```

Check shared memory size and address aligned to 16 byteHere is the call graph for this function:

Here is the caller graph for this function:

30.106.2.10 sbi_pmu_exit()

```
void sbi_pmu_exit (
            struct sbi_scratch * scratch )
```

Reset PMU during hart exit Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.11 sbi_pmu_get_device()

```
const struct sbi_pmu_device * sbi_pmu_get_device (
            void )
```

Get the PMU platform device Here is the caller graph for this function:

30.106.2.12 sbi_pmu_init()

```
int sbi_pmu_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Initialize PMU Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.13 sbi_pmu_irq_bit()

```
int sbi_pmu_irq_bit (
            void )
```

Return the pmu irq bit depending on extension existence Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.14 sbi_pmu_num_ctr()

```
unsigned long sbi_pmu_num_ctr (
            void )
```

Here is the caller graph for this function:

30.106.2.15 sbi_pmu_ovf_irq()

```
void sbi_pmu_ovf_irq ( )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.106.2.16 sbi_pmu_set_device()

```
void sbi_pmu_set_device (
            const struct sbi_pmu_device * dev )
```

Set the PMU platform device

30.107 sbi_pmu.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2021 Western Digital Corporation or its affiliates.
00005  *
00006  * Authors:
```

```
 *   Atish Patra <atish.patra@wdc.com>
 */

#ifndef __SBI_PMU_H__
#define __SBI_PMU_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_trap.h>

struct sbi_scratch;

/* Event related macros */
/* Maximum number of hardware events that can mapped by OpenSBI */
#define SBI_PMU_HW_EVENT_MAX 256

/* Counter related macros */
#define SBI_PMU_FW_CTR_MAX 16
#define SBI_PMU_HW_CTR_MAX 32
#define SBI_PMU_CTR_MAX    (SBI_PMU_HW_CTR_MAX + SBI_PMU_FW_CTR_MAX)
#define SBI_PMU_FIXED_CTR_MASK 0x07
#define SBI_PMU_CY_IR_MASK  0x05

struct sbi_pmu_device {
    char name[32];

    int (*fw_event_validate_encoding)(uint32_t hartid, uint64_t event_data);

    bool (*fw_counter_match_encoding)(uint32_t hartid,
                      uint32_t counter_index,
                      uint64_t event_data);

    int (*fw_counter_width)(void);

    uint64_t (*fw_counter_read_value)(uint32_t hartid,
                      uint32_t counter_index);

    void (*fw_counter_write_value)(uint32_t hartid, uint32_t counter_index,
                       uint64_t value);

    int (*fw_counter_start)(uint32_t hartid, uint32_t counter_index,
                uint64_t event_data);

    int (*fw_counter_stop)(uint32_t hartid, uint32_t counter_index);

    void (*hw_counter_enable_irq)(uint32_t counter_index);

    void (*hw_counter_disable_irq)(uint32_t counter_index);

    int (*hw_counter_irq_bit)(void);

    void (*hw_counter_filter_mode)(unsigned long flags, int counter_index);
};

const struct sbi_pmu_device *sbi_pmu_get_device(void);

void sbi_pmu_set_device(const struct sbi_pmu_device *dev);

int sbi_pmu_init(struct sbi_scratch *scratch, bool cold_boot);

void sbi_pmu_exit(struct sbi_scratch *scratch);

int sbi_pmu_irq_bit(void);

int sbi_pmu_add_hw_event_counter_map(u32 eidx_start, u32 eidx_end, u32 cmap);

int sbi_pmu_add_raw_event_counter_map(uint64_t select, uint64_t select_mask, u32 cmap);

int sbi_pmu_ctr_fw_read(uint32_t cidx, uint64_t *cval);

int sbi_pmu_ctr_stop(unsigned long cidx_base, unsigned long cidx_mask,
             unsigned long flag);

int sbi_pmu_ctr_start(unsigned long cidx_base, unsigned long cidx_mask,
              unsigned long flags, uint64_t ival);

int sbi_pmu_ctr_get_info(uint32_t cidx, unsigned long *ctr_info);
int sbi_pmu_event_get_info(unsigned long shmem_lo, unsigned long shmem_high,
                           unsigned long num_events, unsigned long flags);

unsigned long sbi_pmu_num_ctr(void);

int sbi_pmu_ctr_cfg_match(unsigned long cidx_base, unsigned long cidx_mask,
              unsigned long flags, unsigned long event_idx,
              uint64_t event_data);

int sbi_pmu_ctr_incr_fw(enum sbi_pmu_fw_event_code_id fw_id);

```

```
00156 void sbi_pmu_ovf_irq();
00157
00158 #endif
```

30.108 include/sbi/sbi_scratch.h File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_scratch.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_scratch

Macros

- #define SBI_SCRATCH_FW_START_OFFSET (0 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_FW_SIZE_OFFSET (1 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_FW_RW_OFFSET (2 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_FW_HEAP_OFFSET (3 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_FW_HEAP_SIZE_OFFSET (4 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_NEXT_ARG1_OFFSET (5 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_NEXT_ADDR_OFFSET (6 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_NEXT_MODE_OFFSET (7 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_WARMBOOT_ADDR_OFFSET (8 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_PLATFORM_ADDR_OFFSET (9 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET (10 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_TRAP_CONTEXT_OFFSET (11 ∗ __SIZEOF_POINTER__)

- #define SBI_SCRATCH_TMP0_OFFSET (12 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_OPTIONS_OFFSET (13 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_HARTINDEX_OFFSET (14 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_EXTRA_SPACE_OFFSET (15 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_SIZE (0x1000)
- #define sbi_scratch_thishart_ptr() ((struct sbi_scratch ∗)csr_read_relaxed(CSR_MSCRATCH))
- #define sbi_scratch_thishart_arg1_ptr() ((void ∗)(sbi_scratch_thishart_ptr()->next_arg1))
- #define sbi_scratch_offset_ptr(scratch, offset) (void ∗)((char ∗)(scratch) + (offset))
- #define sbi_scratch_thishart_offset_ptr(offset) (void ∗)((char ∗)sbi_scratch_thishart_ptr() + (offset))
- #define sbi_scratch_alloc_type_offset(__type) sbi_scratch_alloc_offset(sizeof(__type))
- #define sbi_scratch_read_type(__scratch, __type, __offset)
- #define sbi_scratch_write_type(__scratch, __type, __offset, __ptr)
- #define current_hartindex() (sbi_scratch_thishart_ptr()->hartindex)
- #define sbi_scratch_last_hartindex() last_hartindex_having_scratch
- #define sbi_hartindex_valid(__hartindex) (((__hartindex) <= sbi_scratch_last_hartindex()) ? true : false)
- #define sbi_hartindex_to_hartid(__hartindex)
- #define sbi_hartindex_to_scratch(__hartindex)
- #define sbi_hartid_to_scratch(__hartid) sbi_hartindex_to_scratch(sbi_hartid_to_hartindex(__hartid))
- #define sbi_hartid_valid(__hartid) sbi_hartindex_valid(sbi_hartid_to_hartindex(__hartid))

Enumerations

- enum sbi_scratch_options { SBI_SCRATCH_NO_BOOT_PRINTS = (1 << 0) , SBI_SCRATCH_DEBUG_PRINTS = (1 << 1) }

Functions

- int sbi_scratch_init (struct sbi_scratch ∗scratch)
- unsigned long sbi_scratch_alloc_offset (unsigned long size)
- void sbi_scratch_free_offset (unsigned long offset)
- unsigned long sbi_scratch_used_space (void)
- u32 sbi_hartid_to_hartindex (u32 hartid)

Variables

- u32 last_hartindex_having_scratch
- u32 hartindex_to_hartid_table [ ]
- struct sbi_scratch ∗ hartindex_to_scratch_table [ ]

30.108.1 Macro Definition Documentation

30.108.1.1 current_hartindex

```
#define current_hartindex( )  (sbi_scratch_thishart_ptr()->hartindex)
```

Get the hart index of the current hart

30.108.1.2 sbi_hartid_to_scratch

```
#define sbi_hartid_to_scratch(
            __hartid )  sbi_hartindex_to_scratch(sbi_hartid_to_hartindex(__hartid))
```

Get sbi_scratch from HART id

30.108.1.3 sbi_hartid_valid

```
#define sbi_hartid_valid(
            __hartid )  sbi_hartindex_valid(sbi_hartid_to_hartindex(__hartid))
```

Check whether particular HART id is valid or not

30.108.1.4 sbi_hartindex_to_hartid

```
#define sbi_hartindex_to_hartid(
            __hartindex )
```

Value:

```
({                                  \
    ((__hartindex) <= sbi_scratch_last_hartindex()) ?\
    hartindex_to_hartid_table[__hartindex] : -1U;  \
})
```

Get sbi_scratch from HART index

30.108.1.5 sbi_hartindex_to_scratch

```
#define sbi_hartindex_to_scratch(
            __hartindex )
```

Value:

```
({                                  \
    ((__hartindex) <= sbi_scratch_last_hartindex()) ?\
    hartindex_to_scratch_table[__hartindex] : NULL;\
})
```

Get sbi_scratch from HART index

30.108.1.6 sbi_hartindex_valid

```
#define sbi_hartindex_valid(
            __hartindex ) (((__hartindex) <= sbi_scratch_last_hartindex()) ? true : false)
```

Check whether a particular HART index is valid or not

30.108.1.7 sbi_scratch_alloc_type_offset

```
#define sbi_scratch_alloc_type_offset(
            __type )  sbi_scratch_alloc_offset(sizeof(__type))
```

Allocate offset for a data type in sbi_scratch

30.108.1.8 SBI_SCRATCH_EXTRA_SPACE_OFFSET

```
#define SBI_SCRATCH_EXTRA_SPACE_OFFSET (15 * __SIZEOF_POINTER__)
```

Offset of extra space in sbi_scratch

30.108.1.9 SBI_SCRATCH_FW_HEAP_OFFSET

```
#define SBI_SCRATCH_FW_HEAP_OFFSET (3 * __SIZEOF_POINTER__)
```

Offset of fw_heap_offset member in sbi_scratch

30.108.1.10 SBI_SCRATCH_FW_HEAP_SIZE_OFFSET

```
#define SBI_SCRATCH_FW_HEAP_SIZE_OFFSET (4 * __SIZEOF_POINTER__)
```

Offset of fw_heap_size member in sbi_scratch

30.108.1.11 SBI_SCRATCH_FW_RW_OFFSET

```
#define SBI_SCRATCH_FW_RW_OFFSET (2 * __SIZEOF_POINTER__)
```

Offset (in sbi_scratch) of the R/W Offset

30.108.1.12 SBI_SCRATCH_FW_SIZE_OFFSET

```
#define SBI_SCRATCH_FW_SIZE_OFFSET (1 * __SIZEOF_POINTER__)
```

Offset of fw_size member in sbi_scratch

30.108.1.13 SBI_SCRATCH_FW_START_OFFSET

```
#define SBI_SCRATCH_FW_START_OFFSET (0 * __SIZEOF_POINTER__)
```

Offset of fw_start member in sbi_scratch

30.108.1.14 SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET

```
#define SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET (10 * __SIZEOF_POINTER__)
```

Offset of hartid_to_scratch member in sbi_scratch

30.108.1.15 SBI_SCRATCH_HARTINDEX_OFFSET

```
#define SBI_SCRATCH_HARTINDEX_OFFSET (14 * __SIZEOF_POINTER__)
```

Offset of hartindex member in sbi_scratch

30.108.1.16 sbi_scratch_last_hartindex

```
#define sbi_scratch_last_hartindex( ) last_hartindex_having_scratch
```

Get last HART index having a sbi_scratch pointer

30.108.1.17 SBI_SCRATCH_NEXT_ADDR_OFFSET

```
#define SBI_SCRATCH_NEXT_ADDR_OFFSET (6 * __SIZEOF_POINTER__)
```

Offset of next_addr member in sbi_scratch

30.108.1.18 SBI_SCRATCH_NEXT_ARG1_OFFSET

```
#define SBI_SCRATCH_NEXT_ARG1_OFFSET (5 * __SIZEOF_POINTER__)
```

Offset of next_arg1 member in sbi_scratch

30.108.1.19 SBI_SCRATCH_NEXT_MODE_OFFSET

```
#define SBI_SCRATCH_NEXT_MODE_OFFSET (7 * __SIZEOF_POINTER__)
```

Offset of next_mode member in sbi_scratch

30.108.1.20 sbi_scratch_offset_ptr

```
#define sbi_scratch_offset_ptr(
              scratch,
              offset ) (void *)((char *)(scratch) + (offset))
```

Get pointer from offset in sbi_scratch

30.108.1.21 SBI_SCRATCH_OPTIONS_OFFSET

```
#define SBI_SCRATCH_OPTIONS_OFFSET (13 * __SIZEOF_POINTER__)
```

Offset of options member in sbi_scratch

30.108.1.22 SBI_SCRATCH_PLATFORM_ADDR_OFFSET

```
#define SBI_SCRATCH_PLATFORM_ADDR_OFFSET (9 * __SIZEOF_POINTER__)
```

Offset of platform_addr member in sbi_scratch

30.108.1.23 sbi_scratch_read_type

```
#define sbi_scratch_read_type(
              __scratch,
              __type,
              __offset )
```

Value:

```
({                                  \
    *((__type *)sbi_scratch_offset_ptr((__scratch), (__offset))); \
})
```

Read a data type from sbi_scratch at given offset

30.108.1.24 SBI_SCRATCH_SIZE

```
#define SBI_SCRATCH_SIZE (0x1000)
```

Maximum size of sbi_scratch (4KB)

30.108.1.25 sbi_scratch_thishart_arg1_ptr

```
#define sbi_scratch_thishart_arg1_ptr( )  ((void *)(sbi_scratch_thishart_ptr()->next_arg1))
```

Get Arg1 of next booting stage for current HART

30.108.1.26 sbi_scratch_thishart_offset_ptr

```
#define sbi_scratch_thishart_offset_ptr(
            offset )  (void *)((char *)sbi_scratch_thishart_ptr() + (offset))
```

Get pointer from offset in sbi_scratch for current HART

30.108.1.27 sbi_scratch_thishart_ptr

```
#define sbi_scratch_thishart_ptr( )  ((struct sbi_scratch *)csr_read_relaxed(CSR_MSCRATCH))
```

Get pointer to sbi_scratch for current HART

30.108.1.28 SBI_SCRATCH_TMP0_OFFSET

```
#define SBI_SCRATCH_TMP0_OFFSET (12 * __SIZEOF_POINTER__)
```

Offset of tmp0 member in sbi_scratch

30.108.1.29 SBI_SCRATCH_TRAP_CONTEXT_OFFSET

```
#define SBI_SCRATCH_TRAP_CONTEXT_OFFSET (11 * __SIZEOF_POINTER__)
```

Offset of trap_context member in sbi_scratch

30.108.1.30 SBI_SCRATCH_WARMBOOT_ADDR_OFFSET

```
#define SBI_SCRATCH_WARMBOOT_ADDR_OFFSET (8 * __SIZEOF_POINTER__)
```

Offset of warmboot_addr member in sbi_scratch

30.108.1.31 sbi_scratch_write_type

```
#define sbi_scratch_write_type(
            __scratch,
            __type,
            __offset,
            __ptr )
```

Value:

```
do {                                    \
    *((__type *)sbi_scratch_offset_ptr((__scratch), (__offset)))  \
                = (__type)(__ptr);      \
} while (0)
```

Write a data type to sbi_scratch at given offset

30.108.2 Enumeration Type Documentation

30.108.2.1 sbi_scratch_options

```
enum sbi_scratch_options
```

Prevent modification of struct sbi_scratch from affecting SBI_SCRATCH_xxx_OFFSET Possible options for Open↩ SBI library

Enumerator

SBI_SCRATCH_NO_BOOT_PRINTS	Disable prints during boot
SBI_SCRATCH_DEBUG_PRINTS	Enable runtime debug prints

30.108.3 Function Documentation

30.108.3.1 sbi_hartid_to_hartindex()

```
u32 sbi_hartid_to_hartindex (
            u32 hartid )
```

Get logical index for given HART id

Parameters

hartid	physical HART id

Returns

value between 0 to SBI_HARTMASK_MAX_BITS upon success and SBI_HARTMASK_MAX_BITS upon failure.

Here is the caller graph for this function:

30.108.3.2 sbi_scratch_alloc_offset()

```
unsigned long sbi_scratch_alloc_offset (
            unsigned long size )
```

Allocate from extra space in sbi_scratch

Returns

zero on failure and non-zero (>= SBI_SCRATCH_EXTRA_SPACE_OFFSET) on success

Here is the call graph for this function:

Here is the caller graph for this function:

30.108.3.3 sbi_scratch_free_offset()

```
void sbi_scratch_free_offset (
            unsigned long offset )
```

Free-up extra space in sbi_scratch Here is the caller graph for this function:

30.108.3.4 sbi_scratch_init()

```
int sbi_scratch_init (
            struct sbi_scratch * scratch )
```

Initialize scratch table and allocator Here is the caller graph for this function:

30.108.3.5 sbi_scratch_used_space()

```
unsigned long sbi_scratch_used_space (
            void )
```

Amount (in bytes) of used space in in sbi_scratch Here is the call graph for this function:

Here is the caller graph for this function:

30.108.4 Variable Documentation

30.108.4.1 hartindex_to_hartid_table

```
u32 hartindex_to_hartid_table[] [extern]
```

HART index to HART id table

30.108.4.2 hartindex_to_scratch_table

```
struct sbi_scratch* hartindex_to_scratch_table[] [extern]
```

HART index to scratch table

30.108.4.3 last_hartindex_having_scratch

```
u32 last_hartindex_having_scratch [extern]
```

Last HART index having a sbi_scratch pointer

30.109 sbi_scratch.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_SCRATCH_H__
#define __SBI_SCRATCH_H__

#include <sbi/riscv_asm.h>

/* clang-format off */

#define SBI_SCRATCH_FW_START_OFFSET     (0 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_FW_SIZE_OFFSET      (1 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_FW_RW_OFFSET        (2 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_FW_HEAP_OFFSET      (3 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_FW_HEAP_SIZE_OFFSET     (4 * __SIZEOF_POINTER__)
```

```
#define SBI_SCRATCH_NEXT_ARG1_OFFSET        (5 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_NEXT_ADDR_OFFSET        (6 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_NEXT_MODE_OFFSET        (7 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_WARMBOOT_ADDR_OFFSET    (8 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_PLATFORM_ADDR_OFFSET    (9 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET    (10 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_TRAP_CONTEXT_OFFSET     (11 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_TMP0_OFFSET         (12 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_OPTIONS_OFFSET      (13 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_HARTINDEX_OFFSET        (14 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_EXTRA_SPACE_OFFSET      (15 * __SIZEOF_POINTER__)
#define SBI_SCRATCH_SIZE            (0x1000)

/* clang-format on */

#ifndef __ASSEMBLER__

#include <sbi/sbi_types.h>

struct sbi_scratch {
    unsigned long fw_start;
    unsigned long fw_size;
    unsigned long fw_rw_offset;
    unsigned long fw_heap_offset;
    unsigned long fw_heap_size;
    unsigned long next_arg1;
    unsigned long next_addr;
    unsigned long next_mode;
    unsigned long warmboot_addr;
    unsigned long platform_addr;
    unsigned long hartid_to_scratch;
    unsigned long trap_context;
    unsigned long tmp0;
    unsigned long options;
    unsigned long hartindex;
};

_Static_assert(
    offsetof(struct sbi_scratch, fw_start)
        == SBI_SCRATCH_FW_START_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_FW_START_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, fw_size)
        == SBI_SCRATCH_FW_SIZE_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_FW_SIZE_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, next_arg1)
        == SBI_SCRATCH_NEXT_ARG1_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_NEXT_ARG1_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, next_addr)
        == SBI_SCRATCH_NEXT_ADDR_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_NEXT_ADDR_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, next_mode)
        == SBI_SCRATCH_NEXT_MODE_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_NEXT_MODE_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, warmboot_addr)
        == SBI_SCRATCH_WARMBOOT_ADDR_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_WARMBOOT_ADDR_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, platform_addr)
        == SBI_SCRATCH_PLATFORM_ADDR_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_PLATFORM_ADDR_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, hartid_to_scratch)
        == SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, trap_context)
        == SBI_SCRATCH_TRAP_CONTEXT_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_TRAP_CONTEXT_OFFSET");
_Static_assert(
    offsetof(struct sbi_scratch, tmp0)
        == SBI_SCRATCH_TMP0_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_TMP0_OFFSET");
```

```
_Static_assert(
    offsetof(struct sbi_scratch, options)
        == SBI_SCRATCH_OPTIONS_OFFSET,
    "struct sbi_scratch definition has changed, please redefine "
    "SBI_SCRATCH_OPTIONS_OFFSET");

enum sbi_scratch_options {
    SBI_SCRATCH_NO_BOOT_PRINTS = (1 « 0),
    SBI_SCRATCH_DEBUG_PRINTS = (1 « 1),
};

#define sbi_scratch_thishart_ptr() \
    ((struct sbi_scratch *)csr_read_relaxed(CSR_MSCRATCH))

#define sbi_scratch_thishart_arg1_ptr() \
    ((void *)(sbi_scratch_thishart_ptr()->next_arg1))

int sbi_scratch_init(struct sbi_scratch *scratch);

unsigned long sbi_scratch_alloc_offset(unsigned long size);

void sbi_scratch_free_offset(unsigned long offset);

unsigned long sbi_scratch_used_space(void);

#define sbi_scratch_offset_ptr(scratch, offset) (void *)((char *)(scratch) + (offset))

#define sbi_scratch_thishart_offset_ptr(offset) \
    (void *)((char *)sbi_scratch_thishart_ptr() + (offset))

#define sbi_scratch_alloc_type_offset(__type)                \
    sbi_scratch_alloc_offset(sizeof(__type))

#define sbi_scratch_read_type(__scratch, __type, __offset)      \
({                                                  \
    *((__type *)sbi_scratch_offset_ptr((__scratch), (__offset)));   \
})

#define sbi_scratch_write_type(__scratch, __type, __offset, __ptr)  \
do {                                                \
    *((__type *)sbi_scratch_offset_ptr((__scratch), (__offset)))    \
                    = (__type)(__ptr);      \
} while (0)

#define current_hartindex() \
    (sbi_scratch_thishart_ptr()->hartindex)

extern u32 last_hartindex_having_scratch;

#define sbi_scratch_last_hartindex()    last_hartindex_having_scratch

#define sbi_hartindex_valid(__hartindex) \
(((__hartindex) <= sbi_scratch_last_hartindex()) ? true : false)

extern u32 hartindex_to_hartid_table[];

#define sbi_hartindex_to_hartid(__hartindex)        \
({                                          \
    ((__hartindex) <= sbi_scratch_last_hartindex()) ?\
    hartindex_to_hartid_table[__hartindex] : -1U;   \
})

extern struct sbi_scratch *hartindex_to_scratch_table[];

#define sbi_hartindex_to_scratch(__hartindex)       \
({                                          \
    ((__hartindex) <= sbi_scratch_last_hartindex()) ?\
    hartindex_to_scratch_table[__hartindex] : NULL;\
})

u32 sbi_hartid_to_hartindex(u32 hartid);

#define sbi_hartid_to_scratch(__hartid) \
    sbi_hartindex_to_scratch(sbi_hartid_to_hartindex(__hartid))

#define sbi_hartid_valid(__hartid)  \
    sbi_hartindex_valid(sbi_hartid_to_hartindex(__hartid))

#endif

#endif
```

30.110 include/sbi/sbi_sse.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
#include <sbi/riscv_locks.h>
```

Include dependency graph for sbi_sse.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_sse_cb_ops

Macros

- #define EXC_MODE_PP_SHIFT 0
- #define EXC_MODE_PP BIT(EXC_MODE_PP_SHIFT)
- #define EXC_MODE_PV_SHIFT 1
- #define EXC_MODE_PV BIT(EXC_MODE_PV_SHIFT)
- #define EXC_MODE_SSTATUS_SPIE_SHIFT 2
- #define EXC_MODE_SSTATUS_SPIE BIT(EXC_MODE_SSTATUS_SPIE_SHIFT)

Functions

- int sbi_sse_set_cb_ops (uint32_t event_id, const struct sbi_sse_cb_ops *cb_ops)
- int sbi_sse_inject_event (uint32_t event_id)
- void sbi_sse_process_pending_events (struct sbi_trap_regs *regs)
- int sbi_sse_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_sse_exit (struct sbi_scratch *scratch)
- int sbi_sse_register (uint32_t event_id, unsigned long handler_entry_pc, unsigned long handler_entry_arg)

- int sbi_sse_unregister (uint32_t event_id)
- int sbi_sse_hart_mask (void)
- int sbi_sse_hart_unmask (void)
- int sbi_sse_enable (uint32_t event_id)
- int sbi_sse_disable (uint32_t event_id)
- int sbi_sse_complete (struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- int sbi_sse_inject_from_ecall (uint32_t event_id, unsigned long hart_id, struct sbi_ecall_return *out)
- int sbi_sse_read_attrs (uint32_t event_id, uint32_t base_attr_id, uint32_t attr_count, unsigned long output_phys_lo, unsigned long output_phys_hi)
- int sbi_sse_write_attrs (uint32_t event_id, uint32_t base_attr_id, uint32_t attr_count, unsigned long input_phys_lo, unsigned long input_phys_hi)

30.110.1 Macro Definition Documentation

30.110.1.1 EXC_MODE_PP

```
#define EXC_MODE_PP BIT(EXC_MODE_PP_SHIFT)
```

30.110.1.2 EXC_MODE_PP_SHIFT

```
#define EXC_MODE_PP_SHIFT 0
```

30.110.1.3 EXC_MODE_PV

```
#define EXC_MODE_PV BIT(EXC_MODE_PV_SHIFT)
```

30.110.1.4 EXC_MODE_PV_SHIFT

```
#define EXC_MODE_PV_SHIFT 1
```

30.110.1.5 EXC_MODE_SSTATUS_SPIE

```
#define EXC_MODE_SSTATUS_SPIE BIT(EXC_MODE_SSTATUS_SPIE_SHIFT)
```

30.110.1.6 EXC_MODE_SSTATUS_SPIE_SHIFT

```
#define EXC_MODE_SSTATUS_SPIE_SHIFT 2
```

30.110.2 Function Documentation

30.110.2.1 sbi_sse_complete()

```
int sbi_sse_complete (
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.2 sbi_sse_disable()

```
int sbi_sse_disable (
            uint32_t event_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.3 sbi_sse_enable()

```
int sbi_sse_enable (
            uint32_t event_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.4 sbi_sse_exit()

```
void sbi_sse_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.5 sbi_sse_hart_mask()

```
int sbi_sse_hart_mask (
            void  )
```

Here is the caller graph for this function:

30.110.2.6 sbi_sse_hart_unmask()

```
int sbi_sse_hart_unmask (
            void  )
```

Here is the caller graph for this function:

30.110.2.7 sbi_sse_init()

```
int sbi_sse_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.8 sbi_sse_inject_event()

```
int sbi_sse_inject_event (
            uint32_t event_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.9 sbi_sse_inject_from_ecall()

```
int sbi_sse_inject_from_ecall (
            uint32_t event_id,
            unsigned long hart_id,
            struct sbi_ecall_return * out )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.10 sbi_sse_process_pending_events()

```
void sbi_sse_process_pending_events (
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.11 sbi_sse_read_attrs()

```
int sbi_sse_read_attrs (
            uint32_t event_id,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long output_phys_lo,
            unsigned long output_phys_hi )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.12 sbi_sse_register()

```
int sbi_sse_register (
            uint32_t event_id,
            unsigned long handler_entry_pc,
            unsigned long handler_entry_arg )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.13 sbi_sse_set_cb_ops()

```
int sbi_sse_set_cb_ops (
            uint32_t event_id,
            const struct sbi_sse_cb_ops * cb_ops )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.14 sbi_sse_unregister()

```
int sbi_sse_unregister (
            uint32_t event_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.110.2.15 sbi_sse_write_attrs()

```
int sbi_sse_write_attrs (
            uint32_t event_id,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long input_phys_lo,
            unsigned long input_phys_hi )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.111 sbi_sse.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Rivos Systems.
 */

#ifndef __SBI_SSE_H__
#define __SBI_SSE_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
#include <sbi/riscv_locks.h>

struct sbi_scratch;
struct sbi_trap_regs;
struct sbi_ecall_return;

#define EXC_MODE_PP_SHIFT       0
#define EXC_MODE_PP         BIT(EXC_MODE_PP_SHIFT)
#define EXC_MODE_PV_SHIFT       1
#define EXC_MODE_PV         BIT(EXC_MODE_PV_SHIFT)
#define EXC_MODE_SSTATUS_SPIE_SHIFT 2
#define EXC_MODE_SSTATUS_SPIE       BIT(EXC_MODE_SSTATUS_SPIE_SHIFT)

struct sbi_sse_cb_ops {
    void (*set_hartid_cb)(uint32_t event_id, unsigned long hart_id);

    void (*complete_cb)(uint32_t event_id);

    void (*register_cb)(uint32_t event_id);

    void (*unregister_cb)(uint32_t event_id);

    void (*enable_cb)(uint32_t event_id);

    void (*disable_cb)(uint32_t event_id);
};

/* Set the callback operations for an event
 * @param event_id Event identifier (SBI_SSE_EVENT_*)
 * @param cb_ops Callback operations
 * @return 0 on success, error otherwise
 */
```

```
int sbi_sse_set_cb_ops(uint32_t event_id, const struct sbi_sse_cb_ops *cb_ops);

/* Inject an event to the current hard
 * @param event_id Event identifier (SBI_SSE_EVENT_*)
 * @param regs Registers that were used on SBI entry
 * @return 0 on success, error otherwise
 */
int sbi_sse_inject_event(uint32_t event_id);

void sbi_sse_process_pending_events(struct sbi_trap_regs *regs);


int sbi_sse_init(struct sbi_scratch *scratch, bool cold_boot);
void sbi_sse_exit(struct sbi_scratch *scratch);

/* Interface called from sbi_ecall_sse.c */
int sbi_sse_register(uint32_t event_id, unsigned long handler_entry_pc,
             unsigned long handler_entry_arg);
int sbi_sse_unregister(uint32_t event_id);
int sbi_sse_hart_mask(void);
int sbi_sse_hart_unmask(void);
int sbi_sse_enable(uint32_t event_id);
int sbi_sse_disable(uint32_t event_id);
int sbi_sse_complete(struct sbi_trap_regs *regs, struct sbi_ecall_return *out);
int sbi_sse_inject_from_ecall(uint32_t event_id, unsigned long hart_id,
                    struct sbi_ecall_return *out);
int sbi_sse_read_attrs(uint32_t event_id, uint32_t base_attr_id,
               uint32_t attr_count, unsigned long output_phys_lo,
               unsigned long output_phys_hi);
int sbi_sse_write_attrs(uint32_t event_id, uint32_t base_attr_id,
            uint32_t attr_count, unsigned long input_phys_lo,
            unsigned long input_phys_hi);

#endif
```

30.112 include/sbi/sbi_string.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_string.h:

This graph shows which files directly or indirectly include this file:

Functions

- int sbi_strcmp (const char *a, const char *b)
- int sbi_strncmp (const char *a, const char *b, size_t count)
- size_t sbi_strlen (const char *str)
- size_t sbi_strnlen (const char *str, size_t count)
- char * sbi_strcpy (char *dest, const char *src)

- char * sbi_strncpy (char *dest, const char *src, size_t count)
- char * sbi_strchr (const char *s, int c)
- char * sbi_strrchr (const char *s, int c)
- void * sbi_memset (void *s, int c, size_t count)
- void * sbi_memcpy (void *dest, const void *src, size_t count)
- void * sbi_memmove (void *dest, const void *src, size_t count)
- int sbi_memcmp (const void *s1, const void *s2, size_t count)
- void * sbi_memchr (const void *s, int c, size_t count)

30.112.1 Function Documentation

30.112.1.1 sbi_memchr()

```
void * sbi_memchr (
            const void * s,
            int c,
            size_t count )
```

30.112.1.2 sbi_memcmp()

```
int sbi_memcmp (
            const void * s1,
            const void * s2,
            size_t count )
```

Here is the caller graph for this function:

30.112.1.3 sbi_memcpy()

```
void * sbi_memcpy (
            void * dest,
            const void * src,
            size_t count )
```

Here is the caller graph for this function:

30.112.1.4 sbi_memmove()

```
void * sbi_memmove (
            void * dest,
            const void * src,
            size_t count )
```

30.112.1.5 sbi_memset()

```
void * sbi_memset (
            void * s,
            int c,
            size_t count )
```

Here is the caller graph for this function:

30.112.1.6 sbi_strchr()

```
char * sbi_strchr (
            const char * s,
            int c )
```

30.112.1.7 sbi_strcmp()

```
int sbi_strcmp (
            const char * a,
            const char * b )
```

30.112.1.8 sbi_strcpy()

```
char * sbi_strcpy (
            char * dest,
            const char * src )
```

30.112.1.9 sbi_strlen()

```
size_t sbi_strlen (
            const char * str )
```

Here is the caller graph for this function:

30.112.1.10 sbi_strncmp()

```
int sbi_strncmp (
            const char * a,
            const char * b,
            size_t count )
```

30.112.1.11 sbi_strncpy()

```
char * sbi_strncpy (
            char * dest,
            const char * src,
            size_t count )
```

Here is the caller graph for this function:

30.112.1.12 sbi_strnlen()

```
size_t sbi_strnlen (
            const char * str,
            size_t count )
```

30.112.1.13 sbi_strrchr()

```
char * sbi_strrchr (
            const char * s,
            int c )
```

Here is the call graph for this function:

30.113 sbi_string.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Atish Patra <atish.patra@wdc.com>
 */

#ifndef __STRING_H__
#define __STRING_H__

#include <sbi/sbi_types.h>

/*
  Provides sbi_strcmp for the completeness of supporting string functions.
  it is not recommended to use sbi_strcmp() but use sbi_strncmp instead.
*/

int sbi_strcmp(const char *a, const char *b);

int sbi_strncmp(const char *a, const char *b, size_t count);

size_t sbi_strlen(const char *str);

size_t sbi_strnlen(const char *str, size_t count);

```

```
char *sbi_strcpy(char *dest, const char *src);

char *sbi_strncpy(char *dest, const char *src, size_t count);

char *sbi_strchr(const char *s, int c);

char *sbi_strrchr(const char *s, int c);

void *sbi_memset(void *s, int c, size_t count);

void *sbi_memcpy(void *dest, const void *src, size_t count);

void *sbi_memmove(void *dest, const void *src, size_t count);

int sbi_memcmp(const void *s1, const void *s2, size_t count);

void *sbi_memchr(const void *s, int c, size_t count);

#endif
```

30.114 include/sbi/sbi_system.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for sbi_system.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_system_reset_device
- struct sbi_system_suspend_device

Functions

- static struct sbi_system_reset_device ∗ to_system_reset_device (struct sbi_dlist ∗node)
- const struct sbi_system_reset_device ∗ sbi_system_reset_get_device (u32 reset_type, u32 reset_reason)
- void sbi_system_reset_add_device (struct sbi_system_reset_device ∗dev)
- bool sbi_system_reset_supported (u32 reset_type, u32 reset_reason)
- void __noreturn sbi_system_reset (u32 reset_type, u32 reset_reason)

- const struct sbi_system_suspend_device ∗ sbi_system_suspend_get_device (void)
- void sbi_system_suspend_set_device (struct sbi_system_suspend_device ∗dev)
- void sbi_system_suspend_test_enable (void)
- bool sbi_system_suspend_supported (u32 sleep_type)
- int sbi_system_suspend (u32 sleep_type, ulong resume_addr, ulong opaque)

30.114.1 Function Documentation

30.114.1.1 sbi_system_reset()

```
void __noreturn sbi_system_reset (
            u32 reset_type,
            u32 reset_reason )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.114.1.2 sbi_system_reset_add_device()

```
void sbi_system_reset_add_device (
            struct sbi_system_reset_device * dev )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.114.1.3 sbi_system_reset_get_device()

```
const struct sbi_system_reset_device * sbi_system_reset_get_device (
            u32 reset_type,
            u32 reset_reason )
```

lowest priority - any non zero is our candidate
reset_type not supportedHere is the call graph for this function:

Here is the caller graph for this function:

30.114.1.4 sbi_system_reset_supported()

```
bool sbi_system_reset_supported (
            u32 reset_type,
            u32 reset_reason )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.114.1.5 sbi_system_suspend()

```
int sbi_system_suspend (
            u32 sleep_type,
            ulong resume_addr,
            ulong opaque )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.114.1.6 sbi_system_suspend_get_device()

```
const struct sbi_system_suspend_device * sbi_system_suspend_get_device (
            void  )
```

Here is the caller graph for this function:

30.114.1.7 sbi_system_suspend_set_device()

```
void sbi_system_suspend_set_device (
            struct sbi_system_suspend_device * dev )
```

Here is the caller graph for this function:

30.114.1.8 sbi_system_suspend_supported()

```
bool sbi_system_suspend_supported (
            u32 sleep_type )
```

Here is the caller graph for this function:

30.114.1.9 sbi_system_suspend_test_enable()

```
void sbi_system_suspend_test_enable (
            void )
```

Here is the call graph for this function:

30.114.1.10 to_system_reset_device()

```
static struct sbi_system_reset_device * to_system_reset_device (
            struct sbi_dlist * node )  [inline], [static]
```

Here is the caller graph for this function:

30.115 sbi_system.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_SYSTEM_H__
#define __SBI_SYSTEM_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>

struct sbi_system_reset_device {
    char name[32];

    /* Check whether reset type and reason supported by the device */
    int (*system_reset_check)(u32 reset_type, u32 reset_reason);

    void (*system_reset)(u32 reset_type, u32 reset_reason);

    struct sbi_dlist node;
};

static inline struct sbi_system_reset_device *to_system_reset_device(
                        struct sbi_dlist *node)
{
    return container_of(node, struct sbi_system_reset_device, node);
}

const struct sbi_system_reset_device *sbi_system_reset_get_device(
                    u32 reset_type, u32 reset_reason);

void sbi_system_reset_add_device(struct sbi_system_reset_device *dev);

bool sbi_system_reset_supported(u32 reset_type, u32 reset_reason);

void __noreturn sbi_system_reset(u32 reset_type, u32 reset_reason);

struct sbi_system_suspend_device {
    char name[32];

    int (*system_suspend_check)(u32 sleep_type);

    int (*system_suspend)(u32 sleep_type, unsigned long mmode_resume_addr);
};

const struct sbi_system_suspend_device *sbi_system_suspend_get_device(void);
void sbi_system_suspend_set_device(struct sbi_system_suspend_device *dev);
void sbi_system_suspend_test_enable(void);
bool sbi_system_suspend_supported(u32 sleep_type);
int sbi_system_suspend(u32 sleep_type, ulong resume_addr, ulong opaque);

#endif
```

30.116 include/sbi/sbi_timer.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for sbi_timer.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_timer_device

Functions

- void sbi_timer_delay_loop (ulong units, u64 unit_freq, void(∗delay_fn)(void ∗), void ∗opaque)
- static void sbi_timer_mdelay (ulong msecs)
- static void sbi_timer_udelay (ulong usecs)
- bool sbi_timer_waitms_until (bool(∗predicate)(void ∗), void ∗arg, uint64_t timeout_ms)
- u64 sbi_timer_value (void)
- u64 sbi_timer_virt_value (void)
- u64 sbi_timer_get_delta (void)
- void sbi_timer_set_delta (ulong delta)
- void sbi_timer_event_start (u64 next_event)
- void sbi_timer_process (void)
- const struct sbi_timer_device ∗ sbi_timer_get_device (void)
- void sbi_timer_set_device (const struct sbi_timer_device ∗dev)
- int sbi_timer_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_timer_exit (struct sbi_scratch ∗scratch)

30.116.1 Function Documentation

30.116.1.1 sbi_timer_delay_loop()

```
void sbi_timer_delay_loop (
            ulong units,
            u64 unit_freq,
            void(*)(void *) delay_fn,
            void * opaque )
```

Generic delay loop of desired granularity Here is the call graph for this function:

Here is the caller graph for this function:

30.116.1.2 sbi_timer_event_start()

```
void sbi_timer_event_start (
            u64 next_event )
```

Start timer event for current HART Update the stimecmp directly if available. This allows the older software to leverage sstc extension on newer hardware.Here is the call graph for this function:

Here is the caller graph for this function:

30.116.1.3 sbi_timer_exit()

```
void sbi_timer_exit (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.116.1.4 sbi_timer_get_delta()

```
u64 sbi_timer_get_delta (
            void )
```

Get timer delta value for current HART Here is the caller graph for this function:

30.116.1.5 sbi_timer_get_device()

```
const struct sbi_timer_device * sbi_timer_get_device (
            void )
```

Get current timer device Here is the caller graph for this function:

30.116.1.6 sbi_timer_init()

```
int sbi_timer_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.116.1.7 sbi_timer_mdelay()

```
static void sbi_timer_mdelay (
            ulong msecs )  [inline], [static]
```

Provide delay in terms of milliseconds Here is the call graph for this function:

Here is the caller graph for this function:

30.116.1.8 sbi_timer_process()

```
void sbi_timer_process (
            void  )
```

Process timer event for current HART Here is the call graph for this function:

Here is the caller graph for this function:

30.116.1.9 sbi_timer_set_delta()

```
void sbi_timer_set_delta (
            ulong delta )
```

Set timer delta value for current HART Here is the caller graph for this function:

30.116.1.10 sbi_timer_set_device()

```
void sbi_timer_set_device (
            const struct sbi_timer_device * dev )
```

Register timer device Here is the caller graph for this function:

30.116.1.11 sbi_timer_udelay()

```
static void sbi_timer_udelay (
            ulong usecs )  [inline], [static]
```

Provide delay in terms of microseconds Here is the call graph for this function:

Here is the caller graph for this function:

30.116.1.12 sbi_timer_value()

```
u64 sbi_timer_value (
            void  )
```

Get timer value for current HART Here is the caller graph for this function:

30.116.1.13 sbi_timer_virt_value()

```
u64 sbi_timer_virt_value (
            void  )
```

Get virtualized timer value for current HART Here is the call graph for this function:

Here is the caller graph for this function:

30.116.1.14 sbi_timer_waitms_until()

```
bool sbi_timer_waitms_until (
            bool(*)(void *) predicate,
            void * arg,
            uint64_t timeout_ms )
```

A blocking function that will wait until `predicate` returns true or `timeout_ms` milliseconds elapsed. `arg` will be passed as argument to `predicate` function.

Parameters

predicate	Pointer to a function that returns true if certain condition is met. It shouldn't block the code execution.
arg	Argument to pass to `predicate`.
timeout_ms	Timeout value in milliseconds. The function will return false if `timeout_ms` time period elapsed but still `predicate` doesn't return true.

Returns

true if `predicate` returns true within `timeout_ms`, false otherwise.

Here is the call graph for this function:

30.117 sbi_timer.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_TIMER_H__
#define __SBI_TIMER_H__

#include <sbi/sbi_types.h>

struct sbi_timer_device {
    char name[32];

    unsigned long timer_freq;

    u64 (*timer_value)(void);

    void (*timer_event_start)(u64 next_event);

    void (*timer_event_stop)(void);

    int (*warm_init)(void);
};

struct sbi_scratch;

void sbi_timer_delay_loop(ulong units, u64 unit_freq,
              void (*delay_fn)(void *), void *opaque);

static inline void sbi_timer_mdelay(ulong msecs)
{
    sbi_timer_delay_loop(msecs, 1000, NULL, NULL);
}

static inline void sbi_timer_udelay(ulong usecs)
{
    sbi_timer_delay_loop(usecs, 1000000, NULL, NULL);
}

bool sbi_timer_waitms_until(bool (*predicate)(void *), void *arg,
                uint64_t timeout_ms);

u64 sbi_timer_value(void);

u64 sbi_timer_virt_value(void);

u64 sbi_timer_get_delta(void);

void sbi_timer_set_delta(ulong delta);

#if __riscv_xlen == 32
void sbi_timer_set_delta_upper(ulong delta_upper);
#endif

void sbi_timer_event_start(u64 next_event);

void sbi_timer_process(void);

const struct sbi_timer_device *sbi_timer_get_device(void);

void sbi_timer_set_device(const struct sbi_timer_device *dev);

/* Initialize timer */
int sbi_timer_init(struct sbi_scratch *scratch, bool cold_boot);

/* Exit timer */
void sbi_timer_exit(struct sbi_scratch *scratch);

#endif
```

30.118 include/sbi/sbi_tlb.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_hartmask.h>
```

Include dependency graph for sbi_tlb.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_tlb_info

Macros

- #define SBI_TLB_FLUSH_ALL ((unsigned long)-1)
- #define SBI_TLB_INFO_INIT(__p, __start, __size, __asid, __vmid, __type, __src)
- #define SBI_TLB_INFO_SIZE sizeof(struct sbi_tlb_info)

Enumerations

- enum sbi_tlb_type {
  SBI_TLB_FENCE_I = 0 , SBI_TLB_SFENCE_VMA , SBI_TLB_SFENCE_VMA_ASID , SBI_TLB_HFENCE_GVMA_VMID
  ,

SBI_TLB_HFENCE_GVMA , SBI_TLB_HFENCE_VVMA_ASID , SBI_TLB_HFENCE_VVMA , SBI_TLB_TYPE_MAX
}

Functions

- int sbi_tlb_request (ulong hmask, ulong hbase, struct sbi_tlb_info *tinfo)
- int sbi_tlb_init (struct sbi_scratch *scratch, bool cold_boot)

30.118.1 Macro Definition Documentation

30.118.1.1 SBI_TLB_FLUSH_ALL

```
#define SBI_TLB_FLUSH_ALL ((unsigned long)-1)
```

30.118.1.2 SBI_TLB_INFO_INIT

```
#define SBI_TLB_INFO_INIT(
            __p,
            __start,
            __size,
            __asid,
            __vmid,
            __type,
            __src )
```

Value:

```
do { \
    (__p)->start = (__start); \
    (__p)->size = (__size); \
    (__p)->asid = (__asid); \
    (__p)->vmid = (__vmid); \
    (__p)->type = (__type); \
    SBI_HARTMASK_INIT_EXCEPT(&(__p)->smask, (__src)); \
} while (0)
```

30.118.1.3 SBI_TLB_INFO_SIZE

```
#define SBI_TLB_INFO_SIZE sizeof(struct sbi_tlb_info)
```

30.118.2 Enumeration Type Documentation

30.118.2.1 sbi_tlb_type

```
enum sbi_tlb_type
```

Enumerator

SBI_TLB_FENCE_I	
SBI_TLB_SFENCE_VMA	
SBI_TLB_SFENCE_VMA_ASID	
SBI_TLB_HFENCE_GVMA_VMID	
SBI_TLB_HFENCE_GVMA	
SBI_TLB_HFENCE_VVMA_ASID	
SBI_TLB_HFENCE_VVMA	
SBI_TLB_TYPE_MAX	

30.118.3 Function Documentation

30.118.3.1 sbi_tlb_init()

```
int sbi_tlb_init (
```

```
struct sbi_scratch * scratch,
bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.118.3.2 sbi_tlb_request()

```
int sbi_tlb_request (
            ulong hmask,
            ulong hbase,
            struct sbi_tlb_info * tinfo )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.119 sbi_tlb.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Atish Patra <atish.patra@wdc.com>
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_TLB_H__
#define __SBI_TLB_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_hartmask.h>

/* clang-format off */

#define SBI_TLB_FLUSH_ALL           ((unsigned long)-1)

/* clang-format on */

struct sbi_scratch;

enum sbi_tlb_type {
    SBI_TLB_FENCE_I = 0,
    SBI_TLB_SFENCE_VMA,
    SBI_TLB_SFENCE_VMA_ASID,
    SBI_TLB_HFENCE_GVMA_VMID,
    SBI_TLB_HFENCE_GVMA,
    SBI_TLB_HFENCE_VVMA_ASID,
    SBI_TLB_HFENCE_VVMA,
    SBI_TLB_TYPE_MAX,
};

struct sbi_tlb_info {
    unsigned long start;
    unsigned long size;
    uint16_t asid;
    uint16_t vmid;
    enum sbi_tlb_type type;
    struct sbi_hartmask smask;
};

#define SBI_TLB_INFO_INIT(__p, __start, __size, __asid, __vmid, __type, __src) \
do { \
    (__p)->start = (__start); \
    (__p)->size = (__size); \
    (__p)->asid = (__asid); \
    (__p)->vmid = (__vmid); \
    (__p)->type = (__type); \
    SBI_HARTMASK_INIT_EXCEPT(&(__p)->smask, (__src)); \
} while (0)

#define SBI_TLB_INFO_SIZE       sizeof(struct sbi_tlb_info)

int sbi_tlb_request(ulong hmask, ulong hbase, struct sbi_tlb_info *tinfo);

int sbi_tlb_init(struct sbi_scratch *scratch, bool cold_boot);
```

```
00060
00061 #endif
```

30.120 include/sbi/sbi_trap.h File Reference

```
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_scratch.h>
```

Include dependency graph for sbi_trap.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_trap_regs
- struct sbi_trap_info
- struct sbi_trap_context

Macros

- #define SBI_TRAP_REGS_zero 0
- #define SBI_TRAP_REGS_ra 1
- #define SBI_TRAP_REGS_sp 2
- #define SBI_TRAP_REGS_gp 3
- #define SBI_TRAP_REGS_tp 4

- #define SBI_TRAP_REGS_t0 5
- #define SBI_TRAP_REGS_t1 6
- #define SBI_TRAP_REGS_t2 7
- #define SBI_TRAP_REGS_s0 8
- #define SBI_TRAP_REGS_s1 9
- #define SBI_TRAP_REGS_a0 10
- #define SBI_TRAP_REGS_a1 11
- #define SBI_TRAP_REGS_a2 12
- #define SBI_TRAP_REGS_a3 13
- #define SBI_TRAP_REGS_a4 14
- #define SBI_TRAP_REGS_a5 15
- #define SBI_TRAP_REGS_a6 16
- #define SBI_TRAP_REGS_a7 17
- #define SBI_TRAP_REGS_s2 18
- #define SBI_TRAP_REGS_s3 19
- #define SBI_TRAP_REGS_s4 20
- #define SBI_TRAP_REGS_s5 21
- #define SBI_TRAP_REGS_s6 22
- #define SBI_TRAP_REGS_s7 23
- #define SBI_TRAP_REGS_s8 24
- #define SBI_TRAP_REGS_s9 25
- #define SBI_TRAP_REGS_s10 26
- #define SBI_TRAP_REGS_s11 27
- #define SBI_TRAP_REGS_t3 28
- #define SBI_TRAP_REGS_t4 29
- #define SBI_TRAP_REGS_t5 30
- #define SBI_TRAP_REGS_t6 31
- #define SBI_TRAP_REGS_mepc 32
- #define SBI_TRAP_REGS_mstatus 33
- #define SBI_TRAP_REGS_mstatusH 34
- #define SBI_TRAP_REGS_last 35
- #define SBI_TRAP_INFO_cause 0
- #define SBI_TRAP_INFO_tval 1
- #define SBI_TRAP_INFO_tval2 2
- #define SBI_TRAP_INFO_tinst 3
- #define SBI_TRAP_INFO_gva 4
- #define SBI_TRAP_INFO_last 5
- #define SBI_TRAP_REGS_OFFSET(x) ((SBI_TRAP_REGS_##x) ∗ __SIZEOF_POINTER__)
- #define SBI_TRAP_REGS_SIZE SBI_TRAP_REGS_OFFSET(last)
- #define SBI_TRAP_INFO_OFFSET(x) ((SBI_TRAP_INFO_##x) ∗ __SIZEOF_POINTER__)
- #define SBI_TRAP_INFO_SIZE SBI_TRAP_INFO_OFFSET(last)
- #define SBI_TRAP_CONTEXT_SIZE

Functions

- static unsigned long sbi_regs_gva (const struct sbi_trap_regs ∗regs)
- static bool sbi_regs_from_virt (const struct sbi_trap_regs ∗regs)
- static int sbi_mstatus_prev_mode (unsigned long mstatus)
- int sbi_trap_redirect (struct sbi_trap_regs ∗regs, const struct sbi_trap_info ∗trap)
- static struct sbi_trap_context ∗ sbi_trap_get_context (struct sbi_scratch ∗scratch)
- static void sbi_trap_set_context (struct sbi_scratch ∗scratch, struct sbi_trap_context ∗tcntx)
- struct sbi_trap_context ∗ sbi_trap_handler (struct sbi_trap_context ∗tcntx)

30.120.1 Macro Definition Documentation

30.120.1.1 SBI_TRAP_CONTEXT_SIZE

```
#define SBI_TRAP_CONTEXT_SIZE
```

Value:

```
(SBI_TRAP_REGS_SIZE + \
SBI_TRAP_INFO_SIZE + \
__SIZEOF_POINTER__)
```

Size (in bytes) of sbi_trap_context

30.120.1.2 SBI_TRAP_INFO_cause

```
#define SBI_TRAP_INFO_cause 0
```

Index of cause member in sbi_trap_info

30.120.1.3 SBI_TRAP_INFO_gva

```
#define SBI_TRAP_INFO_gva 4
```

Index of gva member in sbi_trap_info

30.120.1.4 SBI_TRAP_INFO_last

```
#define SBI_TRAP_INFO_last 5
```

Last member index in sbi_trap_info

30.120.1.5 SBI_TRAP_INFO_OFFSET

```
#define SBI_TRAP_INFO_OFFSET(
            x ) ((SBI_TRAP_INFO_##x) * __SIZEOF_POINTER__)
```

Get offset of member with name 'x' in sbi_trap_info

30.120.1.6 SBI_TRAP_INFO_SIZE

```
#define SBI_TRAP_INFO_SIZE SBI_TRAP_INFO_OFFSET(last)
```

Size (in bytes) of sbi_trap_info

30.120.1.7 SBI_TRAP_INFO_tinst

```
#define SBI_TRAP_INFO_tinst 3
```

Index of tinst member in sbi_trap_info

30.120.1.8 SBI_TRAP_INFO_tval

```
#define SBI_TRAP_INFO_tval 1
```

Index of tval member in sbi_trap_info

30.120.1.9 SBI_TRAP_INFO_tval2

```
#define SBI_TRAP_INFO_tval2 2
```

Index of tval2 member in sbi_trap_info

30.120.1.10 SBI_TRAP_REGS_a0

```
#define SBI_TRAP_REGS_a0 10
```

Index of a0 member in sbi_trap_regs

30.120.1.11 SBI_TRAP_REGS_a1

```
#define SBI_TRAP_REGS_a1 11
```

Index of a1 member in sbi_trap_regs

30.120.1.12 SBI_TRAP_REGS_a2

```
#define SBI_TRAP_REGS_a2 12
```

Index of a2 member in sbi_trap_regs

30.120.1.13 SBI_TRAP_REGS_a3

```
#define SBI_TRAP_REGS_a3 13
```

Index of a3 member in sbi_trap_regs

30.120.1.14 SBI_TRAP_REGS_a4

```
#define SBI_TRAP_REGS_a4 14
```

Index of a4 member in sbi_trap_regs

30.120.1.15 SBI_TRAP_REGS_a5

```
#define SBI_TRAP_REGS_a5 15
```

Index of a5 member in sbi_trap_regs

30.120.1.16 SBI_TRAP_REGS_a6

```
#define SBI_TRAP_REGS_a6 16
```

Index of a6 member in sbi_trap_regs

30.120.1.17 SBI_TRAP_REGS_a7

```
#define SBI_TRAP_REGS_a7 17
```

Index of a7 member in sbi_trap_regs

30.120.1.18 SBI_TRAP_REGS_gp

```
#define SBI_TRAP_REGS_gp 3
```

Index of gp member in sbi_trap_regs

30.120.1.19 SBI_TRAP_REGS_last

```
#define SBI_TRAP_REGS_last 35
```

Last member index in sbi_trap_regs

30.120.1.20 SBI_TRAP_REGS_mepc

```
#define SBI_TRAP_REGS_mepc 32
```

Index of mepc member in sbi_trap_regs

30.120.1.21 SBI_TRAP_REGS_mstatus

```
#define SBI_TRAP_REGS_mstatus 33
```

Index of mstatus member in sbi_trap_regs

30.120.1.22 SBI_TRAP_REGS_mstatusH

```
#define SBI_TRAP_REGS_mstatusH 34
```

Index of mstatusH member in sbi_trap_regs

30.120.1.23 SBI_TRAP_REGS_OFFSET

```
#define SBI_TRAP_REGS_OFFSET(
            x ) ((SBI_TRAP_REGS_##x) * __SIZEOF_POINTER__)
```

Get offset of member with name 'x' in sbi_trap_regs

30.120.1.24 SBI_TRAP_REGS_ra

```
#define SBI_TRAP_REGS_ra 1
```

Index of ra member in sbi_trap_regs

30.120.1.25 SBI_TRAP_REGS_s0

```
#define SBI_TRAP_REGS_s0 8
```

Index of s0 member in sbi_trap_regs

30.120.1.26 SBI_TRAP_REGS_s1

```
#define SBI_TRAP_REGS_s1 9
```

Index of s1 member in sbi_trap_regs

30.120.1.27 SBI_TRAP_REGS_s10

```
#define SBI_TRAP_REGS_s10 26
```

Index of s10 member in sbi_trap_regs

30.120.1.28 SBI_TRAP_REGS_s11

```
#define SBI_TRAP_REGS_s11 27
```

Index of s11 member in sbi_trap_regs

30.120.1.29 SBI_TRAP_REGS_s2

```
#define SBI_TRAP_REGS_s2 18
```

Index of s2 member in sbi_trap_regs

30.120.1.30 SBI_TRAP_REGS_s3

```
#define SBI_TRAP_REGS_s3 19
```

Index of s3 member in sbi_trap_regs

30.120.1.31 SBI_TRAP_REGS_s4

```
#define SBI_TRAP_REGS_s4 20
```

Index of s4 member in sbi_trap_regs

30.120.1.32 SBI_TRAP_REGS_s5

```
#define SBI_TRAP_REGS_s5 21
```

Index of s5 member in sbi_trap_regs

30.120.1.33 SBI_TRAP_REGS_s6

```
#define SBI_TRAP_REGS_s6 22
```

Index of s6 member in sbi_trap_regs

30.120.1.34 SBI_TRAP_REGS_s7

```
#define SBI_TRAP_REGS_s7 23
```

Index of s7 member in sbi_trap_regs

30.120.1.35 SBI_TRAP_REGS_s8

```
#define SBI_TRAP_REGS_s8 24
```

Index of s8 member in sbi_trap_regs

30.120.1.36 SBI_TRAP_REGS_s9

```
#define SBI_TRAP_REGS_s9 25
```

Index of s9 member in sbi_trap_regs

30.120.1.37 SBI_TRAP_REGS_SIZE

```
#define SBI_TRAP_REGS_SIZE SBI_TRAP_REGS_OFFSET(last)
```

Size (in bytes) of sbi_trap_regs

30.120.1.38 SBI_TRAP_REGS_sp

```
#define SBI_TRAP_REGS_sp 2
```

Index of sp member in sbi_trap_regs

30.120.1.39 SBI_TRAP_REGS_t0

```
#define SBI_TRAP_REGS_t0 5
```

Index of t0 member in sbi_trap_regs

30.120.1.40 SBI_TRAP_REGS_t1

```
#define SBI_TRAP_REGS_t1 6
```

Index of t1 member in sbi_trap_regs

30.120.1.41 SBI_TRAP_REGS_t2

```
#define SBI_TRAP_REGS_t2 7
```

Index of t2 member in sbi_trap_regs

30.120.1.42 SBI_TRAP_REGS_t3

```
#define SBI_TRAP_REGS_t3 28
```

Index of t3 member in sbi_trap_regs

30.120.1.43 SBI_TRAP_REGS_t4

```
#define SBI_TRAP_REGS_t4 29
```

Index of t4 member in sbi_trap_regs

30.120.1.44 SBI_TRAP_REGS_t5

```
#define SBI_TRAP_REGS_t5 30
```

Index of t5 member in sbi_trap_regs

30.120.1.45 SBI_TRAP_REGS_t6

```
#define SBI_TRAP_REGS_t6 31
```

Index of t6 member in sbi_trap_regs

30.120.1.46 SBI_TRAP_REGS_tp

```
#define SBI_TRAP_REGS_tp 4
```

Index of tp member in sbi_trap_regs

30.120.1.47 SBI_TRAP_REGS_zero

```
#define SBI_TRAP_REGS_zero 0
```

Index of zero member in sbi_trap_regs

30.120.2 Function Documentation

30.120.2.1 sbi_mstatus_prev_mode()

```
static int sbi_mstatus_prev_mode (
            unsigned long mstatus )  [inline], [static]
```

Here is the caller graph for this function:

30.120.2.2 sbi_regs_from_virt()

```
static bool sbi_regs_from_virt (
            const struct sbi_trap_regs * regs )  [inline], [static]
```

Here is the caller graph for this function:

30.120.2.3 sbi_regs_gva()

```
static unsigned long sbi_regs_gva (
            const struct sbi_trap_regs * regs )  [inline], [static]
```

30.120.2.4 sbi_trap_get_context()

```
static struct sbi_trap_context * sbi_trap_get_context (
            struct sbi_scratch * scratch )  [inline], [static]
```

Here is the caller graph for this function:

30.120.2.5 sbi_trap_handler()

```
struct sbi_trap_context * sbi_trap_handler (
            struct sbi_trap_context * tcntx )
```

Handle trap/interrupt

This function is called by firmware linked to OpenSBI library for handling trap/interrupt. It expects the following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. The 'mcause' CSR is having exception/interrupt cause
3. The 'mtval' CSR is having additional trap information
4. The 'mtval2' CSR is having additional trap information
5. The 'mtinst' CSR is having decoded trap instruction
6. Stack pointer (SP) is setup for current HART
7. Interrupts are disabled in MSTATUS CSR

Parameters

tcntx	pointer to trap context

Here is the call graph for this function:

30.120.2.6 sbi_trap_redirect()

```
int sbi_trap_redirect (
            struct sbi_trap_regs * regs,
            const struct sbi_trap_info * trap )
```

Redirect trap to lower privledge mode (S-mode or U-mode)

Parameters

regs	pointer to register state
trap	pointer to trap details

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.120.2.7 sbi_trap_set_context()

```
static void sbi_trap_set_context (
            struct sbi_scratch * scratch,
            struct sbi_trap_context * tcntx )  [inline], [static]
```

Here is the caller graph for this function:

30.121 sbi_trap.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_TRAP_H__
#define __SBI_TRAP_H__

#include <sbi/riscv_encoding.h>

/* clang-format off */

```

```
#define SBI_TRAP_REGS_zero          0
#define SBI_TRAP_REGS_ra            1
#define SBI_TRAP_REGS_sp            2
#define SBI_TRAP_REGS_gp            3
#define SBI_TRAP_REGS_tp            4
#define SBI_TRAP_REGS_t0            5
#define SBI_TRAP_REGS_t1            6
#define SBI_TRAP_REGS_t2            7
#define SBI_TRAP_REGS_s0            8
#define SBI_TRAP_REGS_s1            9
#define SBI_TRAP_REGS_a0            10
#define SBI_TRAP_REGS_a1            11
#define SBI_TRAP_REGS_a2            12
#define SBI_TRAP_REGS_a3            13
#define SBI_TRAP_REGS_a4            14
#define SBI_TRAP_REGS_a5            15
#define SBI_TRAP_REGS_a6            16
#define SBI_TRAP_REGS_a7            17
#define SBI_TRAP_REGS_s2            18
#define SBI_TRAP_REGS_s3            19
#define SBI_TRAP_REGS_s4            20
#define SBI_TRAP_REGS_s5            21
#define SBI_TRAP_REGS_s6            22
#define SBI_TRAP_REGS_s7            23
#define SBI_TRAP_REGS_s8            24
#define SBI_TRAP_REGS_s9            25
#define SBI_TRAP_REGS_s10           26
#define SBI_TRAP_REGS_s11           27
#define SBI_TRAP_REGS_t3            28
#define SBI_TRAP_REGS_t4            29
#define SBI_TRAP_REGS_t5            30
#define SBI_TRAP_REGS_t6            31
#define SBI_TRAP_REGS_mepc          32
#define SBI_TRAP_REGS_mstatus           33
#define SBI_TRAP_REGS_mstatusH          34
#define SBI_TRAP_REGS_last          35

#define SBI_TRAP_INFO_cause         0
#define SBI_TRAP_INFO_tval          1
#define SBI_TRAP_INFO_tval2         2
#define SBI_TRAP_INFO_tinst         3
#define SBI_TRAP_INFO_gva           4
#define SBI_TRAP_INFO_last          5

/* clang-format on */

#define SBI_TRAP_REGS_OFFSET(x) ((SBI_TRAP_REGS_##x) * __SIZEOF_POINTER__)
#define SBI_TRAP_REGS_SIZE SBI_TRAP_REGS_OFFSET(last)

#define SBI_TRAP_INFO_OFFSET(x) ((SBI_TRAP_INFO_##x) * __SIZEOF_POINTER__)
#define SBI_TRAP_INFO_SIZE SBI_TRAP_INFO_OFFSET(last)

#define SBI_TRAP_CONTEXT_SIZE (SBI_TRAP_REGS_SIZE + \
                    SBI_TRAP_INFO_SIZE + \
                    __SIZEOF_POINTER__)

#ifndef __ASSEMBLER__

#include <sbi/sbi_types.h>
#include <sbi/sbi_scratch.h>

struct sbi_trap_regs {
    unsigned long zero;
    unsigned long ra;
    unsigned long sp;
    unsigned long gp;
    unsigned long tp;
    unsigned long t0;
    unsigned long t1;
    unsigned long t2;
    unsigned long s0;
    unsigned long s1;
    unsigned long a0;
    unsigned long a1;
    unsigned long a2;
    unsigned long a3;
    unsigned long a4;
    unsigned long a5;
    unsigned long a6;
    unsigned long a7;
    unsigned long s2;
    unsigned long s3;
    unsigned long s4;
    unsigned long s5;
    unsigned long s6;
    unsigned long s7;
    unsigned long s8;
```

```
    unsigned long s9;
    unsigned long s10;
    unsigned long s11;
    unsigned long t3;
    unsigned long t4;
    unsigned long t5;
    unsigned long t6;
    unsigned long mepc;
    unsigned long mstatus;
    unsigned long mstatusH;
};

struct sbi_trap_info {
    unsigned long cause;
    unsigned long tval;
    unsigned long tval2;
    unsigned long tinst;
    unsigned long gva;
};

struct sbi_trap_context {
    struct sbi_trap_regs regs;
    struct sbi_trap_info trap;
    struct sbi_trap_context *prev_context;
};

static inline unsigned long sbi_regs_gva(const struct sbi_trap_regs *regs)
{
    /*
     * If the hypervisor extension is not implemented, mstatus[h].GVA is a
     * WPRI field, which is guaranteed to read as zero. In addition, in this
     * case we don't read mstatush and instead pretend it is zero, which
     * handles privileged spec version < 1.12.
     */

#if __riscv_xlen == 32
    return (regs->mstatusH & MSTATUSH_GVA) ? 1 : 0;
#else
    return (regs->mstatus & MSTATUS_GVA) ? 1 : 0;
#endif
}

static inline bool sbi_regs_from_virt(const struct sbi_trap_regs *regs)
{
#if __riscv_xlen == 32
    return (regs->mstatusH & MSTATUSH_MPV) ? true : false;
#else
    return (regs->mstatus & MSTATUS_MPV) ? true : false;
#endif
}

static inline int sbi_mstatus_prev_mode(unsigned long mstatus)
{
    return (mstatus & MSTATUS_MPP) >> MSTATUS_MPP_SHIFT;
}

int sbi_trap_redirect(struct sbi_trap_regs *regs,
              const struct sbi_trap_info *trap);

static inline struct sbi_trap_context *sbi_trap_get_context(struct sbi_scratch *scratch)
{
    return (scratch) ? (void *)scratch->trap_context : NULL;
}

static inline void sbi_trap_set_context(struct sbi_scratch *scratch,
                    struct sbi_trap_context *tcntx)
{
    scratch->trap_context = (unsigned long)tcntx;
}

struct sbi_trap_context *sbi_trap_handler(struct sbi_trap_context *tcntx);

#endif

#endif
```

30.122 include/sbi/sbi_trap_ldst.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_trap_ldst.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- union sbi_ldst_data

Functions

- int sbi_misaligned_load_handler (struct sbi_trap_context ∗tcntx)
- int sbi_misaligned_store_handler (struct sbi_trap_context ∗tcntx)
- int sbi_load_access_handler (struct sbi_trap_context ∗tcntx)
- int sbi_store_access_handler (struct sbi_trap_context ∗tcntx)
- int sbi_double_trap_handler (struct sbi_trap_context ∗tcntx)
- ulong sbi_misaligned_tinst_fixup (ulong orig_tinst, ulong new_tinst, ulong addr_offset)
- int sbi_misaligned_v_ld_emulator (int rlen, union sbi_ldst_data ∗out_val, struct sbi_trap_context ∗tcntx)
- int sbi_misaligned_v_st_emulator (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context ∗tcntx)

30.122.1 Function Documentation

30.122.1.1 sbi_double_trap_handler()

```
int sbi_double_trap_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.122.1.2 sbi_load_access_handler()

```
int sbi_load_access_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.122.1.3 sbi_misaligned_load_handler()

```
int sbi_misaligned_load_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.122.1.4 sbi_misaligned_store_handler()

```
int sbi_misaligned_store_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.122.1.5 sbi_misaligned_tinst_fixup()

```
ulong sbi_misaligned_tinst_fixup (
            ulong orig_tinst,
            ulong new_tinst,
            ulong addr_offset )
```

Here is the caller graph for this function:

30.122.1.6 sbi_misaligned_v_ld_emulator()

```
int sbi_misaligned_v_ld_emulator (
            int rlen,
            union sbi_ldst_data * out_val,
            struct sbi_trap_context * tcntx )
```

Here is the caller graph for this function:

30.122.1.7 sbi_misaligned_v_st_emulator()

```
int sbi_misaligned_v_st_emulator (
            int wlen,
            union sbi_ldst_data in_val,
            struct sbi_trap_context * tcntx )
```

Here is the caller graph for this function:

30.122.1.8 sbi_store_access_handler()

```
int sbi_store_access_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.123 sbi_trap_ldst.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2019 Western Digital Corporation or its affiliates.
00005  *
00006  * Authors:
00007  *   Anup Patel <anup.patel@wdc.com>
00008  */
00009
00010 #ifndef __SBI_TRAP_LDST_H__
00011 #define __SBI_TRAP_LDST_H__
00012
00013 #include <sbi/sbi_types.h>
00014 #include <sbi/sbi_trap.h>
00015
00016 union sbi_ldst_data {
00017     u64 data_u64;
00018     u32 data_u32;
00019     u8 data_bytes[8];
00020     ulong data_ulong;
00021 };
```

```
int sbi_misaligned_load_handler(struct sbi_trap_context *tcntx);

int sbi_misaligned_store_handler(struct sbi_trap_context *tcntx);

int sbi_load_access_handler(struct sbi_trap_context *tcntx);

int sbi_store_access_handler(struct sbi_trap_context *tcntx);

int sbi_double_trap_handler(struct sbi_trap_context *tcntx);

ulong sbi_misaligned_tinst_fixup(ulong orig_tinst, ulong new_tinst,
                 ulong addr_offset);

int sbi_misaligned_v_ld_emulator(int rlen, union sbi_ldst_data *out_val,
                 struct sbi_trap_context *tcntx);

int sbi_misaligned_v_st_emulator(int wlen, union sbi_ldst_data in_val,
                 struct sbi_trap_context *tcntx);

#endif
```

30.124 include/sbi/sbi_types.h File Reference

Macros

- #define true 1
- #define false 0
- #define NULL ((void *)0)
- #define __packed __attribute__((packed))
- #define __noreturn __attribute__((noreturn))
- #define __aligned(x) __attribute__((aligned(x)))
- #define __always_inline inline __attribute__((always_inline))
- #define likely(x) __builtin_expect((x), 1)
- #define unlikely(x) __builtin_expect((x), 0)
- #define __has_builtin(...) 0
- #define offsetof(TYPE, MEMBER) ((size_t) &((TYPE *)0)->MEMBER)
- #define container_of(ptr, type, member)
- #define array_size(x) (sizeof(x) / sizeof((x)[0]))
- #define MAX(a, b) ((a) > (b) ? (a) : (b))
- #define MIN(a, b) ((a) < (b) ? (a) : (b))
- #define CLAMP(a, lo, hi) MIN(MAX(a, lo), hi)
- #define STR(x) XSTR(x)
- #define XSTR(x) #x
- #define ROUNDUP(a, b) ((((a)-1) / (b) + 1) * (b))
- #define ROUNDDOWN(a, b) ((a) / (b) * (b))

Typedefs

- typedef char s8
- typedef unsigned char u8
- typedef unsigned char uint8_t
- typedef short s16
- typedef unsigned short u16
- typedef short int16_t
- typedef unsigned short uint16_t
- typedef int s32
- typedef unsigned int u32
- typedef int int32_t
- typedef unsigned int uint32_t
- typedef _Bool bool
- typedef unsigned long ulong

- typedef unsigned long uintptr_t
- typedef unsigned long size_t
- typedef long ssize_t
- typedef unsigned long virtual_addr_t
- typedef unsigned long virtual_size_t
- typedef unsigned long physical_addr_t
- typedef unsigned long physical_size_t
- typedef uint16_t le16_t
- typedef uint16_t be16_t
- typedef uint32_t le32_t
- typedef uint32_t be32_t
- typedef uint64_t le64_t
- typedef uint64_t be64_t

30.124.1 Macro Definition Documentation

30.124.1.1 __aligned

```
#define __aligned(
            x ) __attribute__((aligned(x)))
```

30.124.1.2 __always_inline

```
#define __always_inline inline __attribute__((always_inline))
```

30.124.1.3 __has_builtin

```
#define __has_builtin(
            ...  ) 0
```

30.124.1.4 __noreturn

```
#define __noreturn __attribute__((noreturn))
```

30.124.1.5 __packed

```
struct rpmi_message __packed __attribute__((packed))
```

30.124.1.6 array_size

```
#define array_size(
            x ) (sizeof(x) / sizeof((x)[0]))
```

30.124.1.7 CLAMP

```
#define CLAMP(
            a,
            lo,
            hi ) MIN(MAX(a, lo), hi)
```

30.124.1.8 container_of

```
#define container_of(
            ptr,
            type,
            member )
```

Value:

```
({          \
const typeof(((type *)0)->member) * __mptr = (ptr); \
(type *)((char *)__mptr - offsetof(type, member)); })
```

30.124.1.9 false

```
#define false 0
```

30.124.1.10 likely

```
#define likely(
            x ) __builtin_expect((x), 1)
```

30.124.1.11 MAX

```
#define MAX(
            a,
            b ) ((a) > (b) ? (a) : (b))
```

30.124.1.12 MIN

```
#define MIN(
            a,
            b ) ((a) < (b) ? (a) : (b))
```

30.124.1.13 NULL

```
#define NULL ((void *)0)
```

30.124.1.14 offsetof

```
#define offsetof(
            TYPE,
            MEMBER ) ((size_t) &((TYPE *)0)->MEMBER)
```

30.124.1.15 ROUNDDOWN

```
#define ROUNDDOWN(
            a,
            b ) ((a) / (b) * (b))
```

30.124.1.16 ROUNDUP

```
#define ROUNDUP(
            a,
            b ) ((((a)-1) / (b) + 1) * (b))
```

30.124.1.17 STR

```
#define STR(
            x ) XSTR(x)
```

30.124.1.18 true

```
#define true 1
```

30.124.1.19 unlikely

```
#define unlikely(
            x ) __builtin_expect((x), 0)
```

30.124.1.20 XSTR

```
#define XSTR(
            x ) #x
```

30.124.2 Typedef Documentation

30.124.2.1 be16_t

```
typedef uint16_t be16_t
```

30.124.2.2 be32_t

```
typedef uint32_t be32_t
```

30.124.2.3 be64_t

```
typedef uint64_t be64_t
```

30.124.2.4 bool

```
typedef _Bool bool
```

30.124.2.5 int16_t

```
typedef short int16_t
```

30.124.2.6 int32_t

```
typedef int int32_t
```

30.124.2.7 le16_t

```
typedef uint16_t le16_t
```

30.124.2.8 le32_t

```
typedef uint32_t le32_t
```

30.124.2.9 le64_t

```
typedef uint64_t le64_t
```

30.124.2.10 physical_addr_t

```
typedef unsigned long physical_addr_t
```

30.124.2.11 physical_size_t

```
typedef unsigned long physical_size_t
```

30.124.2.12 s16

```
typedef short s16
```

30.124.2.13 s32

```
typedef int s32
```

30.124.2.14 s8

```
typedef char s8
```

30.124.2.15 size_t

```
typedef unsigned long size_t
```

30.124.2.16 ssize_t

```
typedef long ssize_t
```

30.124.2.17 u16

```
typedef unsigned short u16
```

30.124.2.18 u32

```
typedef unsigned int u32
```

30.124.2.19 u8

```
typedef unsigned char u8
```

30.124.2.20 uint16_t

```
typedef unsigned short uint16_t
```

30.124.2.21 uint32_t

```
typedef unsigned int uint32_t
```

30.124.2.22 uint8_t

```
typedef unsigned char uint8_t
```

30.124.2.23 uintptr_t

```
typedef unsigned long uintptr_t
```

30.124.2.24 ulong

```
typedef unsigned long ulong
```

30.124.2.25 virtual_addr_t

```
typedef unsigned long virtual_addr_t
```

30.124.2.26 virtual_size_t

```
typedef unsigned long virtual_size_t
```

30.125 sbi_types.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_TYPES_H__
#define __SBI_TYPES_H__

#ifndef OPENSBI_EXTERNAL_SBI_TYPES

/* clang-format off */

typedef char            s8;
typedef unsigned char       u8;
typedef unsigned char       uint8_t;
```

```
typedef short           s16;
typedef unsigned short      u16;
typedef short           int16_t;
typedef unsigned short      uint16_t;

typedef int         s32;
typedef unsigned int        u32;
typedef int         int32_t;
typedef unsigned int        uint32_t;

#if __riscv_xlen == 64
typedef long            s64;
typedef unsigned long       u64;
typedef long            int64_t;
typedef unsigned long       uint64_t;
#define PRILX           "016lx"
#elif __riscv_xlen == 32
typedef long long       s64;
typedef unsigned long long  u64;
typedef long long       int64_t;
typedef unsigned long long  uint64_t;
#define PRILX           "08lx"
#else
#error "Unexpected __riscv_xlen"
#endif

#if __STDC_VERSION__ < 202000L
typedef _Bool           bool;
#define true            1
#define false           0
#endif

typedef unsigned long       ulong;
typedef unsigned long       uintptr_t;
typedef unsigned long       size_t;
typedef long            ssize_t;
typedef unsigned long       virtual_addr_t;
typedef unsigned long       virtual_size_t;
typedef unsigned long       physical_addr_t;
typedef unsigned long       physical_size_t;

typedef uint16_t        le16_t;
typedef uint16_t        be16_t;
typedef uint32_t        le32_t;
typedef uint32_t        be32_t;
typedef uint64_t        le64_t;
typedef uint64_t        be64_t;

#define NULL            ((void *)0)

#define __packed        __attribute__((packed))
#define __noreturn      __attribute__((noreturn))
#define __aligned(x)        __attribute__((aligned(x)))

#ifndef __always_inline
#define __always_inline inline __attribute__((always_inline))
#endif

#define likely(x) __builtin_expect((x), 1)
#define unlikely(x) __builtin_expect((x), 0)

#ifndef __has_builtin
#define __has_builtin(...) 0
#endif

#undef offsetof
#if __has_builtin(__builtin_offsetof)
#define offsetof(TYPE, MEMBER) __builtin_offsetof(TYPE,MEMBER)
#elif defined(__compiler_offsetof)
#define offsetof(TYPE, MEMBER) __compiler_offsetof(TYPE,MEMBER)
#else
#define offsetof(TYPE, MEMBER) ((size_t) &((TYPE *)0)->MEMBER)
#endif

#define container_of(ptr, type, member) ({          \
    const typeof(((type *)0)->member) * __mptr = (ptr); \
    (type *)((char *)__mptr - offsetof(type, member)); })

#define array_size(x)   (sizeof(x) / sizeof((x)[0]))

#define MAX(a, b) ((a) > (b) ? (a) : (b))
#define MIN(a, b) ((a) < (b) ? (a) : (b))
#define CLAMP(a, lo, hi) MIN(MAX(a, lo), hi)

#define STR(x) XSTR(x)
#define XSTR(x) #x
```

```

#define ROUNDUP(a, b) ((((a)-1) / (b) + 1) * (b))
#define ROUNDDOWN(a, b) ((a) / (b) * (b))

/* clang-format on */

#else
/*
 * OPENSBI_EXTERNAL_SBI_TYPES could be defined in CFLAGS for using the
 * external definitions of data types and common macros.
 * OPENSBI_EXTERNAL_SBI_TYPES is the file name to external header file,
 * the external build system should address the additional include
 * directory ccordingly.
 */

#define XSTR(x) #x
#define STR(x) XSTR(x)
#include STR(OPENSBI_EXTERNAL_SBI_TYPES)
#endif

#endif
```

30.126 include/sbi/sbi_unit_test.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define run_all_tests()

30.126.1 Macro Definition Documentation

30.126.1.1 run_all_tests

```
#define run_all_tests(
              void )
```

30.127 sbi_unit_test.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Author: Ivan Orlov <ivan.orlov0322@gmail.com>
 */
#ifdef CONFIG_SBIUNIT
#ifndef __SBI_UNIT_H__
#define __SBI_UNIT_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_string.h>

struct sbiunit_test_case {
    const char *name;
    bool failed;
    void (*test_func)(struct sbiunit_test_case *test);
};

struct sbiunit_test_suite {
    const char *name;
    void (*init)(void);
    struct sbiunit_test_case *cases;
};

```

```
#define SBIUNIT_TEST_CASE(func)     \
    {               \
        .name = #func,      \
        .failed = false,    \
        .test_func = (func) \
    }

#define SBIUNIT_END_CASE { }

#define SBIUNIT_TEST_SUITE(suite_name, cases_arr)       \
    struct sbiunit_test_suite suite_name = {        \
        .name = #suite_name,                \
        .init = NULL,                   \
        .cases = cases_arr              \
    }

#define _sbiunit_msg(test, msg) "[SBIUnit] [%s:%d]: %s: %s", __FILE__,  \
                __LINE__, test->name, msg

#define SBIUNIT_INFO(test, msg) sbi_printf(_sbiunit_msg(test, msg))
#define SBIUNIT_PANIC(test, msg) sbi_panic(_sbiunit_msg(test, msg))

#define SBIUNIT_EXPECT(test, cond) do {                         \
    if (!(cond)) {                              \
        test->failed = true;                            \
        SBIUNIT_INFO(test, "Condition \"" #cond "\" expected to be true!\n");   \
    }                                       \
} while (0)

#define SBIUNIT_ASSERT(test, cond) do {                     \
    if (!(cond))                                \
        SBIUNIT_PANIC(test, "Condition \"" #cond "\" must be true!\n"); \
} while (0)

#define SBIUNIT_EXPECT_EQ(test, a, b) SBIUNIT_EXPECT(test, (a) == (b))
#define SBIUNIT_ASSERT_EQ(test, a, b) SBIUNIT_ASSERT(test, (a) == (b))
#define SBIUNIT_EXPECT_NE(test, a, b) SBIUNIT_EXPECT(test, (a) != (b))
#define SBIUNIT_ASSERT_NE(test, a, b) SBIUNIT_ASSERT(test, (a) != (b))
#define SBIUNIT_EXPECT_MEMEQ(test, a, b, len) SBIUNIT_EXPECT(test, !sbi_memcmp(a, b, len))
#define SBIUNIT_ASSERT_MEMEQ(test, a, b, len) SBIUNIT_ASSERT(test, !sbi_memcmp(a, b, len))
#define SBIUNIT_EXPECT_STREQ(test, a, b, len) SBIUNIT_EXPECT(test, !sbi_strncmp(a, b, len))
#define SBIUNIT_ASSERT_STREQ(test, a, b, len) SBIUNIT_ASSERT(test, !sbi_strncmp(a, b, len))

void run_all_tests(void);
#endif
#else
#define run_all_tests()
#endif
```

30.128 include/sbi/sbi_unpriv.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sbi_unpriv.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define DECLARE_UNPRIVILEGED_LOAD_FUNCTION(type)
- #define DECLARE_UNPRIVILEGED_STORE_FUNCTION(type)

Functions

- ulong sbi_get_insn (ulong mepc, struct sbi_trap_info *trap)

30.128.1 Macro Definition Documentation

30.128.1.1 DECLARE_UNPRIVILEGED_LOAD_FUNCTION

```
#define DECLARE_UNPRIVILEGED_LOAD_FUNCTION(
            type )
```

Value:

```
type sbi_load_##type(const type *addr,            \
            struct sbi_trap_info *trap);
```

30.128.1.2 DECLARE_UNPRIVILEGED_STORE_FUNCTION

```
#define DECLARE_UNPRIVILEGED_STORE_FUNCTION(
            type )
```

Value:

```
void sbi_store_##type(type *addr, type val,       \
            struct sbi_trap_info *trap);
```

30.128.2 Function Documentation

30.128.2.1 sbi_get_insn()

```
ulong sbi_get_insn (
            ulong mepc,
            struct sbi_trap_info * trap )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.129 sbi_unpriv.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_UNPRIV_H__
#define __SBI_UNPRIV_H__

#include <sbi/sbi_types.h>

struct sbi_scratch;
struct sbi_trap_info;

#define DECLARE_UNPRIVILEGED_LOAD_FUNCTION(type)            \
    type sbi_load_##type(const type *addr,              \
                 struct sbi_trap_info *trap);

#define DECLARE_UNPRIVILEGED_STORE_FUNCTION(type)           \
    void sbi_store_##type(type *addr, type val,         \
                  struct sbi_trap_info *trap);

DECLARE_UNPRIVILEGED_LOAD_FUNCTION(u8)
DECLARE_UNPRIVILEGED_LOAD_FUNCTION(u16)
DECLARE_UNPRIVILEGED_LOAD_FUNCTION(s8)
DECLARE_UNPRIVILEGED_LOAD_FUNCTION(s16)
DECLARE_UNPRIVILEGED_LOAD_FUNCTION(s32)
DECLARE_UNPRIVILEGED_STORE_FUNCTION(u8)
DECLARE_UNPRIVILEGED_STORE_FUNCTION(u16)
DECLARE_UNPRIVILEGED_STORE_FUNCTION(u32)
DECLARE_UNPRIVILEGED_LOAD_FUNCTION(u32)
DECLARE_UNPRIVILEGED_LOAD_FUNCTION(u64)
DECLARE_UNPRIVILEGED_STORE_FUNCTION(u64)
DECLARE_UNPRIVILEGED_LOAD_FUNCTION(ulong)

ulong sbi_get_insn(ulong mepc, struct sbi_trap_info *trap);

#endif
```

30.130 include/sbi/sbi_version.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define OPENSBI_VERSION_MAJOR 1
- #define OPENSBI_VERSION_MINOR 6
- #define OPENSBI_VERSION

30.130.1 Macro Definition Documentation

30.130.1.1 OPENSBI_VERSION

```
#define OPENSBI_VERSION
```

Value:

```
((OPENSBI_VERSION_MAJOR « 16) | \
(OPENSBI_VERSION_MINOR))
```

OpenSBI 32-bit version with:

1. upper 16-bits as major number
2. lower 16-bits as minor number

30.130.1.2 OPENSBI_VERSION_MAJOR

```
#define OPENSBI_VERSION_MAJOR 1
```

30.130.1.3 OPENSBI_VERSION_MINOR

```
#define OPENSBI_VERSION_MINOR 6
```

30.131 sbi_version.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SBI_VERSION_H__
#define __SBI_VERSION_H__

#define OPENSBI_VERSION_MAJOR 1
#define OPENSBI_VERSION_MINOR 6

#define OPENSBI_VERSION ((OPENSBI_VERSION_MAJOR « 16) | \
             (OPENSBI_VERSION_MINOR))

#endif
```

30.132 include/sbi_utils/cppc/fdt_cppc.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>
```

Include dependency graph for fdt_cppc.h:

This graph shows which files directly or indirectly include this file:

Functions

- static void fdt_cppc_init (const void *fdt)

30.132.1 Function Documentation

30.132.1.1 fdt_cppc_init()

```
static void fdt_cppc_init (
            const void * fdt )  [inline], [static]
```

30.133 fdt_cppc.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
```

```
 *
 * Copyright (c) 2024 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __FDT_CPPC_H__
#define __FDT_CPPC_H__

#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>

#ifdef CONFIG_FDT_CPPC

void fdt_cppc_init(const void *fdt);

#else

static inline void fdt_cppc_init(const void *fdt) { }

#endif

#endif
```

30.134 include/sbi_utils/fdt/fdt_domain.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for fdt_domain.h:

This graph shows which files directly or indirectly include this file:

Functions

- static void fdt_domain_fixup (void ∗fdt)
- static int fdt_domains_populate (const void ∗fdt)

30.134.1 Function Documentation

30.134.1.1 fdt_domain_fixup()

```
static void fdt_domain_fixup (
            void * fdt )  [inline], [static]
```

30.134.1.2 fdt_domains_populate()

```
static int fdt_domains_populate (
            const void * fdt )  [inline], [static]
```

30.135 fdt_domain.h

Go to the documentation of this file.

```
00001 // SPDX-License-Identifier: BSD-2-Clause
00002 /*
00003  * fdt_domain.c - Flat Device Tree Domain helper routines
00004  *
00005  * Copyright (c) 2020 Western Digital Corporation or its affiliates.
00006  *
00007  * Authors:
00008  *   Anup Patel <anup.patel@wdc.com>
00009  */
00010
00011 #ifndef __FDT_DOMAIN_H__
00012 #define __FDT_DOMAIN_H__
00013
00014 #include <sbi/sbi_types.h>
00015
00016 #ifdef CONFIG_FDT_DOMAIN
00017
00018 struct sbi_domain;
00019
00029 int fdt_iterate_each_domain(void *fdt, void *opaque,
00030                 int (*fn)(void *fdt, int domain_offset,
00031                       void *opaque));
00032
00043 int fdt_iterate_each_memregion(void *fdt, int domain_offset, void *opaque,
00044                    int (*fn)(void *fdt, int domain_offset,
00045                      int region_offset, u32 region_access,
00046                      void *opaque));
00047
00061 void fdt_domain_fixup(void *fdt);
00062
00073 int fdt_domains_populate(const void *fdt);
00074
00075 #else
00076
00077 static inline void fdt_domain_fixup(void *fdt) { }
00078 static inline int fdt_domains_populate(const void *fdt) { return 0; }
00079
00080 #endif
00081
00082 #endif /* __FDT_DOMAIN_H__ */
```

30.136 include/sbi_utils/fdt/fdt_driver.h File Reference

`#include <sbi_utils/fdt/fdt_helper.h>`
Include dependency graph for fdt_driver.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_driver

Functions

- int fdt_driver_init_by_offset (const void ∗fdt, int nodeoff, const struct fdt_driver ∗const ∗drivers)
- int fdt_driver_init_all (const void ∗fdt, const struct fdt_driver ∗const ∗drivers)
- int fdt_driver_init_one (const void ∗fdt, const struct fdt_driver ∗const ∗drivers)

30.136.1 Function Documentation

30.136.1.1 fdt_driver_init_all()

```
int fdt_driver_init_all (
            const void * fdt,
            const struct fdt_driver *const * drivers )
```

Initialize a driver instance for each DT node that matches any of the provided drivers

Parameters

fdt	devicetree blob
drivers	NULL-terminated array of drivers to match against each node

Returns

0 if drivers for all matches (if any) were successfully initialized or a negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.136.1.2 fdt_driver_init_by_offset()

```
int fdt_driver_init_by_offset (
            const void * fdt,
            int nodeoff,
            const struct fdt_driver *const * drivers )
```

Initialize a driver instance for a specific DT node

Parameters

fdt	devicetree blob

Parameters

nodeoff	offset of a node in the devicetree blob
drivers	NULL-terminated array of drivers to match against this node

Returns

0 if a driver was matched and successfully initialized or a negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.136.1.3 fdt_driver_init_one()

```
int fdt_driver_init_one (
            const void * fdt,
            const struct fdt_driver *const * drivers )
```

Initialize a driver instance for the first DT node that matches any of the provided drivers

Parameters

fdt	devicetree blob
drivers	NULL-terminated array of drivers to match against each node

Returns

0 if a driver was matched and successfully initialized or a negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.137 fdt_driver.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * fdt_driver.h - Generic support for initializing drivers from DT nodes.
 *
 * Copyright (c) 2024 SiFive
 */

#ifndef __FDT_DRIVER_H__
#define __FDT_DRIVER_H__

#include <sbi_utils/fdt/fdt_helper.h>

struct fdt_driver {
    const struct fdt_match *match_table;
    int (*init)(const void *fdt, int nodeoff,
            const struct fdt_match *match);
    bool experimental;
};

int fdt_driver_init_by_offset(const void *fdt, int nodeoff,
                  const struct fdt_driver *const *drivers);

int fdt_driver_init_all(const void *fdt,
            const struct fdt_driver *const *drivers);

int fdt_driver_init_one(const void *fdt,
            const struct fdt_driver *const *drivers);

#endif /* __FDT_DRIVER_H__ */
```

30.138 include/sbi_utils/fdt/fdt_fixup.h File Reference

```
#include <sbi/sbi_list.h>
```

Include dependency graph for fdt_fixup.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_cpu_idle_state
- struct fdt_general_fixup

Functions

- int fdt_add_cpu_idle_states (void ∗fdt, const struct sbi_cpu_idle_state ∗state)
- void fdt_cpu_fixup (void ∗fdt)
- void fdt_aplic_fixup (void ∗fdt)
- void fdt_imsic_fixup (void ∗fdt)
- void fdt_plic_fixup (void ∗fdt)
- int fdt_reserved_memory_fixup (void ∗fdt)
- int fdt_register_general_fixup (struct fdt_general_fixup ∗fixup)

- void fdt_unregister_general_fixup (struct fdt_general_fixup *fixup)
- void fdt_fixups (void *fdt)

30.138.1 Function Documentation

30.138.1.1 fdt_add_cpu_idle_states()

```
int fdt_add_cpu_idle_states (
            void * fdt,
            const struct sbi_cpu_idle_state * state )
```

Add CPU idle states to cpu nodes in the DT

Add information about CPU idle states to the devicetree. This function assumes that CPU idle states are not already present in the devicetree, and that all CPU states are equally applicable to all CPUs.

Parameters

fdt	device tree blob
states	array of idle state descriptions, ending with empty element

Returns

zero on success and -ve on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.138.1.2 fdt_aplic_fixup()

```
void fdt_aplic_fixup (
            void * fdt )
```

Fix up the APLIC nodes in the device tree
This routine disables APLIC nodes which are not accessible to the next booting stage based on currently assigned domain.
It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

Here is the caller graph for this function:

30.138.1.3 fdt_cpu_fixup()

```
void fdt_cpu_fixup (
            void * fdt )
```

Fix up the CPU node in the device tree
This routine updates the "status" property of a CPU node in the device tree to "disabled" if that hart is in disabled state in OpenSBI.
It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

30.138.1.4 fdt_fixups()

```
void fdt_fixups (
            void * fdt )
```

General device tree fix-up

This routine do all required device tree fix-ups for a typical platform. It fixes up the PLIC node, IMSIC nodes, APLIC nodes, and the reserved memory node in the device tree by calling the corresponding helper routines to accomplish the task.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

30.138.1.5 fdt_imsic_fixup()

```
void fdt_imsic_fixup (
            void * fdt )
```

Fix up the IMSIC nodes in the device tree

This routine disables IMSIC nodes which are not accessible to the next booting stage based on currently assigned domain.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

Here is the caller graph for this function:

30.138.1.6 fdt_plic_fixup()

```
void fdt_plic_fixup (
            void * fdt )
```

Fix up the PLIC node in the device tree

This routine updates the "interrupt-extended" property of the PLIC node in the device tree to hide the M-mode external interrupt from CPUs.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

Here is the caller graph for this function:

30.138.1.7 fdt_register_general_fixup()

```
int fdt_register_general_fixup (
            struct fdt_general_fixup * fixup )
```

Register a general fixup Here is the call graph for this function:

Here is the caller graph for this function:

30.138.1.8 fdt_reserved_memory_fixup()

```
int fdt_reserved_memory_fixup (
            void * fdt )
```

Fix up the reserved memory node in the device tree

This routine inserts a child node of the reserved memory node in the device tree that describes the protected memory region done by OpenSBI via PMP.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Returns

zero on success and -ve on failure

We use PMP to protect OpenSBI firmware to safe-guard it from buggy S-mode software, see pmp_init() in lib/sbi/sbi_hart.c. The protected memory region information needs to be conveyed to S-mode software (e.g.↩ : operating system) via some well-known method.
With device tree, this can be done by inserting a child node of the reserved memory node which is used to specify one or more regions of reserved memory.
For the reserved memory node bindings, see Linux kernel documentation at Documentation/devicetree/bindings/reserved-memory/reserved-memory.txt
Some additional memory spaces may be protected by platform codes via PMP as well, and corresponding child nodes will be inserted. Here is the call graph for this function:

Here is the caller graph for this function:

30.138.1.9 fdt_unregister_general_fixup()

```
void fdt_unregister_general_fixup (
            struct fdt_general_fixup * fixup )
```

UnRegister a general fixup Here is the call graph for this function:

30.139 fdt_fixup.h

Go to the documentation of this file.

```
// SPDX-License-Identifier: BSD-2-Clause
/*
 * fdt_fixup.h - Flat Device Tree manipulation helper routines
 * Implement platform specific DT fixups on top of libfdt.
 *
 * Copyright (C) 2020 Bin Meng <bmeng.cn@gmail.com>
 */

#ifndef __FDT_FIXUP_H__
#define __FDT_FIXUP_H__

#include <sbi/sbi_list.h>

struct sbi_cpu_idle_state {
    const char *name;
    uint32_t suspend_param;
    bool local_timer_stop;
    uint32_t entry_latency_us;
    uint32_t exit_latency_us;
    uint32_t min_residency_us;
    uint32_t wakeup_latency_us;
};

int fdt_add_cpu_idle_states(void *fdt, const struct sbi_cpu_idle_state *state);

void fdt_cpu_fixup(void *fdt);

void fdt_aplic_fixup(void *fdt);

void fdt_imsic_fixup(void *fdt);

void fdt_plic_fixup(void *fdt);

int fdt_reserved_memory_fixup(void *fdt);

struct fdt_general_fixup {
    struct sbi_dlist head;
    const char *name;
    void (*do_fixup)(struct fdt_general_fixup *f, void *fdt);
};

int fdt_register_general_fixup(struct fdt_general_fixup *fixup);

void fdt_unregister_general_fixup(struct fdt_general_fixup *fixup);

void fdt_fixups(void *fdt);

#endif /* __FDT_FIXUP_H__ */
```

30.140 include/sbi_utils/fdt/fdt_helper.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_domain.h>
```

Include dependency graph for fdt_helper.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_match
- struct fdt_phandle_args
- struct platform_uart_data

Macros

- #define FDT_MAX_PHANDLE_ARGS 16

Functions

- const struct fdt_match ∗ fdt_match_node (const void ∗fdt, int nodeoff, const struct fdt_match ∗match_table)
- int fdt_find_match (const void ∗fdt, int startoff, const struct fdt_match ∗match_table, const struct fdt_match ∗∗out_match)
- int fdt_parse_phandle_with_args (const void ∗fdt, int nodeoff, const char ∗prop, const char ∗cells_prop, int index, struct fdt_phandle_args ∗out_args)
- int fdt_get_node_addr_size (const void ∗fdt, int node, int index, uint64_t ∗addr, uint64_t ∗size)
- int fdt_get_node_addr_size_by_name (const void ∗fdt, int node, const char ∗name, uint64_t ∗addr, uint64_t ∗size)
- bool fdt_node_is_enabled (const void ∗fdt, int nodeoff)
- int fdt_parse_hart_id (const void ∗fdt, int cpu_offset, u32 ∗hartid)
- int fdt_parse_max_enabled_hart_id (const void ∗fdt, u32 ∗max_hartid)
- int fdt_parse_timebase_frequency (const void ∗fdt, unsigned long ∗freq)
- int fdt_parse_isa_extensions (const void ∗fdt, unsigned int hard_id, unsigned long ∗extensions)
- int fdt_parse_gaisler_uart_node (const void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_renesas_scif_node (const void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_shakti_uart_node (const void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_sifive_uart_node (const void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)

- int fdt_parse_uart_node (const void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_uart8250 (const void ∗fdt, struct platform_uart_data ∗uart, const char ∗compatible)
- int fdt_parse_xlnx_uartlite_node (const void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_aplic_node (const void ∗fdt, int nodeoff, struct aplic_data ∗aplic)
- bool fdt_check_imsic_mlevel (const void ∗fdt)
- int fdt_parse_imsic_node (const void ∗fdt, int nodeoff, struct imsic_data ∗imsic)
- int fdt_parse_plic_node (const void ∗fdt, int nodeoffset, struct plic_data ∗plic)
- int fdt_parse_plic (const void ∗fdt, struct plic_data ∗plic, const char ∗compat)
- int fdt_parse_aclint_node (const void ∗fdt, int nodeoffset, bool for_timer, bool allow_regname, unsigned long ∗out_addr1, unsigned long ∗out_size1, unsigned long ∗out_addr2, unsigned long ∗out_size2, u32 ∗out_↩ first_hartid, u32 ∗out_hart_count)
- int fdt_parse_plmt_node (const void ∗fdt, int nodeoffset, unsigned long ∗plmt_base, unsigned long ∗plmt_↩ size, u32 ∗hart_count)
- int fdt_parse_plicsw_node (const void ∗fdt, int nodeoffset, unsigned long ∗plicsw_base, unsigned long ∗size, u32 ∗hart_count)
- int fdt_parse_compat_addr (const void ∗fdt, uint64_t ∗addr, const char ∗compatible)
- static const void ∗ fdt_get_address (void)
- static void ∗ fdt_get_address_rw (void)

30.140.1 Macro Definition Documentation

30.140.1.1 FDT_MAX_PHANDLE_ARGS

```
#define FDT_MAX_PHANDLE_ARGS 16
```

30.140.2 Function Documentation

30.140.2.1 fdt_check_imsic_mlevel()

```
bool fdt_check_imsic_mlevel (
            const void * fdt )
```

Here is the call graph for this function:

30.140.2.2 fdt_find_match()

```
int fdt_find_match (
            const void * fdt,
            int startoff,
            const struct fdt_match * match_table,
            const struct fdt_match ** out_match )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.3 fdt_get_address()

```
static const void * fdt_get_address (
            void  )  [inline], [static]
```

Here is the caller graph for this function:

30.140.2.4 fdt_get_address_rw()

```
static void * fdt_get_address_rw (
            void  )  [inline], [static]
```

30.140.2.5 fdt_get_node_addr_size()

```
int fdt_get_node_addr_size (
            const void * fdt,
            int node,
            int index,
            uint64_t * addr,
            uint64_t * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.6 fdt_get_node_addr_size_by_name()

```
int fdt_get_node_addr_size_by_name (
            const void * fdt,
            int node,
            const char * name,
            uint64_t * addr,
            uint64_t * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.7 fdt_match_node()

```
const struct fdt_match * fdt_match_node (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match_table )
```

Here is the call graph for this function:

30.140.2.8 fdt_node_is_enabled()

```
bool fdt_node_is_enabled (
            const void * fdt,
            int nodeoff )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.9 fdt_parse_aclint_node()

```
int fdt_parse_aclint_node (
            const void * fdt,
            int nodeoffset,
            bool for_timer,
            bool allow_regname,
            unsigned long * out_addr1,
            unsigned long * out_size1,
            unsigned long * out_addr2,
            unsigned long * out_size2,
            u32 * out_first_hartid,
            u32 * out_hart_count )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.10 fdt_parse_aplic_node()

```
int fdt_parse_aplic_node (
            const void * fdt,
            int nodeoff,
            struct aplic_data * aplic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.11 fdt_parse_compat_addr()

```
int fdt_parse_compat_addr (
            const void * fdt,
            uint64_t * addr,
            const char * compatible )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.12 fdt_parse_gaisler_uart_node()

```
int fdt_parse_gaisler_uart_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.13 fdt_parse_hart_id()

```
int fdt_parse_hart_id (
            const void * fdt,
            int cpu_offset,
            u32 * hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.14 fdt_parse_imsic_node()

```
int fdt_parse_imsic_node (
            const void * fdt,
            int nodeoff,
            struct imsic_data * imsic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.15 fdt_parse_isa_extensions()

```
int fdt_parse_isa_extensions (
            const void * fdt,
            unsigned int hard_id,
            unsigned long * extensions )
```

Here is the call graph for this function:

30.140.2.16 fdt_parse_max_enabled_hart_id()

```
int fdt_parse_max_enabled_hart_id (
            const void * fdt,
            u32 * max_hartid )
```

Here is the call graph for this function:

30.140.2.17 fdt_parse_phandle_with_args()

```
int fdt_parse_phandle_with_args (
            const void * fdt,
            int nodeoff,
            const char * prop,
            const char * cells_prop,
            int index,
            struct fdt_phandle_args * out_args )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.18 fdt_parse_plic()

```
int fdt_parse_plic (
            const void * fdt,
            struct plic_data * plic,
            const char * compat )
```

Here is the call graph for this function:

30.140.2.19 fdt_parse_plic_node()

```
int fdt_parse_plic_node (
            const void * fdt,
            int nodeoffset,
            struct plic_data * plic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.20 fdt_parse_plicsw_node()

```
int fdt_parse_plicsw_node (
            const void * fdt,
            int nodeoffset,
            unsigned long * plicsw_base,
            unsigned long * size,
            u32 * hart_count )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.21 fdt_parse_plmt_node()

```
int fdt_parse_plmt_node (
            const void * fdt,
            int nodeoffset,
            unsigned long * plmt_base,
            unsigned long * plmt_size,
            u32 * hart_count )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.22 fdt_parse_renesas_scif_node()

```
int fdt_parse_renesas_scif_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.23 fdt_parse_shakti_uart_node()

```
int fdt_parse_shakti_uart_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.24 fdt_parse_sifive_uart_node()

```
int fdt_parse_sifive_uart_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.25 fdt_parse_timebase_frequency()

```
int fdt_parse_timebase_frequency (
            const void * fdt,
            unsigned long * freq )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.26 fdt_parse_uart8250()

```
int fdt_parse_uart8250 (
            const void * fdt,
            struct platform_uart_data * uart,
            const char * compatible )
```

Here is the call graph for this function:

30.140.2.27 fdt_parse_uart_node()

```
int fdt_parse_uart_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.140.2.28 fdt_parse_xlnx_uartlite_node()

```
int fdt_parse_xlnx_uartlite_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.141 fdt_helper.h

Go to the documentation of this file.

```
// SPDX-License-Identifier: BSD-2-Clause
/*
 * fdt_helper.h - Flat Device Tree parsing helper routines
 * Implement helper routines to parse FDT nodes on top of
 * libfdt for OpenSBI usage
 *
 * Copyright (C) 2020 Bin Meng <bmeng.cn@gmail.com>
 */

#ifndef __FDT_HELPER_H__
#define __FDT_HELPER_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_domain.h>

struct fdt_match {
    const char *compatible;
    const void *data;
};

#define FDT_MAX_PHANDLE_ARGS 16
struct fdt_phandle_args {
    int node_offset;
    int args_count;
    u32 args[FDT_MAX_PHANDLE_ARGS];
};

struct platform_uart_data {
    unsigned long addr;
    unsigned long freq;
    unsigned long baud;
    unsigned long reg_shift;
    unsigned long reg_io_width;
    unsigned long reg_offset;
};

const struct fdt_match *fdt_match_node(const void *fdt, int nodeoff,
                       const struct fdt_match *match_table);

int fdt_find_match(const void *fdt, int startoff,
           const struct fdt_match *match_table,
           const struct fdt_match **out_match);

int fdt_parse_phandle_with_args(const void *fdt, int nodeoff,
                const char *prop, const char *cells_prop,
                int index, struct fdt_phandle_args *out_args);

int fdt_get_node_addr_size(const void *fdt, int node, int index,
               uint64_t *addr, uint64_t *size);

int fdt_get_node_addr_size_by_name(const void *fdt, int node, const char *name,
                   uint64_t *addr, uint64_t *size);

bool fdt_node_is_enabled(const void *fdt, int nodeoff);

int fdt_parse_hart_id(const void *fdt, int cpu_offset, u32 *hartid);

int fdt_parse_max_enabled_hart_id(const void *fdt, u32 *max_hartid);

int fdt_parse_timebase_frequency(const void *fdt, unsigned long *freq);

int fdt_parse_isa_extensions(const void *fdt, unsigned int hard_id,
                 unsigned long *extensions);

int fdt_parse_gaisler_uart_node(const void *fdt, int nodeoffset,
```

```
                struct platform_uart_data *uart);

int fdt_parse_renesas_scif_node(const void *fdt, int nodeoffset,
                struct platform_uart_data *uart);

int fdt_parse_shakti_uart_node(const void *fdt, int nodeoffset,
                   struct platform_uart_data *uart);

int fdt_parse_sifive_uart_node(const void *fdt, int nodeoffset,
                   struct platform_uart_data *uart);

int fdt_parse_uart_node(const void *fdt, int nodeoffset,
            struct platform_uart_data *uart);

int fdt_parse_uart8250(const void *fdt, struct platform_uart_data *uart,
               const char *compatible);

int fdt_parse_xlnx_uartlite_node(const void *fdt, int nodeoffset,
                   struct platform_uart_data *uart);

struct aplic_data;

int fdt_parse_aplic_node(const void *fdt, int nodeoff, struct aplic_data *aplic);

struct imsic_data;

bool fdt_check_imsic_mlevel(const void *fdt);

int fdt_parse_imsic_node(const void *fdt, int nodeoff, struct imsic_data *imsic);

struct plic_data;

int fdt_parse_plic_node(const void *fdt, int nodeoffset, struct plic_data *plic);

int fdt_parse_plic(const void *fdt, struct plic_data *plic, const char *compat);

int fdt_parse_aclint_node(const void *fdt, int nodeoffset,
              bool for_timer, bool allow_regname,
              unsigned long *out_addr1, unsigned long *out_size1,
              unsigned long *out_addr2, unsigned long *out_size2,
              u32 *out_first_hartid, u32 *out_hart_count);

int fdt_parse_plmt_node(const void *fdt, int nodeoffset, unsigned long *plmt_base,
            unsigned long *plmt_size, u32 *hart_count);

int fdt_parse_plicsw_node(const void *fdt, int nodeoffset, unsigned long *plicsw_base,
              unsigned long *size, u32 *hart_count);

int fdt_parse_compat_addr(const void *fdt, uint64_t *addr,
              const char *compatible);

static inline const void *fdt_get_address(void)
{
    return (const void *)root.next_arg1;
}

static inline void *fdt_get_address_rw(void)
{
    return (void *)root.next_arg1;
}

#endif /* __FDT_HELPER_H__ */
```

30.142 include/sbi_utils/fdt/fdt_pmu.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_pmu.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_pmu_hw_event_select_map
- struct fdt_pmu_hw_event_counter_map
- struct fdt_pmu_raw_event_counter_map

Functions

- static void fdt_pmu_fixup (void ∗fdt)
- static int fdt_pmu_setup (const void ∗fdt)
- static uint64_t fdt_pmu_get_select_value (uint32_t event_idx)

30.142.1 Function Documentation

30.142.1.1 fdt_pmu_fixup()

```
static void fdt_pmu_fixup (
            void * fdt )  [inline], [static]
```

Here is the caller graph for this function:

30.142.1.2 fdt_pmu_get_select_value()

```
static uint64_t fdt_pmu_get_select_value (
            uint32_t event_idx )  [inline], [static]
```

30.142.1.3 fdt_pmu_setup()

```
static int fdt_pmu_setup (
            const void * fdt )  [inline], [static]
```

30.143 fdt_pmu.h

Go to the documentation of this file.

```
00001 // SPDX-License-Identifier: BSD-2-Clause
00002 /*
00003  * fdt_pmu.c - Flat Device Tree PMU helper routines
00004  *
00005  * Copyright (c) 2021 Western Digital Corporation or its affiliates.
00006  *
00007  * Authors:
00008  *   Atish Patra <atish.patra@wdc.com>
00009  */
00010
00011 #ifndef __FDT_PMU_H__
00012 #define __FDT_PMU_H__
00013
00014 #include <sbi/sbi_types.h>
00015
00016 struct fdt_pmu_hw_event_select_map {
00017     uint32_t eidx;
00018     uint64_t select;
00019 };
00020
00021 struct fdt_pmu_hw_event_counter_map {
00022     uint32_t eidx_start;
00023     uint32_t eidx_end;
00024     uint32_t ctr_map;
00025 };
00026
00027 struct fdt_pmu_raw_event_counter_map {
00028     uint64_t select;
00029     uint64_t select_mask;
00030     uint32_t ctr_map;
00031 };
00032
00033 #ifdef CONFIG_FDT_PMU
00034
00046 int fdt_pmu_fixup(void *fdt);
00047
00055 int fdt_pmu_setup(const void *fdt);
00056
00063 uint64_t fdt_pmu_get_select_value(uint32_t event_idx);
00064
00066 extern struct fdt_pmu_hw_event_select_map fdt_pmu_evt_select[];
00068 extern uint32_t hw_event_count;
00069
00070 #else
00071
00072 static inline void fdt_pmu_fixup(void *fdt) { }
00073 static inline int fdt_pmu_setup(const void *fdt) { return 0; }
00074 static inline uint64_t fdt_pmu_get_select_value(uint32_t event_idx) { return 0; }
```

```
00075
00076 #endif
00077
00078 #endif
```

30.144 include/sbi_utils/gpio/fdt_gpio.h File Reference

```
#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/gpio/gpio.h>
```

Include dependency graph for fdt_gpio.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_gpio

Functions

- int fdt_gpio_pin_get (const void ∗fdt, int nodeoff, int index, struct gpio_pin ∗out_pin)
- int fdt_gpio_simple_xlate (struct gpio_chip ∗chip, const struct fdt_phandle_args ∗pargs, struct gpio_pin ∗out_pin)

30.144.1 Function Documentation

30.144.1.1 fdt_gpio_pin_get()

```
int fdt_gpio_pin_get (
```

```
                const void * fdt,
                int nodeoff,
                int index,
                struct gpio_pin * out_pin )
```

Get a GPIO pin using "gpios" DT property of client DT node Here is the call graph for this function:

Here is the caller graph for this function:

30.144.1.2 fdt_gpio_simple_xlate()

```
int fdt_gpio_simple_xlate (
                struct gpio_chip * chip,
                const struct fdt_phandle_args * pargs,
                struct gpio_pin * out_pin )
```

Simple xlate function to convert two GPIO FDT cells into GPIO pin

30.145 fdt_gpio.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __FDT_GPIO_H__
#define __FDT_GPIO_H__

#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/gpio/gpio.h>

struct fdt_phandle_args;

struct fdt_gpio {
    struct fdt_driver driver;
    int (*xlate)(struct gpio_chip *chip,
                 const struct fdt_phandle_args *pargs,
                 struct gpio_pin *out_pin);
};

int fdt_gpio_pin_get(const void *fdt, int nodeoff, int index,
```

```
00028                struct gpio_pin *out_pin);
00029
00031 int fdt_gpio_simple_xlate(struct gpio_chip *chip,
00032                 const struct fdt_phandle_args *pargs,
00033                 struct gpio_pin *out_pin);
00034
00035 #endif
```

30.146 include/sbi_utils/gpio/gpio.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for gpio.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct gpio_pin
- struct gpio_chip

Macros

- #define GPIO_LINE_DIRECTION_IN 1
- #define GPIO_LINE_DIRECTION_OUT 0
- #define GPIO_FLAG_ACTIVE_LOW 0x1

- #define GPIO_FLAG_SINGLE_ENDED 0x2
- #define GPIO_FLAG_OPEN_DRAIN 0x4
- #define GPIO_FLAG_TRANSITORY 0x8
- #define GPIO_FLAG_PULL_UP 0x10
- #define GPIO_FLAG_PULL_DOWN 0x20

Functions

- static struct gpio_chip * to_gpio_chip (struct sbi_dlist *node)
- struct gpio_chip * gpio_chip_find (unsigned int id)
- int gpio_chip_add (struct gpio_chip *gc)
- void gpio_chip_remove (struct gpio_chip *gc)
- int gpio_get_direction (struct gpio_pin *gp)
- int gpio_direction_input (struct gpio_pin *gp)
- int gpio_direction_output (struct gpio_pin *gp, int value)
- int gpio_get (struct gpio_pin *gp)
- int gpio_set (struct gpio_pin *gp, int value)

30.146.1 Macro Definition Documentation

30.146.1.1 GPIO_FLAG_ACTIVE_LOW

```
#define GPIO_FLAG_ACTIVE_LOW 0x1
```

30.146.1.2 GPIO_FLAG_OPEN_DRAIN

```
#define GPIO_FLAG_OPEN_DRAIN 0x4
```

30.146.1.3 GPIO_FLAG_PULL_DOWN

```
#define GPIO_FLAG_PULL_DOWN 0x20
```

30.146.1.4 GPIO_FLAG_PULL_UP

```
#define GPIO_FLAG_PULL_UP 0x10
```

30.146.1.5 GPIO_FLAG_SINGLE_ENDED

```
#define GPIO_FLAG_SINGLE_ENDED 0x2
```

30.146.1.6 GPIO_FLAG_TRANSITORY

```
#define GPIO_FLAG_TRANSITORY 0x8
```

30.146.1.7 GPIO_LINE_DIRECTION_IN

```
#define GPIO_LINE_DIRECTION_IN 1
```

30.146.1.8 GPIO_LINE_DIRECTION_OUT

```
#define GPIO_LINE_DIRECTION_OUT 0
```

30.146.2 Function Documentation

30.146.2.1 gpio_chip_add()

```
int gpio_chip_add (
            struct gpio_chip * gc )
```

Register GPIO chip Here is the call graph for this function:

Here is the caller graph for this function:

30.146.2.2 gpio_chip_find()

```
struct gpio_chip * gpio_chip_find (
            unsigned int id )
```

Find a registered GPIO chip Here is the call graph for this function:

Here is the caller graph for this function:

30.146.2.3 gpio_chip_remove()

```
void gpio_chip_remove (
            struct gpio_chip * gc )
```

Un-register GPIO chip Here is the call graph for this function:

30.146.2.4 gpio_direction_input()

```
int gpio_direction_input (
            struct gpio_pin * gp )
```

Set input direction of GPIO pin Here is the caller graph for this function:

30.146.2.5 gpio_direction_output()

```
int gpio_direction_output (
            struct gpio_pin * gp,
            int value )
```

Set output direction of GPIO pin Here is the caller graph for this function:

30.146.2.6 gpio_get()

```
int gpio_get (
            struct gpio_pin * gp )
```

Get current value of GPIO pin

30.146.2.7 gpio_get_direction()

```
int gpio_get_direction (
            struct gpio_pin * gp )
```

Get current direction of GPIO pin

30.146.2.8 gpio_set()

```
int gpio_set (
            struct gpio_pin * gp,
            int value )
```

Set output value of GPIO pin Here is the caller graph for this function:

30.146.2.9 to_gpio_chip()

```
static struct gpio_chip * to_gpio_chip (
            struct sbi_dlist * node )  [inline], [static]
```

Here is the caller graph for this function:

30.147 gpio.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __GPIO_H__
#define __GPIO_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>

#define GPIO_LINE_DIRECTION_IN  1
#define GPIO_LINE_DIRECTION_OUT 0

struct gpio_pin {
    struct gpio_chip *chip;
    unsigned int offset;
    unsigned int flags;
#define GPIO_FLAG_ACTIVE_LOW    0x1
#define GPIO_FLAG_SINGLE_ENDED  0x2
#define GPIO_FLAG_OPEN_DRAIN    0x4
#define GPIO_FLAG_TRANSITORY    0x8
#define GPIO_FLAG_PULL_UP   0x10
#define GPIO_FLAG_PULL_DOWN 0x20
};

struct gpio_chip {
    const void *driver;
    unsigned int id;
    unsigned int ngpio;
    int (*get_direction)(struct gpio_pin *gp);
    int (*direction_input)(struct gpio_pin *gp);
    int (*direction_output)(struct gpio_pin *gp, int value);
    int (*get)(struct gpio_pin *gp);
    void (*set)(struct gpio_pin *gp, int value);
    struct sbi_dlist node;
};

static inline struct gpio_chip *to_gpio_chip(struct sbi_dlist *node)
{
    return container_of(node, struct gpio_chip, node);
}

struct gpio_chip *gpio_chip_find(unsigned int id);

int gpio_chip_add(struct gpio_chip *gc);

void gpio_chip_remove(struct gpio_chip *gc);

int gpio_get_direction(struct gpio_pin *gp);

int gpio_direction_input(struct gpio_pin *gp);

int gpio_direction_output(struct gpio_pin *gp, int value);

int gpio_get(struct gpio_pin *gp);

int gpio_set(struct gpio_pin *gp, int value);

#endif
```

30.148 include/sbi_utils/hsm/fdt_hsm.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>
```

Include dependency graph for fdt_hsm.h:

This graph shows which files directly or indirectly include this file:

Functions

- static void fdt_hsm_init (const void *fdt)

30.148.1 Function Documentation

30.148.1.1 fdt_hsm_init()

```
static void fdt_hsm_init (
            const void * fdt )  [inline], [static]
```

30.149 fdt_hsm.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
```

```
 *
 * Copyright (c) 2024 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __FDT_HSM_H__
#define __FDT_HSM_H__

#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>

#ifdef CONFIG_FDT_HSM

void fdt_hsm_init(const void *fdt);

#else

static inline void fdt_hsm_init(const void *fdt) { }

#endif

#endif
```

30.150 include/sbi_utils/i2c/dw_i2c.h File Reference

`#include <sbi_utils/i2c/i2c.h>`

Include dependency graph for dw_i2c.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct dw_i2c_adapter

Functions

- int dw_i2c_init (struct i2c_adapter *, int nodeoff)

30.150.1 Function Documentation

30.150.1.1 dw_i2c_init()

```
int dw_i2c_init (
            struct i2c_adapter * adapter,
            int nodeoff )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.151 dw_i2c.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2022 StarFive Technology Co., Ltd.
 *
 * Author: Minda Chen <minda.chen@starfivetech.com>
 */

#ifndef __DW_I2C_H__
#define __DW_I2C_H__

#include <sbi_utils/i2c/i2c.h>

int dw_i2c_init(struct i2c_adapter *, int nodeoff);

struct dw_i2c_adapter {
    unsigned long addr;
    struct i2c_adapter adapter;
};

#endif
```

30.152 include/sbi_utils/i2c/fdt_i2c.h File Reference

```
#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/i2c/i2c.h>
```

Include dependency graph for fdt_i2c.h:

This graph shows which files directly or indirectly include this file:

Functions

- int fdt_i2c_adapter_get (const void *fdt, int nodeoff, struct i2c_adapter **out_adapter)

30.152.1 Function Documentation

30.152.1.1 fdt_i2c_adapter_get()

```
int fdt_i2c_adapter_get (
            const void * fdt,
            int nodeoff,
            struct i2c_adapter ** out_adapter )
```

Get I2C adapter identified by nodeoff Here is the call graph for this function:

Here is the caller graph for this function:

30.153 fdt_i2c.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 YADRO
 *
 * Authors:
 *   Nikita Shubin <n.shubin@yadro.com>
 */

#ifndef __FDT_I2C_H__
#define __FDT_I2C_H__
```

```
#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/i2c/i2c.h>

int fdt_i2c_adapter_get(const void *fdt, int nodeoff,
            struct i2c_adapter **out_adapter);

#endif
```

30.154 include/sbi_utils/i2c/i2c.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for i2c.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct i2c_adapter

Functions

- static struct i2c_adapter ∗ to_i2c_adapter (struct sbi_dlist ∗node)
- struct i2c_adapter ∗ i2c_adapter_find (int id)
- int i2c_adapter_add (struct i2c_adapter ∗ia)
- void i2c_adapter_remove (struct i2c_adapter ∗ia)
- int i2c_adapter_write (struct i2c_adapter ∗ia, uint8_t addr, uint8_t reg, uint8_t ∗buffer, int len)

- int i2c_adapter_read (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- static int i2c_adapter_reg_write (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t val)
- static int i2c_adapter_reg_read (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *val)

30.154.1 Function Documentation

30.154.1.1 i2c_adapter_add()

```
int i2c_adapter_add (
            struct i2c_adapter * ia )
```

Register I2C adapter Here is the call graph for this function:

Here is the caller graph for this function:

30.154.1.2 i2c_adapter_find()

```
struct i2c_adapter * i2c_adapter_find (
            int id )
```

Find a registered I2C adapter Here is the call graph for this function:

Here is the caller graph for this function:

30.154.1.3 i2c_adapter_read()

```
int i2c_adapter_read (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )
```

Read from device on I2C adapter bus Here is the caller graph for this function:

30.154.1.4 i2c_adapter_reg_read()

```
static int i2c_adapter_reg_read (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * val )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.154.1.5 i2c_adapter_reg_write()

```
static int i2c_adapter_reg_write (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t val )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.154.1.6 i2c_adapter_remove()

```
void i2c_adapter_remove (
            struct i2c_adapter * ia )
```

Un-register I2C adapter Here is the call graph for this function:

30.154.1.7 i2c_adapter_write()

```
int i2c_adapter_write (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )
```

Send to device on I2C adapter bus Here is the caller graph for this function:

30.154.1.8 to_i2c_adapter()

```
static struct i2c_adapter * to_i2c_adapter (
            struct sbi_dlist * node )  [inline], [static]
```

Here is the caller graph for this function:

30.155 i2c.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 YADRO
 *
 * Authors:
 *   Nikita Shubin <n.shubin@yadro.com>
 */

#ifndef __I2C_H__
#define __I2C_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>

struct i2c_adapter {
    int id;

    int (*write)(struct i2c_adapter *ia, uint8_t addr, uint8_t reg,
            uint8_t *buffer, int len);

    int (*read)(struct i2c_adapter *ia, uint8_t addr, uint8_t reg,
            uint8_t *buffer, int len);

    struct sbi_dlist node;
};

static inline struct i2c_adapter *to_i2c_adapter(struct sbi_dlist *node)
{
    return container_of(node, struct i2c_adapter, node);
}

struct i2c_adapter *i2c_adapter_find(int id);

int i2c_adapter_add(struct i2c_adapter *ia);

void i2c_adapter_remove(struct i2c_adapter *ia);

int i2c_adapter_write(struct i2c_adapter *ia, uint8_t addr, uint8_t reg,
               uint8_t *buffer, int len);

int i2c_adapter_read(struct i2c_adapter *ia, uint8_t addr, uint8_t reg,
             uint8_t *buffer, int len);

```

```
static inline int i2c_adapter_reg_write(struct i2c_adapter *ia, uint8_t addr,
                  uint8_t reg, uint8_t val)
{
    return i2c_adapter_write(ia, addr, reg, &val, 1);
}

static inline int i2c_adapter_reg_read(struct i2c_adapter *ia, uint8_t addr,
                        uint8_t reg, uint8_t *val)
{
    uint8_t buf;
    int ret = i2c_adapter_read(ia, addr, reg, &buf, 1);

    if (ret)
        return ret;

    *val = buf;
    return 0;
}

#endif
```

30.156 include/sbi_utils/ipi/aclint_mswi.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for aclint_mswi.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct aclint_mswi_data

Macros

- #define ACLINT_MSWI_ALIGN 0x1000
- #define ACLINT_MSWI_SIZE 0x4000
- #define ACLINT_MSWI_MAX_HARTS 4095
- #define CLINT_MSWI_OFFSET 0x0000

Functions

- int aclint_mswi_cold_init (struct aclint_mswi_data *mswi)

30.156.1 Macro Definition Documentation

30.156.1.1 ACLINT_MSWI_ALIGN

```
#define ACLINT_MSWI_ALIGN 0x1000
```

30.156.1.2 ACLINT_MSWI_MAX_HARTS

```
#define ACLINT_MSWI_MAX_HARTS 4095
```

30.156.1.3 ACLINT_MSWI_SIZE

```
#define ACLINT_MSWI_SIZE 0x4000
```

30.156.1.4 CLINT_MSWI_OFFSET

```
#define CLINT_MSWI_OFFSET 0x0000
```

30.156.2 Function Documentation

30.156.2.1 aclint_mswi_cold_init()

```
int aclint_mswi_cold_init (
            struct aclint_mswi_data * mswi )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.157 aclint_mswi.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __IPI_ACLINT_MSWI_H__
#define __IPI_ACLINT_MSWI_H__

#include <sbi/sbi_types.h>

#define ACLINT_MSWI_ALIGN       0x1000
#define ACLINT_MSWI_SIZE        0x4000
#define ACLINT_MSWI_MAX_HARTS       4095

#define CLINT_MSWI_OFFSET       0x0000

struct aclint_mswi_data {
    /* Public details */
    unsigned long addr;
    unsigned long size;
    u32 first_hartid;
    u32 hart_count;
};

int aclint_mswi_cold_init(struct aclint_mswi_data *mswi);

#endif
```

30.158 include/sbi_utils/ipi/andes_plicsw.h File Reference

This graph shows which files directly or indirectly include this file:

Data Structures

- struct plicsw_data

Macros

- #define PLICSW_PRIORITY_BASE 0x4
- #define PLICSW_PENDING_BASE 0x1000
- #define PLICSW_ENABLE_BASE 0x2000
- #define PLICSW_ENABLE_STRIDE 0x80
- #define PLICSW_CONTEXT_BASE 0x200000
- #define PLICSW_CONTEXT_STRIDE 0x1000

- #define PLICSW_CONTEXT_CLAIM 0x4
- #define PLICSW_REGION_ALIGN 0x1000

Functions

- int plicsw_cold_ipi_init (struct plicsw_data *plicsw)

30.158.1 Macro Definition Documentation

30.158.1.1 PLICSW_CONTEXT_BASE

```
#define PLICSW_CONTEXT_BASE 0x200000
```

30.158.1.2 PLICSW_CONTEXT_CLAIM

```
#define PLICSW_CONTEXT_CLAIM 0x4
```

30.158.1.3 PLICSW_CONTEXT_STRIDE

```
#define PLICSW_CONTEXT_STRIDE 0x1000
```

30.158.1.4 PLICSW_ENABLE_BASE

```
#define PLICSW_ENABLE_BASE 0x2000
```

30.158.1.5 PLICSW_ENABLE_STRIDE

```
#define PLICSW_ENABLE_STRIDE 0x80
```

30.158.1.6 PLICSW_PENDING_BASE

```
#define PLICSW_PENDING_BASE 0x1000
```

30.158.1.7 PLICSW_PRIORITY_BASE

```
#define PLICSW_PRIORITY_BASE 0x4
```

30.158.1.8 PLICSW_REGION_ALIGN

```
#define PLICSW_REGION_ALIGN 0x1000
```

30.158.2 Function Documentation

30.158.2.1 plicsw_cold_ipi_init()

```
int plicsw_cold_ipi_init (
            struct plicsw_data * plicsw )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.159 andes_plicsw.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2022 Andes Technology Corporation
 *
 * Authors:
 *   Zong Li <zong@andestech.com>
 *   Nylon Chen <nylon7@andestech.com>
 *   Leo Yu-Chi Liang <ycliang@andestech.com>
 *   Yu Chien Peter Lin <peterlin@andestech.com>
 */

#ifndef _IPI_ANDES_PLICSW_H_
#define _IPI_ANDES_PLICSW_H_

#define PLICSW_PRIORITY_BASE    0x4

#define PLICSW_PENDING_BASE 0x1000

#define PLICSW_ENABLE_BASE  0x2000
#define PLICSW_ENABLE_STRIDE    0x80

#define PLICSW_CONTEXT_BASE 0x200000
#define PLICSW_CONTEXT_STRIDE   0x1000
#define PLICSW_CONTEXT_CLAIM    0x4

#define PLICSW_REGION_ALIGN 0x1000

struct plicsw_data {
    unsigned long addr;
    unsigned long size;
    uint32_t hart_count;
};

int plicsw_cold_ipi_init(struct plicsw_data *plicsw);

#endif /* _IPI_ANDES_PLICSW_H_ */
```

30.160 include/sbi_utils/ipi/fdt_ipi.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>
```

Include dependency graph for fdt_ipi.h:

This graph shows which files directly or indirectly include this file:

Functions

- static int fdt_ipi_init (void)

30.160.1 Function Documentation

30.160.1.1 fdt_ipi_init()

```
static int fdt_ipi_init (
            void  )  [inline], [static]
```

30.161 fdt_ipi.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
```

```
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __FDT_IPI_H__
#define __FDT_IPI_H__

#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>

#ifdef CONFIG_FDT_IPI

int fdt_ipi_init(void);

#else

static inline int fdt_ipi_init(void) { return 0; }

#endif

#endif
```

30.162 include/sbi_utils/irqchip/aplic.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_irqchip.h>
```

Include dependency graph for aplic.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct aplic_msicfg_data
- struct aplic_delegate_data
- struct aplic_data

Macros

- #define APLIC_MAX_DELEGATE 16

Functions

- int aplic_cold_irqchip_init (struct aplic_data ∗aplic)

30.162.1 Macro Definition Documentation

30.162.1.1 APLIC_MAX_DELEGATE

```
#define APLIC_MAX_DELEGATE 16
```

30.162.2 Function Documentation

30.162.2.1 aplic_cold_irqchip_init()

```
int aplic_cold_irqchip_init (
            struct aplic_data * aplic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.163 aplic.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Western Digital Corporation or its affiliates.
 * Copyright (c) 2022 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __IRQCHIP_APLIC_H__
#define __IRQCHIP_APLIC_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_irqchip.h>

#define APLIC_MAX_DELEGATE  16

struct aplic_msicfg_data {
    unsigned long lhxs;
    unsigned long lhxw;
    unsigned long hhxs;
    unsigned long hhxw;
    unsigned long base_addr;
};

struct aplic_delegate_data {
    u32 first_irq;
    u32 last_irq;
    u32 child_index;
};

struct aplic_data {
    /* Private members */
    struct sbi_irqchip_device irqchip;
    /* Public members */
    unsigned long addr;
    unsigned long size;
    unsigned long num_idc;
    unsigned long num_source;
    bool targets_mmode;
    bool has_msicfg_mmode;
    struct aplic_msicfg_data msicfg_mmode;
    bool has_msicfg_smode;
    struct aplic_msicfg_data msicfg_smode;
    struct aplic_delegate_data delegate[APLIC_MAX_DELEGATE];
};

int aplic_cold_irqchip_init(struct aplic_data *aplic);

#endif
```

30.164 include/sbi_utils/irqchip/fdt_irqchip.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for fdt_irqchip.h:

This graph shows which files directly or indirectly include this file:

Functions

- static int fdt_irqchip_init (void)

30.164.1 Function Documentation

30.164.1.1 fdt_irqchip_init()

```
static int fdt_irqchip_init (
            void  )  [inline], [static]
```

30.165 fdt_irqchip.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2020 Western Digital Corporation or its affiliates.
00005  *
00006  * Authors:
00007  *   Anup Patel <anup.patel@wdc.com>
00008  */
00009
00010 #ifndef __FDT_IRQCHIP_H__
00011 #define __FDT_IRQCHIP_H__
00012
00013 #include <sbi/sbi_types.h>
00014
00015 #ifdef CONFIG_FDT_IRQCHIP
```

```
struct fdt_irqchip {
    const struct fdt_match *match_table;
    int (*cold_init)(const void *fdt, int nodeoff, const struct fdt_match *match);
};

int fdt_irqchip_init(void);

#else

static inline int fdt_irqchip_init(void) { return 0; }

#endif

#endif
```

30.166 include/sbi_utils/irqchip/imsic.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for imsic.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct imsic_regs
- struct imsic_data

Macros

- #define IMSIC_MMIO_PAGE_SHIFT 12
- #define IMSIC_MMIO_PAGE_SZ (1UL << IMSIC_MMIO_PAGE_SHIFT)
- #define IMSIC_MAX_REGS 16

Functions

- static void imsic_local_irqchip_init (void)
- static int imsic_data_check (struct imsic_data *imsic)

30.166.1 Macro Definition Documentation

30.166.1.1 IMSIC_MAX_REGS

```
#define IMSIC_MAX_REGS 16
```

30.166.1.2 IMSIC_MMIO_PAGE_SHIFT

```
#define IMSIC_MMIO_PAGE_SHIFT 12
```

30.166.1.3 IMSIC_MMIO_PAGE_SZ

```
#define IMSIC_MMIO_PAGE_SZ (1UL << IMSIC_MMIO_PAGE_SHIFT)
```

30.166.2 Function Documentation

30.166.2.1 imsic_data_check()

```
static int imsic_data_check (
            struct imsic_data * imsic )  [inline], [static]
```

Here is the caller graph for this function:

30.166.2.2 imsic_local_irqchip_init()

```
static void imsic_local_irqchip_init (
            void  )  [inline], [static]
```

30.167 imsic.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Western Digital Corporation or its affiliates.
 * Copyright (c) 2022 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __IRQCHIP_IMSIC_H__
#define __IRQCHIP_IMSIC_H__

#include <sbi/sbi_types.h>

#define IMSIC_MMIO_PAGE_SHIFT       12
#define IMSIC_MMIO_PAGE_SZ      (1UL « IMSIC_MMIO_PAGE_SHIFT)

#define IMSIC_MAX_REGS          16

struct imsic_regs {
    unsigned long addr;
    unsigned long size;
```

```
};

struct imsic_data {
    bool targets_mmode;
    u32 guest_index_bits;
    u32 hart_index_bits;
    u32 group_index_bits;
    u32 group_index_shift;
    unsigned long num_ids;
    struct imsic_regs regs[IMSIC_MAX_REGS];
};

#ifdef CONFIG_IRQCHIP_IMSIC

int imsic_map_hartid_to_data(u32 hartid, struct imsic_data *imsic, int file);

struct imsic_data *imsic_get_data(u32 hartindex);

int imsic_get_target_file(u32 hartindex);

void imsic_local_irqchip_init(void);

int imsic_data_check(struct imsic_data *imsic);

int imsic_cold_irqchip_init(struct imsic_data *imsic);

#else

static inline void imsic_local_irqchip_init(void) { }

static inline int imsic_data_check(struct imsic_data *imsic) { return 0; }

#endif

#endif
```

30.168 include/sbi_utils/irqchip/plic.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_irqchip.h>
```

Include dependency graph for plic.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct plic_data

Macros

- #define PLIC_FLAG_ARIANE_BUG BIT(0)
- #define PLIC_FLAG_THEAD_DELEGATION BIT(1)
- #define PLIC_FLAG_ENABLE_PM BIT(2)
- #define PLIC_M_CONTEXT 0
- #define PLIC_S_CONTEXT 1
- #define PLIC_DATA_SIZE(__hart_count)
- #define PLIC_IE_WORDS(__p) ((__p)->num_src / 32 + 1)

Functions

- struct plic_data * plic_get (void)
- void plic_suspend (void)
- void plic_resume (void)
- int plic_cold_irqchip_init (struct plic_data *plic)

30.168.1 Macro Definition Documentation

30.168.1.1 PLIC_DATA_SIZE

```
#define PLIC_DATA_SIZE(
            __hart_count )
```

Value:

```
(sizeof(struct plic_data) + \
(__hart_count) * 2 * sizeof(s16))
```

30.168.1.2 PLIC_FLAG_ARIANE_BUG

```
#define PLIC_FLAG_ARIANE_BUG BIT(0)
```

Work around a bug on Ariane that requires enabling interrupts at boot

30.168.1.3 PLIC_FLAG_ENABLE_PM

```
#define PLIC_FLAG_ENABLE_PM BIT(2)
```

Allocate space for power management save/restore operations

30.168.1.4 PLIC_FLAG_THEAD_DELEGATION

```
#define PLIC_FLAG_THEAD_DELEGATION BIT(1)
```

PLIC must be delegated to S-mode like T-HEAD C906 and C910

30.168.1.5 PLIC_IE_WORDS

```
#define PLIC_IE_WORDS(
            __p ) ((__p)->num_src / 32 + 1)
```

30.168.1.6 PLIC_M_CONTEXT

```
#define PLIC_M_CONTEXT 0
```

30.168.1.7 PLIC_S_CONTEXT

```
#define PLIC_S_CONTEXT 1
```

30.168.2 Function Documentation

30.168.2.1 plic_cold_irqchip_init()

```
int plic_cold_irqchip_init (
            struct plic_data * plic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.168.2.2 plic_get()

```
struct plic_data * plic_get (
            void )
```

30.168.2.3 plic_resume()

```
void plic_resume (
            void )
```

Here is the call graph for this function:

30.168.2.4 plic_suspend()

```
void plic_suspend (
            void  )
```

Here is the call graph for this function:

30.169 plic.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __IRQCHIP_PLIC_H__
#define __IRQCHIP_PLIC_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_irqchip.h>

```

```
struct plic_data {
    /* Private members */
    struct sbi_irqchip_device irqchip;
    /* Public members */
    unsigned long addr;
    unsigned long size;
    unsigned long num_src;
    unsigned long flags;
    void *pm_data;
    s16 context_map[][2];
};

#define PLIC_FLAG_ARIANE_BUG        BIT(0)
#define PLIC_FLAG_THEAD_DELEGATION  BIT(1)
#define PLIC_FLAG_ENABLE_PM     BIT(2)

#define PLIC_M_CONTEXT          0
#define PLIC_S_CONTEXT          1

#define PLIC_DATA_SIZE(__hart_count)    (sizeof(struct plic_data) + \
                    (__hart_count) * 2 * sizeof(s16))

#define PLIC_IE_WORDS(__p)      ((__p)->num_src / 32 + 1)

struct plic_data *plic_get(void);

void plic_suspend(void);

void plic_resume(void);

int plic_cold_irqchip_init(struct plic_data *plic);

#endif
```

30.170 include/sbi_utils/mailbox/fdt_mailbox.h File Reference

```
#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/mailbox/mailbox.h>
```

Include dependency graph for fdt_mailbox.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_mailbox

Functions

- int fdt_mailbox_request_chan (const void ∗fdt, int nodeoff, int index, struct mbox_chan ∗∗out_chan)
- int fdt_mailbox_simple_xlate (struct mbox_controller ∗mbox, const struct fdt_phandle_args ∗pargs, u32 ∗out_chan_args)

30.170.1 Function Documentation

30.170.1.1 fdt_mailbox_request_chan()

```
int fdt_mailbox_request_chan (
            const void * fdt,
            int nodeoff,
            int index,
            struct mbox_chan ** out_chan )
```

Request a mailbox channel using "mboxes" DT property of client DT node Here is the call graph for this function:

Here is the caller graph for this function:

30.170.1.2 fdt_mailbox_simple_xlate()

```
int fdt_mailbox_simple_xlate (
            struct mbox_controller * mbox,
            const struct fdt_phandle_args * pargs,
            u32 * out_chan_args )
```

Simple xlate function to convert one mailbox FDT cell into channel args

30.171 fdt_mailbox.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2024 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __FDT_MAILBOX_H__
#define __FDT_MAILBOX_H__

#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/mailbox/mailbox.h>

struct fdt_phandle_args;

struct fdt_mailbox {
    struct fdt_driver driver;
    int (*xlate)(struct mbox_controller *mbox,
             const struct fdt_phandle_args *pargs,
             u32 *out_chan_args);
};

int fdt_mailbox_request_chan(const void *fdt, int nodeoff, int index,
                 struct mbox_chan **out_chan);

int fdt_mailbox_simple_xlate(struct mbox_controller *mbox,
                 const struct fdt_phandle_args *pargs,
                 u32 *out_chan_args);

#endif
```

30.172 include/sbi_utils/mailbox/mailbox.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
#include <sbi/riscv_atomic.h>
```

Include dependency graph for mailbox.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct mbox_chan
- struct mbox_xfer
- struct mbox_controller

Macros

- #define MBOX_CHAN_MAX_ARGS 2
- #define to_mbox_chan(__node) container_of((__node), struct mbox_chan, node)
- #define MBOX_XFER_SEQ (1UL << 0)
- #define mbox_xfer_init_tx(__p, __a, __t, __t_len, __t_tim)
- #define mbox_xfer_init_rx(__p, __a, __r, __r_len, __r_tim)
- #define mbox_xfer_init_txrx(__p, __a, __t, __t_len, __t_tim, __r, __r_len, __r_tim)
- #define mbox_xfer_set_sequence(__p, __seq)
- #define to_mbox_controller(__node) container_of((__node), struct mbox_controller, node)

Functions

- struct mbox_controller * mbox_controller_find (unsigned int id)
- int mbox_controller_add (struct mbox_controller *mbox)
- void mbox_controller_remove (struct mbox_controller *mbox)
- struct mbox_chan * mbox_controller_request_chan (struct mbox_controller *mbox, u32 *chan_args)
- void mbox_controller_free_chan (struct mbox_chan *chan)
- int mbox_chan_xfer (struct mbox_chan *chan, struct mbox_xfer *xfer)

- int mbox_chan_get_attribute (struct mbox_chan *chan, int attr_id, void *out_value)
- int mbox_chan_set_attribute (struct mbox_chan *chan, int attr_id, void *new_value)

30.172.1 Macro Definition Documentation

30.172.1.1 MBOX_CHAN_MAX_ARGS

```
#define MBOX_CHAN_MAX_ARGS 2
```

Arguments (or parameters) to identify a mailbox channel within a mailbox controller.

30.172.1.2 mbox_xfer_init_rx

```
#define mbox_xfer_init_rx(
            __p,
            __a,
            __r,
            __r_len,
            __r_tim )
```

Value:

```
do {                                    \
    (__p)->flags = 0;                       \
    (__p)->args = (__a);                        \
    (__p)->tx = NULL;                       \
    (__p)->tx_len = 0;                      \
    (__p)->tx_timeout = 0;                      \
    (__p)->rx = (__r);                      \
    (__p)->rx_len = (__r_len);                  \
    (__p)->rx_timeout = (__r_tim);                  \
} while (0)
```

30.172.1.3 mbox_xfer_init_tx

```
#define mbox_xfer_init_tx(
            __p,
            __a,
            __t,
            __t_len,
            __t_tim )
```

Value:

```
do {                                    \
    (__p)->flags = 0;                       \
    (__p)->args = (__a);                        \
    (__p)->tx = (__t);                      \
    (__p)->tx_len = (__t_len);                  \
    (__p)->tx_timeout = (__t_tim);                  \
    (__p)->rx = NULL;                       \
    (__p)->rx_len = 0;                      \
    (__p)->rx_timeout = 0;                      \
} while (0)
```

30.172.1.4 mbox_xfer_init_txrx

```
#define mbox_xfer_init_txrx(
            __p,
            __a,
            __t,
            __t_len,
            __t_tim,
            __r,
            __r_len,
            __r_tim )
```

Value:

```
do {                                    \
    (__p)->flags = 0;                       \
    (__p)->args = (__a);                        \
    (__p)->tx = (__t);                      \
    (__p)->tx_len = (__t_len);                  \
    (__p)->tx_timeout = (__t_tim);                  \
```

```
    (__p)->rx = (__r);                          \
    (__p)->rx_len = (__r_len);                      \
    (__p)->rx_timeout = (__r_tim);                      \
} while (0)
```

30.172.1.5 MBOX_XFER_SEQ

```
#define MBOX_XFER_SEQ (1UL << 0)
```

30.172.1.6 mbox_xfer_set_sequence

```
#define mbox_xfer_set_sequence(
            __p,
            __seq )
```

Value:

```
do {                                        \
    (__p)->flags |= MBOX_XFER_SEQ;                  \
    (__p)->seq = (__seq);                       \
} while (0)
```

30.172.1.7 to_mbox_chan

```
#define to_mbox_chan(
            __node )  container_of((__node), struct mbox_chan, node)
```

30.172.1.8 to_mbox_controller

```
#define to_mbox_controller(
            __node )  container_of((__node), struct mbox_controller, node)
```

30.172.2 Function Documentation

30.172.2.1 mbox_chan_get_attribute()

```
int mbox_chan_get_attribute (
            struct mbox_chan * chan,
            int attr_id,
            void * out_value )
```

Get an attribute of mailbox channel

30.172.2.2 mbox_chan_set_attribute()

```
int mbox_chan_set_attribute (
            struct mbox_chan * chan,
            int attr_id,
            void * new_value )
```

Set an attribute of mailbox channel

30.172.2.3 mbox_chan_xfer()

```
int mbox_chan_xfer (
            struct mbox_chan * chan,
            struct mbox_xfer * xfer )
```

Data transfer over mailbox channel Here is the call graph for this function:

Here is the caller graph for this function:

30.172.2.4 mbox_controller_add()

```
int mbox_controller_add (
            struct mbox_controller * mbox )
```

Register mailbox controller Here is the call graph for this function:

Here is the caller graph for this function:

30.172.2.5 mbox_controller_find()

```
struct mbox_controller * mbox_controller_find (
            unsigned int id )
```

Find a registered mailbox controller Here is the caller graph for this function:

30.172.2.6 mbox_controller_free_chan()

```
void mbox_controller_free_chan (
            struct mbox_chan * chan )
```

Free a mailbox channel Here is the call graph for this function:

Here is the caller graph for this function:

30.172.2.7 mbox_controller_remove()

```
void mbox_controller_remove (
            struct mbox_controller * mbox )
```

Un-register mailbox controller Here is the call graph for this function:

Here is the caller graph for this function:

30.172.2.8 mbox_controller_request_chan()

```
struct mbox_chan * mbox_controller_request_chan (
            struct mbox_controller * mbox,
            u32 * chan_args )
```

Request a mailbox channel Here is the call graph for this function:

Here is the caller graph for this function:

30.173 mailbox.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2024 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __MAILBOX_H__
#define __MAILBOX_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
#include <sbi/riscv_atomic.h>

struct mbox_chan {
    struct sbi_dlist node;
    struct mbox_controller *mbox;
#define MBOX_CHAN_MAX_ARGS  2
    u32 chan_args[MBOX_CHAN_MAX_ARGS];
};

#define to_mbox_chan(__node)    \
    container_of((__node), struct mbox_chan, node)

struct mbox_xfer {
#define MBOX_XFER_SEQ           (1UL << 0)
    unsigned long flags;
    void *args;
    long seq;
    void *tx;
    unsigned long tx_len;
    unsigned long tx_timeout;
    void *rx;
    unsigned long rx_len;
```

```
    unsigned long rx_timeout;
};

#define mbox_xfer_init_tx(__p, __a, __t, __t_len, __t_tim)      \
do {                                                            \
    (__p)->flags = 0;                                           \
    (__p)->args = (__a);                                        \
    (__p)->tx = (__t);                                          \
    (__p)->tx_len = (__t_len);                                  \
    (__p)->tx_timeout = (__t_tim);                              \
    (__p)->rx = NULL;                                           \
    (__p)->rx_len = 0;                                          \
    (__p)->rx_timeout = 0;                                      \
} while (0)

#define mbox_xfer_init_rx(__p, __a, __r, __r_len, __r_tim)      \
do {                                                            \
    (__p)->flags = 0;                                           \
    (__p)->args = (__a);                                        \
    (__p)->tx = NULL;                                           \
    (__p)->tx_len = 0;                                          \
    (__p)->tx_timeout = 0;                                      \
    (__p)->rx = (__r);                                          \
    (__p)->rx_len = (__r_len);                                  \
    (__p)->rx_timeout = (__r_tim);                              \
} while (0)

#define mbox_xfer_init_txrx(__p, __a, __t, __t_len, __t_tim, __r, __r_len, __r_tim)\
do {                                                            \
    (__p)->flags = 0;                                           \
    (__p)->args = (__a);                                        \
    (__p)->tx = (__t);                                          \
    (__p)->tx_len = (__t_len);                                  \
    (__p)->tx_timeout = (__t_tim);                              \
    (__p)->rx = (__r);                                          \
    (__p)->rx_len = (__r_len);                                  \
    (__p)->rx_timeout = (__r_tim);                              \
} while (0)

#define mbox_xfer_set_sequence(__p, __seq)              \
do {                                                            \
    (__p)->flags |= MBOX_XFER_SEQ;                              \
    (__p)->seq = (__seq);                                       \
} while (0)

struct mbox_controller {
    struct sbi_dlist node;
    atomic_t xfer_next_seq;
    /* List of mailbox channels */
    struct sbi_dlist chan_list;
    unsigned int id;
    unsigned int max_xfer_len;
    void *driver;
    struct mbox_chan *(*request_chan)(struct mbox_controller *mbox,
                      u32 *chan_args);
    void (*free_chan)(struct mbox_controller *mbox,
              struct mbox_chan *chan);
    int (*xfer)(struct mbox_chan *chan, struct mbox_xfer *xfer);
    int (*get_attribute)(struct mbox_chan *chan, int attr_id, void *out_value);
    int (*set_attribute)(struct mbox_chan *chan, int attr_id, void *new_value);
};

#define to_mbox_controller(__node)  \
    container_of((__node), struct mbox_controller, node)

struct mbox_controller *mbox_controller_find(unsigned int id);

int mbox_controller_add(struct mbox_controller *mbox);

void mbox_controller_remove(struct mbox_controller *mbox);

struct mbox_chan *mbox_controller_request_chan(struct mbox_controller *mbox,
                           u32 *chan_args);

void mbox_controller_free_chan(struct mbox_chan *chan);

int mbox_chan_xfer(struct mbox_chan *chan, struct mbox_xfer *xfer);

int mbox_chan_get_attribute(struct mbox_chan *chan, int attr_id, void *out_value);

int mbox_chan_set_attribute(struct mbox_chan *chan, int attr_id, void *new_value);

#endif
```

30.174 include/sbi_utils/mailbox/rpmi_mailbox.h File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/mailbox/rpmi_msgprot.h>
```

Include dependency graph for rpmi_mailbox.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define rpmi_u32_count(__var) (sizeof(__var) / sizeof(u32))

Functions

- int rpmi_xlate_error (enum rpmi_error error)
- int rpmi_normal_request_with_status (struct mbox_chan *chan, u32 service_id, void *req, u32 req_words, u32 req_endian_words, void *resp, u32 resp_words, u32 resp_endian_words)
- int rpmi_posted_request (struct mbox_chan *chan, u32 service_id, void *req, u32 req_words, u32 req_↩ endian_words)

30.174.1 Macro Definition Documentation

30.174.1.1 rpmi_u32_count

```
#define rpmi_u32_count(
            __var ) (sizeof(__var) / sizeof(u32))
```

30.174.2 Function Documentation

30.174.2.1 rpmi_normal_request_with_status()

```
int rpmi_normal_request_with_status (
            struct mbox_chan * chan,
            u32 service_id,
            void * req,
            u32 req_words,
            u32 req_endian_words,
            void * resp,
            u32 resp_words,
            u32 resp_endian_words )
```

Typical RPMI normal request with at least status code in response Here is the call graph for this function:

Here is the caller graph for this function:

30.174.2.2 rpmi_posted_request()

```
int rpmi_posted_request (
            struct mbox_chan * chan,
            u32 service_id,
            void * req,
            u32 req_words,
            u32 req_endian_words )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.174.2.3 rpmi_xlate_error()

```
int rpmi_xlate_error (
            enum rpmi_error error )
```

Convert RPMI error to SBI error Here is the caller graph for this function:

30.175 rpmi_mailbox.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __RPMI_MAILBOX_H__
#define __RPMI_MAILBOX_H__

#include <sbi/sbi_error.h>
#include <sbi_utils/mailbox/rpmi_msgprot.h>

#define rpmi_u32_count(__var)   (sizeof(__var) / sizeof(u32))

```

```
int rpmi_xlate_error(enum rpmi_error error);

int rpmi_normal_request_with_status(
            struct mbox_chan *chan, u32 service_id,
            void *req, u32 req_words, u32 req_endian_words,
            void *resp, u32 resp_words, u32 resp_endian_words);

/* RPMI posted request which is without any response*/
int rpmi_posted_request(
        struct mbox_chan *chan, u32 service_id,
        void *req, u32 req_words, u32 req_endian_words);

#endif /* !__RPMI_MAILBOX_H__ */
```

30.176 include/sbi_utils/mailbox/rpmi_msgprot.h File Reference

```
#include <sbi/sbi_byteorder.h>
#include <sbi/sbi_error.h>
```

Include dependency graph for rpmi_msgprot.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct rpmi_message_header
- struct rpmi_message
- struct rpmi_message_args

- struct rpmi_enable_notification_req
- struct rpmi_enable_notification_resp
- struct rpmi_base_get_attributes_resp
- struct rpmi_base_get_platform_info_resp
- struct rpmi_sysrst_get_reset_attributes_resp
- struct rpmi_syssusp_get_attr_req
- struct rpmi_syssusp_get_attr_resp
- struct rpmi_syssusp_suspend_req
- struct rpmi_syssusp_suspend_resp
- struct rpmi_hsm_hart_start_req
- struct rpmi_hsm_hart_start_resp
- struct rpmi_hsm_hart_stop_req
- struct rpmi_hsm_hart_stop_resp
- struct rpmi_hsm_hart_susp_req
- struct rpmi_hsm_hart_susp_resp
- struct rpmi_hsm_get_hart_status_req
- struct rpmi_hsm_get_hart_status_resp
- struct rpmi_hsm_get_hart_list_req
- struct rpmi_hsm_get_hart_list_resp
- struct rpmi_hsm_get_susp_types_req
- struct rpmi_hsm_get_susp_types_resp
- struct rpmi_hsm_get_susp_info_req
- struct rpmi_hsm_get_susp_info_resp
- struct rpmi_cppc_probe_req
- struct rpmi_cppc_probe_resp
- struct rpmi_cppc_read_reg_req
- struct rpmi_cppc_read_reg_resp
- struct rpmi_cppc_write_reg_req
- struct rpmi_cppc_write_reg_resp
- struct rpmi_cppc_get_fastchan_offset_req
- struct rpmi_cppc_get_fastchan_offset_resp
- struct rpmi_cppc_get_fastchan_region_resp
- struct rpmi_cppc_hart_list_req
- struct rpmi_cppc_hart_list_resp
- struct rpmi_clock_get_num_clocks_resp
- struct rpmi_clock_get_attributes_req
- struct rpmi_clock_get_attributes_resp
- struct rpmi_clock_get_supported_rates_req
- struct rpmi_clock_get_supported_rates_resp
- struct rpmi_clock_set_config_req
- struct rpmi_clock_set_config_resp
- struct rpmi_clock_get_config_req
- struct rpmi_clock_get_config_resp
- struct rpmi_clock_set_rate_req
- struct rpmi_clock_set_rate_resp
- struct rpmi_clock_get_rate_req
- struct rpmi_clock_get_rate_resp

Macros

- #define RPMI_MSG_HDR_OFFSET (0x0)
- #define RPMI_MSG_HDR_SIZE (8)
- #define RPMI_MSG_SERVICEGROUP_ID_OFFSET (0x0)
- #define RPMI_MSG_SERVICEGROUP_ID_SIZE (2)
- #define RPMI_MSG_SERVICE_ID_OFFSET (0x2)
- #define RPMI_MSG_SERVICE_ID_SIZE (1)
- #define RPMI_MSG_FLAGS_OFFSET (0x3)
- #define RPMI_MSG_FLAGS_SIZE (1)
- #define RPMI_MSG_FLAGS_TYPE_POS (0U)
- #define RPMI_MSG_FLAGS_TYPE_MASK 0x7
- #define RPMI_MSG_FLAGS_TYPE ((0x7) << RPMI_MSG_FLAGS_TYPE_POS)
- #define RPMI_MSG_FLAGS_DOORBELL_POS (3U)
- #define RPMI_MSG_FLAGS_DOORBELL_MASK 0x1
- #define RPMI_MSG_FLAGS_DOORBELL ((0x1) << RPMI_MSG_FLAGS_DOORBELL_POS)
- #define RPMI_MSG_DATALEN_OFFSET (0x4)
- #define RPMI_MSG_DATALEN_SIZE (2)
- #define RPMI_MSG_TOKEN_OFFSET (0x6)
- #define RPMI_MSG_TOKEN_SIZE (2)
- #define RPMI_MSG_TOKEN_MASK (0xffffU)
- #define RPMI_MSG_DATA_OFFSET (RPMI_MSG_HDR_SIZE)
- #define RPMI_MSG_DATA_SIZE(__slot_size) ((__slot_size) - RPMI_MSG_HDR_SIZE)
- #define RPMI_SLOT_SIZE_MIN (64)
- #define RPMI_NAME_CHARS_MAX (16)
- #define RPMI_QUEUE_HEAD_SLOT 0
- #define RPMI_QUEUE_TAIL_SLOT 1
- #define RPMI_QUEUE_HEADER_SLOTS 2
- #define RPMI_DEF_TX_TIMEOUT 20
- #define RPMI_DEF_RX_TIMEOUT 20
- #define RPMI_VERSION(__major, __minor) (((__major) << 16) | (__minor))
- #define RPMI_MSG_FLAGS_NO_TX (1U << 0)
- #define RPMI_MSG_FLAGS_NO_RX (1U << 1)
- #define RPMI_MSG_FLAGS_NO_RX_TOKEN (1U << 2)
- #define RPMI_BASE_FLAGS_F0_PRIVILEGE (1U << 2)
- #define RPMI_BASE_FLAGS_F0_EV_NOTIFY (1U << 1)
- #define RPMI_BASE_FLAGS_F0_MSI_EN (1U)
- #define RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS (1)
- #define RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_MASK (1U << RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS
- #define RPMI_SYSSUSP_ATTRS_FLAGS_RESUMEADDR (1U << 1)
- #define RPMI_SYSSUSP_ATTRS_FLAGS_SUSPENDTYPE 1U
- #define RPMI_HSM_SUSPEND_INFO_FLAGS_TIMER_STOP 1U
- #define RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS 3
- #define RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MASK (3U << RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS
- #define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS 1
- #define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_MASK (3U << RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_W
- #define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_SUPPORTED (1U << 0)
- #define RPMI_CLOCK_FLAGS_FORMAT_POS 30
- #define RPMI_CLOCK_FLAGS_FORMAT_MASK (3U << RPMI_CLOCK_FLAGS_CLOCK_FORMAT_↩ POS)
- #define RPMI_CLOCK_FLAGS_FORMAT_DISCRETE 0
- #define RPMI_CLOCK_FLAGS_FORMAT_LINEAR 1
- #define RPMI_CLOCK_CONFIG_ENABLE (1U << 0)
- #define RPMI_CLOCK_SET_RATE_FLAGS_MASK (3U << 0)
- #define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_DOWN 0
- #define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_UP 1
- #define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_PLAT 2

Enumerations

- enum rpmi_message_type { RPMI_MSG_NORMAL_REQUEST = 0x0 , RPMI_MSG_POSTED_REQUEST = 0x1 , RPMI_MSG_ACKNOWLDGEMENT = 0x2 , RPMI_MSG_NOTIFICATION = 0x3 }
- enum rpmi_error { RPMI_SUCCESS = 0 , RPMI_ERR_FAILED = -1 , RPMI_ERR_NOTSUPP = -2 , RPMI_ERR_INVALID_PARAM = -3 , RPMI_ERR_DENIED = -4 , RPMI_ERR_INVALID_ADDR = -5 , RPMI_ERR_ALREADY = -6 , RPMI_ERR_EXTENSION = -7 , RPMI_ERR_HW_FAULT = -8 , RPMI_ERR_BUSY = -9 , RPMI_ERR_INVALID_STATE = -10 , RPMI_ERR_BAD_RANGE = -11 , RPMI_ERR_TIMEOUT = -12 , RPMI_ERR_IO = -13 , RPMI_ERR_NO_DATA = -14 , RPMI_ERR_RESERVED_START = -15 , RPMI_ERR_RESERVED_END = -127 , RPMI_ERR_VENDOR_START = -128 }
- enum rpmi_servicegroup_id { RPMI_SRVGRP_ID_MIN = 0 , RPMI_SRVGRP_BASE = 0x0001 , RPMI_SRVGRP_SYSTEM_RESET = 0x0002 , RPMI_SRVGRP_SYSTEM_SUSPEND = 0x0003 , RPMI_SRVGRP_HSM = 0x0004 , RPMI_SRVGRP_CPPC = 0x0005 , RPMI_SRVGRP_CLOCK = 0x0007 , RPMI_SRVGRP_ID_MAX_COUNT , RPMI_SRVGRP_RESERVE_START = RPMI_SRVGRP_ID_MAX_COUNT , RPMI_SRVGRP_RESERVE_END = 0x7FFF , RPMI_SRVGRP_VENDOR_START = 0x8000 , RPMI_SRVGRP_VENDOR_END = 0xFFFF }
- enum rpmi_base_service_id { RPMI_BASE_SRV_ENABLE_NOTIFICATION = 0x01 , RPMI_BASE_SRV_GET_IMPLEMENTATION_VERSION = 0x02 , RPMI_BASE_SRV_GET_IMPLEMENTATION_IDN = 0x03 , RPMI_BASE_SRV_GET_SPEC_VERSION = 0x04 , RPMI_BASE_SRV_GET_PLATFORM_INFO = 0x05 , RPMI_BASE_SRV_PROBE_SERVICE_GROUP = 0x06 , RPMI_BASE_SRV_GET_ATTRIBUTES = 0x07 , RPMI_BASE_SRV_SET_MSI = 0x08 }
- enum rpmi_base_context_priv_level { RPMI_BASE_CONTEXT_PRIV_S_MODE , RPMI_BASE_CONTEXT_PRIV_M_MODE }
- enum rpmi_system_reset_service_id { RPMI_SYSRST_SRV_ENABLE_NOTIFICATION = 0x01 , RPMI_SYSRST_SRV_GET_ATTRIBUTES = 0x02 , RPMI_SYSRST_SRV_SYSTEM_RESET = 0x03 , RPMI_SYSRST_SRV_ID_MAX_COUNT }
- enum rpmi_sysrst_reset_type { RPMI_SYSRST_TYPE_SHUTDOWN = 0x0 , RPMI_SYSRST_TYPE_COLD_REBOOT = 0x1 , RPMI_SYSRST_TYPE_WARM_REBOOT = 0x2 , RPMI_SYSRST_TYPE_MAX }
- enum rpmi_system_suspend_service_id { RPMI_SYSSUSP_SRV_ENABLE_NOTIFICATION = 0x01 , RPMI_SYSSUSP_SRV_GET_ATTRIBUTES = 0x02 , RPMI_SYSSUSP_SRV_SYSTEM_SUSPEND = 0x03 , RPMI_SYSSUSP_SRV_ID_MAX_COUNT }
- enum rpmi_hsm_service_id { RPMI_HSM_SRV_ENABLE_NOTIFICATION = 0x01 , RPMI_HSM_SRV_GET_HART_STATUS = 0x02 , RPMI_HSM_SRV_GET_HART_LIST = 0x03 , RPMI_HSM_SRV_GET_SUSPEND_TYPES = 0x04 , RPMI_HSM_SRV_GET_SUSPEND_INFO = 0x05 , RPMI_HSM_SRV_HART_START = 0x06 , RPMI_HSM_SRV_HART_STOP = 0x07 , RPMI_HSM_SRV_HART_SUSPEND = 0x08 , RPMI_HSM_SRV_ID_MAX = 0x09 }
- enum rpmi_cppc_service_id { RPMI_CPPC_SRV_ENABLE_NOTIFICATION = 0x01 , RPMI_CPPC_SRV_PROBE_REG = 0x02 , RPMI_CPPC_SRV_READ_REG = 0x03 , RPMI_CPPC_SRV_WRITE_REG = 0x04 , RPMI_CPPC_SRV_GET_FAST_CHANNEL_REGION = 0x05 , RPMI_CPPC_SRV_GET_FAST_CHANNEL_OFFSET = 0x06 , RPMI_CPPC_SRV_GET_HART_LIST = 0x07 , RPMI_CPPC_SRV_MAX_COUNT }
- enum rpmi_cppc_fast_channel_db_width { RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_8 = 0x0 , RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_16 = 0x1 , RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_32 = 0x2 , RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_64 = 0x3 }
- enum rpmi_cppc_fast_channel_cppc_mode { RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_PASSIVE = 0x0 , RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_ACTIVE = 0x1 , RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MAX_I }
- enum rpmi_clock_service_id { RPMI_CLOCK_SRV_ENABLE_NOTIFICATION = 0x01 , RPMI_CLOCK_SRV_GET_NUM_CLOCKS = 0x02 , RPMI_CLOCK_SRV_GET_ATTRIBUTES = 0x03 , RPMI_CLOCK_SRV_GET_SUPPORTED_RATES = 0x04 ,

RPMI_CLOCK_SRV_SET_CONFIG = 0x05 , RPMI_CLOCK_SRV_GET_CONFIG = 0x06 , RPMI_CLOCK_SRV_SET_RATE = 0x07 , RPMI_CLOCK_SRV_GET_RATE = 0x08 ,
RPMI_CLOCK_SRV_MAX_COUNT }

Variables

- struct rpmi_message_header __packed

30.176.1 Macro Definition Documentation

30.176.1.1 RPMI_BASE_FLAGS_F0_EV_NOTIFY

```
#define RPMI_BASE_FLAGS_F0_EV_NOTIFY (1U << 1)
```

30.176.1.2 RPMI_BASE_FLAGS_F0_MSI_EN

```
#define RPMI_BASE_FLAGS_F0_MSI_EN (1U)
```

30.176.1.3 RPMI_BASE_FLAGS_F0_PRIVILEGE

```
#define RPMI_BASE_FLAGS_F0_PRIVILEGE (1U << 2)
```

30.176.1.4 RPMI_CLOCK_CONFIG_ENABLE

```
#define RPMI_CLOCK_CONFIG_ENABLE (1U << 0)
```

30.176.1.5 RPMI_CLOCK_FLAGS_FORMAT_DISCRETE

```
#define RPMI_CLOCK_FLAGS_FORMAT_DISCRETE 0
```

30.176.1.6 RPMI_CLOCK_FLAGS_FORMAT_LINEAR

```
#define RPMI_CLOCK_FLAGS_FORMAT_LINEAR 1
```

30.176.1.7 RPMI_CLOCK_FLAGS_FORMAT_MASK

```
#define RPMI_CLOCK_FLAGS_FORMAT_MASK  (3U << RPMI_CLOCK_FLAGS_CLOCK_FORMAT_POS)
```

30.176.1.8 RPMI_CLOCK_FLAGS_FORMAT_POS

```
#define RPMI_CLOCK_FLAGS_FORMAT_POS 30
```

30.176.1.9 RPMI_CLOCK_SET_RATE_FLAGS_MASK

```
#define RPMI_CLOCK_SET_RATE_FLAGS_MASK (3U << 0)
```

30.176.1.10 RPMI_CLOCK_SET_RATE_FLAGS_ROUND_DOWN

```
#define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_DOWN 0
```

30.176.1.11 RPMI_CLOCK_SET_RATE_FLAGS_ROUND_PLAT

```
#define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_PLAT 2
```

30.176.1.12 RPMI_CLOCK_SET_RATE_FLAGS_ROUND_UP

```
#define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_UP 1
```

30.176.1.13 RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MASK

```
#define RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MASK  (3U << RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS)
```

30.176.1.14 RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS

```
#define RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS 3
```

30.176.1.15 RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_SUPPORTED

```
#define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_SUPPORTED (1U << 0)
```

30.176.1.16 RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_MASK

```
#define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_MASK  (3U << RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS)
```

30.176.1.17 RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS

```
#define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS 1
```

30.176.1.18 RPMI_DEF_RX_TIMEOUT

```
#define RPMI_DEF_RX_TIMEOUT 20
```

30.176.1.19 RPMI_DEF_TX_TIMEOUT

```
#define RPMI_DEF_TX_TIMEOUT 20
```

Default timeout values

30.176.1.20 RPMI_HSM_SUSPEND_INFO_FLAGS_TIMER_STOP

```
#define RPMI_HSM_SUSPEND_INFO_FLAGS_TIMER_STOP 1U
```

30.176.1.21 RPMI_MSG_DATA_OFFSET

```
#define RPMI_MSG_DATA_OFFSET (RPMI_MSG_HDR_SIZE)
```

Data field byte offset

30.176.1.22 RPMI_MSG_DATA_SIZE

```
#define RPMI_MSG_DATA_SIZE(
            __slot_size ) ((__slot_size) - RPMI_MSG_HDR_SIZE)
```

Data field size in bytes

30.176.1.23 RPMI_MSG_DATALEN_OFFSET

```
#define RPMI_MSG_DATALEN_OFFSET (0x4)
```

Data length field byte offset

30.176.1.24 RPMI_MSG_DATALEN_SIZE

```
#define RPMI_MSG_DATALEN_SIZE (2)
```

Data length field size in bytes

30.176.1.25 RPMI_MSG_FLAGS_DOORBELL

```
#define RPMI_MSG_FLAGS_DOORBELL  ((0x1) << RPMI_MSG_FLAGS_DOORBELL_POS)
```

30.176.1.26 RPMI_MSG_FLAGS_DOORBELL_MASK

```
#define RPMI_MSG_FLAGS_DOORBELL_MASK 0x1
```

30.176.1.27 RPMI_MSG_FLAGS_DOORBELL_POS

```
#define RPMI_MSG_FLAGS_DOORBELL_POS (3U)
```

30.176.1.28 RPMI_MSG_FLAGS_NO_RX

```
#define RPMI_MSG_FLAGS_NO_RX (1U << 1)
```

30.176.1.29 RPMI_MSG_FLAGS_NO_RX_TOKEN

```
#define RPMI_MSG_FLAGS_NO_RX_TOKEN (1U << 2)
```

30.176.1.30 RPMI_MSG_FLAGS_NO_TX

```
#define RPMI_MSG_FLAGS_NO_TX (1U << 0)
```

30.176.1.31 RPMI_MSG_FLAGS_OFFSET

```
#define RPMI_MSG_FLAGS_OFFSET (0x3)
```

Flags field byte offset

30.176.1.32 RPMI_MSG_FLAGS_SIZE

```
#define RPMI_MSG_FLAGS_SIZE (1)
```

Flags field size in bytes

30.176.1.33 RPMI_MSG_FLAGS_TYPE

```
#define RPMI_MSG_FLAGS_TYPE  ((0x7) << RPMI_MSG_FLAGS_TYPE_POS)
```

30.176.1.34 RPMI_MSG_FLAGS_TYPE_MASK

```
#define RPMI_MSG_FLAGS_TYPE_MASK 0x7
```

30.176.1.35 RPMI_MSG_FLAGS_TYPE_POS

```
#define RPMI_MSG_FLAGS_TYPE_POS (0U)
```

30.176.1.36 RPMI_MSG_HDR_OFFSET

```
#define RPMI_MSG_HDR_OFFSET (0x0)
```

Message Header byte offset

30.176.1.37 RPMI_MSG_HDR_SIZE

```
#define RPMI_MSG_HDR_SIZE (8)
```

Message Header Size in bytes

30.176.1.38 RPMI_MSG_SERVICE_ID_OFFSET

```
#define RPMI_MSG_SERVICE_ID_OFFSET (0x2)
```

Service ID field byte offset

30.176.1.39 RPMI_MSG_SERVICE_ID_SIZE

```
#define RPMI_MSG_SERVICE_ID_SIZE (1)
```

Service ID field size in bytes

30.176.1.40 RPMI_MSG_SERVICEGROUP_ID_OFFSET

```
#define RPMI_MSG_SERVICEGROUP_ID_OFFSET (0x0)
```

ServiceGroup ID field byte offset

30.176.1.41 RPMI_MSG_SERVICEGROUP_ID_SIZE

```
#define RPMI_MSG_SERVICEGROUP_ID_SIZE (2)
```

ServiceGroup ID field size in bytes

30.176.1.42 RPMI_MSG_TOKEN_MASK

```
#define RPMI_MSG_TOKEN_MASK (0xffffU)
```

Token field mask

30.176.1.43 RPMI_MSG_TOKEN_OFFSET

```
#define RPMI_MSG_TOKEN_OFFSET (0x6)
```

Token field byte offset

30.176.1.44 RPMI_MSG_TOKEN_SIZE

```
#define RPMI_MSG_TOKEN_SIZE (2)
```

Token field size in bytes

30.176.1.45 RPMI_NAME_CHARS_MAX

```
#define RPMI_NAME_CHARS_MAX (16)
```

Name length of 16 characters

30.176.1.46 RPMI_QUEUE_HEAD_SLOT

```
#define RPMI_QUEUE_HEAD_SLOT 0
```

Queue layout

30.176.1.47 RPMI_QUEUE_HEADER_SLOTS

```
#define RPMI_QUEUE_HEADER_SLOTS 2
```

30.176.1.48 RPMI_QUEUE_TAIL_SLOT

```
#define RPMI_QUEUE_TAIL_SLOT 1
```

30.176.1.49 RPMI_SLOT_SIZE_MIN

```
#define RPMI_SLOT_SIZE_MIN (64)
```

Minimum slot size in bytes

30.176.1.50 RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_MASK

```
#define RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_MASK  (1U << RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS)
```

30.176.1.51 RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS

```
#define RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS (1)
```

30.176.1.52 RPMI_SYSSUSP_ATTRS_FLAGS_RESUMEADDR

```
#define RPMI_SYSSUSP_ATTRS_FLAGS_RESUMEADDR (1U << 1)
```

30.176.1.53 RPMI_SYSSUSP_ATTRS_FLAGS_SUSPENDTYPE

```
#define RPMI_SYSSUSP_ATTRS_FLAGS_SUSPENDTYPE 1U
```

30.176.1.54 RPMI_VERSION

```
#define RPMI_VERSION(
            __major,
            __minor ) (((__major) << 16) | (__minor))
```

Common macro to generate composite version from major and minor version numbers.
RPMI has Specification version, Implementation version Service group versions which follow the same versioning encoding as below.

30.176.2 Enumeration Type Documentation

30.176.2.1 rpmi_base_context_priv_level

enum rpmi_base_context_priv_level

Enumerator

RPMI_BASE_CONTEXT_PRIV_S_MODE	
RPMI_BASE_CONTEXT_PRIV_M_MODE	

30.176.2.2 rpmi_base_service_id

enum rpmi_base_service_id
RPMI Base ServiceGroup Service IDs

Enumerator

RPMI_BASE_SRV_ENABLE_NOTIFICATION	
RPMI_BASE_SRV_GET_IMPLEMENTATION_VERSION	
RPMI_BASE_SRV_GET_IMPLEMENTATION_IDN	
RPMI_BASE_SRV_GET_SPEC_VERSION	
RPMI_BASE_SRV_GET_PLATFORM_INFO	
RPMI_BASE_SRV_PROBE_SERVICE_GROUP	
RPMI_BASE_SRV_GET_ATTRIBUTES	
RPMI_BASE_SRV_SET_MSI	

30.176.2.3 rpmi_clock_service_id

enum rpmi_clock_service_id
RPMI Clock ServiceGroup Service IDs

Enumerator

RPMI_CLOCK_SRV_ENABLE_NOTIFICATION	
RPMI_CLOCK_SRV_GET_NUM_CLOCKS	
RPMI_CLOCK_SRV_GET_ATTRIBUTES	
RPMI_CLOCK_SRV_GET_SUPPORTED_RATES	
RPMI_CLOCK_SRV_SET_CONFIG	
RPMI_CLOCK_SRV_GET_CONFIG	
RPMI_CLOCK_SRV_SET_RATE	
RPMI_CLOCK_SRV_GET_RATE	
RPMI_CLOCK_SRV_MAX_COUNT	

30.176.2.4 rpmi_cppc_fast_channel_cppc_mode

enum rpmi_cppc_fast_channel_cppc_mode

Enumerator

RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_PASSIVE	
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_ACTIVE	
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MAX_IDX	

30.176.2.5 rpmi_cppc_fast_channel_db_width

enum rpmi_cppc_fast_channel_db_width

Enumerator

RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_8	
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_16	
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_32	
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_64	

30.176.2.6 rpmi_cppc_service_id

enum rpmi_cppc_service_id
RPMI CPPC ServiceGroup Service IDs

Enumerator

RPMI_CPPC_SRV_ENABLE_NOTIFICATION	
RPMI_CPPC_SRV_PROBE_REG	
RPMI_CPPC_SRV_READ_REG	
RPMI_CPPC_SRV_WRITE_REG	
RPMI_CPPC_SRV_GET_FAST_CHANNEL_REGION	
RPMI_CPPC_SRV_GET_FAST_CHANNEL_OFFSET	
RPMI_CPPC_SRV_GET_HART_LIST	
RPMI_CPPC_SRV_MAX_COUNT	

30.176.2.7 rpmi_error

enum rpmi_error
RPMI Error Types

Enumerator

RPMI_SUCCESS	
RPMI_ERR_FAILED	
RPMI_ERR_NOTSUPP	
RPMI_ERR_INVALID_PARAM	
RPMI_ERR_DENIED	
RPMI_ERR_INVALID_ADDR	
RPMI_ERR_ALREADY	
RPMI_ERR_EXTENSION	
RPMI_ERR_HW_FAULT	

Enumerator

RPMI_ERR_BUSY	
RPMI_ERR_INVALID_STATE	
RPMI_ERR_BAD_RANGE	
RPMI_ERR_TIMEOUT	
RPMI_ERR_IO	
RPMI_ERR_NO_DATA	
RPMI_ERR_RESERVED_START	
RPMI_ERR_RESERVED_END	
RPMI_ERR_VENDOR_START	

30.176.2.8 rpmi_hsm_service_id

`enum rpmi_hsm_service_id`

RPMI HSM State Management ServiceGroup Service IDs

Enumerator

RPMI_HSM_SRV_ENABLE_NOTIFICATION	
RPMI_HSM_SRV_GET_HART_STATUS	
RPMI_HSM_SRV_GET_HART_LIST	
RPMI_HSM_SRV_GET_SUSPEND_TYPES	
RPMI_HSM_SRV_GET_SUSPEND_INFO	
RPMI_HSM_SRV_HART_START	
RPMI_HSM_SRV_HART_STOP	
RPMI_HSM_SRV_HART_SUSPEND	
RPMI_HSM_SRV_ID_MAX	

30.176.2.9 rpmi_message_type

`enum rpmi_message_type`

RPMI Messages Types

Enumerator

RPMI_MSG_NORMAL_REQUEST	
RPMI_MSG_POSTED_REQUEST	
RPMI_MSG_ACKNOWLDGEMENT	
RPMI_MSG_NOTIFICATION	

30.176.2.10 rpmi_servicegroup_id

`enum rpmi_servicegroup_id`

RPMI ServiceGroups IDs

Enumerator

RPMI_SRVGRP_ID_MIN	
RPMI_SRVGRP_BASE	
RPMI_SRVGRP_SYSTEM_RESET	
RPMI_SRVGRP_SYSTEM_SUSPEND	

Enumerator

RPMI_SRVGRP_HSM	
RPMI_SRVGRP_CPPC	
RPMI_SRVGRP_CLOCK	
RPMI_SRVGRP_ID_MAX_COUNT	
RPMI_SRVGRP_RESERVE_START	
RPMI_SRVGRP_RESERVE_END	
RPMI_SRVGRP_VENDOR_START	
RPMI_SRVGRP_VENDOR_END	

30.176.2.11 rpmi_sysrst_reset_type

`enum rpmi_sysrst_reset_type`
RPMI System Reset types

Enumerator

RPMI_SYSRST_TYPE_SHUTDOWN	
RPMI_SYSRST_TYPE_COLD_REBOOT	
RPMI_SYSRST_TYPE_WARM_REBOOT	
RPMI_SYSRST_TYPE_MAX	

30.176.2.12 rpmi_system_reset_service_id

`enum rpmi_system_reset_service_id`
RPMI System Reset ServiceGroup Service IDs

Enumerator

RPMI_SYSRST_SRV_ENABLE_NOTIFICATION	
RPMI_SYSRST_SRV_GET_ATTRIBUTES	
RPMI_SYSRST_SRV_SYSTEM_RESET	
RPMI_SYSRST_SRV_ID_MAX_COUNT	

30.176.2.13 rpmi_system_suspend_service_id

`enum rpmi_system_suspend_service_id`
RPMI System Suspend ServiceGroup Service IDs

Enumerator

RPMI_SYSSUSP_SRV_ENABLE_NOTIFICATION	
RPMI_SYSSUSP_SRV_GET_ATTRIBUTES	
RPMI_SYSSUSP_SRV_SYSTEM_SUSPEND	
RPMI_SYSSUSP_SRV_ID_MAX_COUNT	

30.176.3 Variable Documentation

30.176.3.1 __packed

`struct rpmi_message __packed`

30.177 rpmi_msgprot.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Rahul Pathak <rpathak@ventanamicro.com>
 *   Subrahmanya Lingappa <slingappa@ventanamicro.com>
 */

#ifndef __RPMI_MSGPROT_H__
#define __RPMI_MSGPROT_H__

#include <sbi/sbi_byteorder.h>
#include <sbi/sbi_error.h>

/*
 * 31                                                0
 * +---------------------+-----------------------+
 * | FLAGS | SERVICE_ID  |   SERVICEGROUP_ID     |
 * +---------------------+-----------------------+
 * |        TOKEN        |     DATA LENGTH       |
 * +---------------------+-----------------------+
 * |                 DATA/PAYLOAD                |
 * +---------------------------------------------+
 */

#define RPMI_MSG_HDR_OFFSET         (0x0)
#define RPMI_MSG_HDR_SIZE           (8)

#define RPMI_MSG_SERVICEGROUP_ID_OFFSET     (0x0)
#define RPMI_MSG_SERVICEGROUP_ID_SIZE       (2)

#define RPMI_MSG_SERVICE_ID_OFFSET      (0x2)
#define RPMI_MSG_SERVICE_ID_SIZE        (1)

#define RPMI_MSG_FLAGS_OFFSET           (0x3)
#define RPMI_MSG_FLAGS_SIZE         (1)

#define RPMI_MSG_FLAGS_TYPE_POS         (0U)
#define RPMI_MSG_FLAGS_TYPE_MASK        0x7
#define RPMI_MSG_FLAGS_TYPE         \
    ((0x7) << RPMI_MSG_FLAGS_TYPE_POS)

#define RPMI_MSG_FLAGS_DOORBELL_POS     (3U)
#define RPMI_MSG_FLAGS_DOORBELL_MASK        0x1
#define RPMI_MSG_FLAGS_DOORBELL         \
    ((0x1) << RPMI_MSG_FLAGS_DOORBELL_POS)

#define RPMI_MSG_DATALEN_OFFSET         (0x4)
#define RPMI_MSG_DATALEN_SIZE           (2)

#define RPMI_MSG_TOKEN_OFFSET           (0x6)
#define RPMI_MSG_TOKEN_SIZE         (2)
#define RPMI_MSG_TOKEN_MASK         (0xffffU)

#define RPMI_MSG_DATA_OFFSET            (RPMI_MSG_HDR_SIZE)
#define RPMI_MSG_DATA_SIZE(__slot_size)     ((__slot_size) - RPMI_MSG_HDR_SIZE)

#define RPMI_SLOT_SIZE_MIN          (64)

#define RPMI_NAME_CHARS_MAX         (16)

#define RPMI_QUEUE_HEAD_SLOT        0
#define RPMI_QUEUE_TAIL_SLOT        1
#define RPMI_QUEUE_HEADER_SLOTS     2

#define RPMI_DEF_TX_TIMEOUT         20
#define RPMI_DEF_RX_TIMEOUT         20

#define RPMI_VERSION(__major, __minor) (((__major) << 16) | (__minor))

struct rpmi_message_header {
    le16_t servicegroup_id;
    uint8_t service_id;
    uint8_t flags;
    le16_t datalen;
    le16_t token;
} __packed;

struct rpmi_message {
    struct rpmi_message_header header;
    u8 data[0];
```

```
} __packed;

enum rpmi_message_type {
    /* Normal request backed with ack */
    RPMI_MSG_NORMAL_REQUEST = 0x0,
    /* Request without any ack */
    RPMI_MSG_POSTED_REQUEST = 0x1,
    /* Acknowledgment for normal request message */
    RPMI_MSG_ACKNOWLDGEMENT = 0x2,
    /* Notification message */
    RPMI_MSG_NOTIFICATION = 0x3,
};

enum rpmi_error {
    /* Success */
    RPMI_SUCCESS        = 0,
    /* General failure  */
    RPMI_ERR_FAILED      = -1,
    /* Service or feature not supported */
    RPMI_ERR_NOTSUPP    = -2,
    /* Invalid Parameter  */
    RPMI_ERR_INVALID_PARAM    = -3,
    /*
     * Denied to insufficient permissions
     * or due to unmet prerequisite
     */
    RPMI_ERR_DENIED      = -4,
    /* Invalid address or offset */
    RPMI_ERR_INVALID_ADDR  = -5,
    /*
     * Operation failed as it was already in
     * progress or the state has changed already
     * for which the operation was carried out.
     */
    RPMI_ERR_ALREADY    = -6,
    /*
     * Error in implementation which violates
     * the specification version
     */
    RPMI_ERR_EXTENSION    = -7,
    /* Operation failed due to hardware issues */
    RPMI_ERR_HW_FAULT  = -8,
    /* System, device or resource is busy */
    RPMI_ERR_BUSY      = -9,
    /* System or device or resource in invalid state */
    RPMI_ERR_INVALID_STATE    = -10,
    /* Index, offset or address is out of range */
    RPMI_ERR_BAD_RANGE    = -11,
    /* Operation timed out */
    RPMI_ERR_TIMEOUT    = -12,
    /*
     * Error in input or output or
     * error in sending or receiving data
     * through communication medium
     */
    RPMI_ERR_IO      = -13,
    /* No data available */
    RPMI_ERR_NO_DATA    = -14,
    RPMI_ERR_RESERVED_START  = -15,
    RPMI_ERR_RESERVED_END  = -127,
    RPMI_ERR_VENDOR_START  = -128,
};

struct rpmi_message_args {
    u32 flags;
#define RPMI_MSG_FLAGS_NO_TX        (1U << 0)
#define RPMI_MSG_FLAGS_NO_RX        (1U << 1)
#define RPMI_MSG_FLAGS_NO_RX_TOKEN  (1U << 2)
    enum rpmi_message_type type;
    u8 service_id;
    u32 tx_endian_words;
    u32 rx_endian_words;
    u16 rx_token;
    u32 rx_data_len;
};

/*
 * RPMI SERVICEGROUPS AND SERVICES
 */

enum rpmi_servicegroup_id {
    RPMI_SRVGRP_ID_MIN = 0,
    RPMI_SRVGRP_BASE = 0x0001,
    RPMI_SRVGRP_SYSTEM_RESET = 0x0002,
    RPMI_SRVGRP_SYSTEM_SUSPEND = 0x0003,
    RPMI_SRVGRP_HSM = 0x0004,
    RPMI_SRVGRP_CPPC = 0x0005,
```

```
    RPMI_SRVGRP_CLOCK = 0x0007,
    RPMI_SRVGRP_ID_MAX_COUNT,

    /* Reserved range for service groups */
    RPMI_SRVGRP_RESERVE_START = RPMI_SRVGRP_ID_MAX_COUNT,
    RPMI_SRVGRP_RESERVE_END = 0x7FFF,

    /* Vendor/Implementation-specific service groups range */
    RPMI_SRVGRP_VENDOR_START = 0x8000,
    RPMI_SRVGRP_VENDOR_END = 0xFFFF,
};

struct rpmi_enable_notification_req {
    u32 eventid;
};

struct rpmi_enable_notification_resp {
    s32 status;
};

enum rpmi_base_service_id {
    RPMI_BASE_SRV_ENABLE_NOTIFICATION = 0x01,
    RPMI_BASE_SRV_GET_IMPLEMENTATION_VERSION = 0x02,
    RPMI_BASE_SRV_GET_IMPLEMENTATION_IDN = 0x03,
    RPMI_BASE_SRV_GET_SPEC_VERSION = 0x04,
    RPMI_BASE_SRV_GET_PLATFORM_INFO = 0x05,
    RPMI_BASE_SRV_PROBE_SERVICE_GROUP = 0x06,
    RPMI_BASE_SRV_GET_ATTRIBUTES = 0x07,
    RPMI_BASE_SRV_SET_MSI = 0x08,
};

#define RPMI_BASE_FLAGS_F0_PRIVILEGE        (1U << 2)
#define RPMI_BASE_FLAGS_F0_EV_NOTIFY        (1U << 1)
#define RPMI_BASE_FLAGS_F0_MSI_EN       (1U)

enum rpmi_base_context_priv_level {
    RPMI_BASE_CONTEXT_PRIV_S_MODE,
    RPMI_BASE_CONTEXT_PRIV_M_MODE,
};

struct rpmi_base_get_attributes_resp {
    s32 status_code;
    u32 f0;
    u32 f1;
    u32 f2;
    u32 f3;
};

struct rpmi_base_get_platform_info_resp {
    s32 status;
    u32 plat_info_len;
    char plat_info[];
};

enum rpmi_system_reset_service_id {
    RPMI_SYSRST_SRV_ENABLE_NOTIFICATION = 0x01,
    RPMI_SYSRST_SRV_GET_ATTRIBUTES = 0x02,
    RPMI_SYSRST_SRV_SYSTEM_RESET = 0x03,
    RPMI_SYSRST_SRV_ID_MAX_COUNT,
};

enum rpmi_sysrst_reset_type {
    RPMI_SYSRST_TYPE_SHUTDOWN = 0x0,
    RPMI_SYSRST_TYPE_COLD_REBOOT = 0x1,
    RPMI_SYSRST_TYPE_WARM_REBOOT = 0x2,
    RPMI_SYSRST_TYPE_MAX,
};

#define RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS       (1)
#define RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_MASK      \
            (1U << RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS)

struct rpmi_sysrst_get_reset_attributes_resp {
    s32 status;
    u32 flags;
};

enum rpmi_system_suspend_service_id {
    RPMI_SYSSUSP_SRV_ENABLE_NOTIFICATION = 0x01,
    RPMI_SYSSUSP_SRV_GET_ATTRIBUTES = 0x02,
    RPMI_SYSSUSP_SRV_SYSTEM_SUSPEND = 0x03,
    RPMI_SYSSUSP_SRV_ID_MAX_COUNT,
};

struct rpmi_syssusp_get_attr_req {
    u32 susp_type;
};
```

```

#define RPMI_SYSSUSP_ATTRS_FLAGS_RESUMEADDR (1U << 1)
#define RPMI_SYSSUSP_ATTRS_FLAGS_SUSPENDTYPE    1U

struct rpmi_syssusp_get_attr_resp {
    s32 status;
    u32 flags;
};

struct rpmi_syssusp_suspend_req {
    u32 hartid;
    u32 suspend_type;
    u32 resume_addr_lo;
    u32 resume_addr_hi;
};

struct rpmi_syssusp_suspend_resp {
    s32 status;
};

enum rpmi_hsm_service_id {
    RPMI_HSM_SRV_ENABLE_NOTIFICATION = 0x01,
    RPMI_HSM_SRV_GET_HART_STATUS = 0x02,
    RPMI_HSM_SRV_GET_HART_LIST = 0x03,
    RPMI_HSM_SRV_GET_SUSPEND_TYPES = 0x04,
    RPMI_HSM_SRV_GET_SUSPEND_INFO = 0x05,
    RPMI_HSM_SRV_HART_START = 0x06,
    RPMI_HSM_SRV_HART_STOP = 0x07,
    RPMI_HSM_SRV_HART_SUSPEND = 0x08,
    RPMI_HSM_SRV_ID_MAX = 0x09,
};

/* HSM service group request and response structs */
struct rpmi_hsm_hart_start_req {
    u32 hartid;
    u32 start_addr_lo;
    u32 start_addr_hi;
};

struct rpmi_hsm_hart_start_resp {
    s32 status;
};

struct rpmi_hsm_hart_stop_req {
    u32 hartid;
};

struct rpmi_hsm_hart_stop_resp {
    s32 status;
};

struct rpmi_hsm_hart_susp_req {
    u32 hartid;
    u32 suspend_type;
    u32 resume_addr_lo;
    u32 resume_addr_hi;
};

struct rpmi_hsm_hart_susp_resp {
    s32 status;
};

struct rpmi_hsm_get_hart_status_req {
    u32 hartid;
};

struct rpmi_hsm_get_hart_status_resp {
    s32 status;
    u32 hart_status;
};

struct rpmi_hsm_get_hart_list_req {
    u32 start_index;
};

struct rpmi_hsm_get_hart_list_resp {
    s32 status;
    u32 remaining;
    u32 returned;
    /* remaining space need to be adjusted for the above 3 u32's */
    u32 hartid[(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) - (sizeof(u32) * 3)) / sizeof(u32)];
};

struct rpmi_hsm_get_susp_types_req {
    u32 start_index;
};

```

```
struct rpmi_hsm_get_susp_types_resp {
    s32 status;
    u32 remaining;
    u32 returned;
    /* remaining space need to be adjusted for the above 3 u32's */
    u32 types[(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) - (sizeof(u32) * 3)) / sizeof(u32)];
};

struct rpmi_hsm_get_susp_info_req {
    u32 suspend_type;
};

#define RPMI_HSM_SUSPEND_INFO_FLAGS_TIMER_STOP      1U

struct rpmi_hsm_get_susp_info_resp {
    s32 status;
    u32 flags;
    u32 entry_latency_us;
    u32 exit_latency_us;
    u32 wakeup_latency_us;
    u32 min_residency_us;
};

enum rpmi_cppc_service_id {
    RPMI_CPPC_SRV_ENABLE_NOTIFICATION = 0x01,
    RPMI_CPPC_SRV_PROBE_REG = 0x02,
    RPMI_CPPC_SRV_READ_REG = 0x03,
    RPMI_CPPC_SRV_WRITE_REG = 0x04,
    RPMI_CPPC_SRV_GET_FAST_CHANNEL_REGION = 0x05,
    RPMI_CPPC_SRV_GET_FAST_CHANNEL_OFFSET = 0x06,
    RPMI_CPPC_SRV_GET_HART_LIST = 0x07,
    RPMI_CPPC_SRV_MAX_COUNT,
};

struct rpmi_cppc_probe_req {
    u32 hart_id;
    u32 reg_id;
};

struct rpmi_cppc_probe_resp {
    s32 status;
    u32 reg_len;
};

struct rpmi_cppc_read_reg_req {
    u32 hart_id;
    u32 reg_id;
};

struct rpmi_cppc_read_reg_resp {
    s32 status;
    u32 data_lo;
    u32 data_hi;
};

struct rpmi_cppc_write_reg_req {
    u32 hart_id;
    u32 reg_id;
    u32 data_lo;
    u32 data_hi;
};

struct rpmi_cppc_write_reg_resp {
    s32 status;
};

struct rpmi_cppc_get_fastchan_offset_req {
    u32 hart_id;
};

struct rpmi_cppc_get_fastchan_offset_resp {
    s32 status;
    u32 fc_perf_request_offset_lo;
    u32 fc_perf_request_offset_hi;
    u32 fc_perf_feedback_offset_lo;
    u32 fc_perf_feedback_offset_hi;
};

#define RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS        3
#define RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MASK       \
            (3U << RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS)
#define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS   1
#define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_MASK  \
            (3U << RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS)
#define RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_SUPPORTED   (1U << 0)

struct rpmi_cppc_get_fastchan_region_resp {
```

```
    s32 status;
    u32 flags;
    u32 region_addr_lo;
    u32 region_addr_hi;
    u32 region_size_lo;
    u32 region_size_hi;
    u32 db_addr_lo;
    u32 db_addr_hi;
    u32 db_setmask_lo;
    u32 db_setmask_hi;
    u32 db_preservemask_lo;
    u32 db_preservemask_hi;
};

enum rpmi_cppc_fast_channel_db_width {
    RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_8 = 0x0,
    RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_16 = 0x1,
    RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_32 = 0x2,
    RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_64 = 0x3,
};

enum rpmi_cppc_fast_channel_cppc_mode {
    RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_PASSIVE = 0x0,
    RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_ACTIVE = 0x1,
    RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MAX_IDX,
};

struct rpmi_cppc_hart_list_req {
    u32 start_index;
};

struct rpmi_cppc_hart_list_resp {
    s32 status;
    u32 remaining;
    u32 returned;
    /* remaining space need to be adjusted for the above 3 u32's */
    u32 hartid[(RPMI_MSG_DATA_SIZE(RPMI_SLOT_SIZE_MIN) - (sizeof(u32) * 3)) / sizeof(u32)];
};

enum rpmi_clock_service_id {
    RPMI_CLOCK_SRV_ENABLE_NOTIFICATION = 0x01,
    RPMI_CLOCK_SRV_GET_NUM_CLOCKS = 0x02,
    RPMI_CLOCK_SRV_GET_ATTRIBUTES = 0x03,
    RPMI_CLOCK_SRV_GET_SUPPORTED_RATES = 0x04,
    RPMI_CLOCK_SRV_SET_CONFIG = 0x05,
    RPMI_CLOCK_SRV_GET_CONFIG = 0x06,
    RPMI_CLOCK_SRV_SET_RATE = 0x07,
    RPMI_CLOCK_SRV_GET_RATE = 0x08,
    RPMI_CLOCK_SRV_MAX_COUNT,
};

struct rpmi_clock_get_num_clocks_resp {
    s32 status;
    u32 num_clocks;
};

struct rpmi_clock_get_attributes_req {
    u32 clock_id;
};

struct rpmi_clock_get_attributes_resp {
    s32 status;
#define RPMI_CLOCK_FLAGS_FORMAT_POS     30
#define RPMI_CLOCK_FLAGS_FORMAT_MASK        \
            (3U « RPMI_CLOCK_FLAGS_CLOCK_FORMAT_POS)
#define RPMI_CLOCK_FLAGS_FORMAT_DISCRETE    0
#define RPMI_CLOCK_FLAGS_FORMAT_LINEAR      1
    u32 flags;
    u32 num_rates;
    u32 transition_latency;
    u8 name[16];
};

struct rpmi_clock_get_supported_rates_req {
    u32 clock_id;
    u32 clock_rate_index;
};

struct rpmi_clock_get_supported_rates_resp {
    s32 status;
    u32 flags;
    u32 remaining;
    u32 returned;
    u32 clock_rate[0];
};

struct rpmi_clock_set_config_req {
```

```
    u32 clock_id;
#define RPMI_CLOCK_CONFIG_ENABLE        (1U « 0)
    u32 config;
};

struct rpmi_clock_set_config_resp {
    s32 status;
};

struct rpmi_clock_get_config_req {
    u32 clock_id;
};

struct rpmi_clock_get_config_resp {
    s32 status;
    u32 config;
};

struct rpmi_clock_set_rate_req {
    u32 clock_id;
#define RPMI_CLOCK_SET_RATE_FLAGS_MASK      (3U « 0)
#define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_DOWN    0
#define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_UP  1
#define RPMI_CLOCK_SET_RATE_FLAGS_ROUND_PLAT    2
    u32 flags;
    u32 clock_rate_low;
    u32 clock_rate_high;
};

struct rpmi_clock_set_rate_resp {
    s32 status;
};

struct rpmi_clock_get_rate_req {
    u32 clock_id;
};

struct rpmi_clock_get_rate_resp {
    s32 status;
    u32 clock_rate_low;
    u32 clock_rate_high;
};

#endif /* !__RPMI_MSGPROT_H__ */
```

30.178 include/sbi_utils/mpxy/fdt_mpxy.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>
```

Include dependency graph for fdt_mpxy.h:

This graph shows which files directly or indirectly include this file:

Functions

- static void fdt_mpxy_init (const void *fdt)

30.178.1 Function Documentation

30.178.1.1 fdt_mpxy_init()

```
static void fdt_mpxy_init (
            const void * fdt )  [inline], [static]
```

30.179 fdt_mpxy.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2024 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __FDT_MPXY_H__
#define __FDT_MPXY_H__

#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>

#ifdef CONFIG_FDT_MPXY

void fdt_mpxy_init(const void *fdt);

#else

static inline void fdt_mpxy_init(const void *fdt) { }

#endif

#endif
```

30.180 include/sbi_utils/regmap/fdt_regmap.h File Reference

```
#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/regmap/regmap.h>
```

Include dependency graph for fdt_regmap.h:

This graph shows which files directly or indirectly include this file:

Functions

- int fdt_regmap_get_by_phandle (const void ∗fdt, u32 phandle, struct regmap ∗∗out_rmap)
- int fdt_regmap_get (const void ∗fdt, int nodeoff, struct regmap ∗∗out_rmap)

30.180.1 Function Documentation

30.180.1.1 fdt_regmap_get()

```
int fdt_regmap_get (
            const void * fdt,
            int nodeoff,
            struct regmap ** out_rmap )
```

Get regmap instance based on "regmap" property of the specified DT node Here is the call graph for this function:

Here is the caller graph for this function:

30.180.1.2 fdt_regmap_get_by_phandle()

```
int fdt_regmap_get_by_phandle (
            const void * fdt,
            u32 phandle,
            struct regmap ** out_rmap )
```

Get regmap instance based on phandle Here is the call graph for this function:

Here is the caller graph for this function:

30.181 fdt_regmap.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __FDT_REGMAP_H__
#define __FDT_REGMAP_H__

#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/regmap/regmap.h>

struct fdt_phandle_args;

int fdt_regmap_get_by_phandle(const void *fdt, u32 phandle,
                  struct regmap **out_rmap);

int fdt_regmap_get(const void *fdt, int nodeoff, struct regmap **out_rmap);

#endif
```

30.182 include/sbi_utils/regmap/regmap.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for regmap.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct regmap

Functions

- static struct regmap ∗ to_regmap (struct sbi_dlist ∗node)
- struct regmap ∗ regmap_find (unsigned int id)
- int regmap_add (struct regmap ∗rmap)
- void regmap_remove (struct regmap ∗rmap)
- int regmap_read (struct regmap ∗rmap, unsigned int reg, unsigned int ∗val)
- int regmap_write (struct regmap ∗rmap, unsigned int reg, unsigned int val)
- int regmap_update_bits (struct regmap ∗rmap, unsigned int reg, unsigned int mask, unsigned int val)

30.182.1 Function Documentation

30.182.1.1 regmap_add()

```
int regmap_add (
            struct regmap * rmap )
```

Register a regmap instance Here is the call graph for this function:

Here is the caller graph for this function:

30.182.1.2 regmap_find()

```
struct regmap * regmap_find (
            unsigned int id )
```

Find a registered regmap instance Here is the call graph for this function:

Here is the caller graph for this function:

30.182.1.3 regmap_read()

```
int regmap_read (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )
```

Read a register in a regmap instance Here is the call graph for this function:

30.182.1.4 regmap_remove()

```
void regmap_remove (
            struct regmap * rmap )
```

Un-register a regmap instance Here is the call graph for this function:

30.182.1.5 regmap_update_bits()

```
int regmap_update_bits (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int mask,
            unsigned int val )
```

Read-modify-write a register in a regmap instance Here is the call graph for this function:

Here is the caller graph for this function:

30.182.1.6 regmap_write()

```
int regmap_write (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )
```

Write a register in a regmap instance Here is the call graph for this function:

30.182.1.7 to_regmap()

```
static struct regmap * to_regmap (
            struct sbi_dlist * node )  [inline], [static]
```

Here is the caller graph for this function:

30.183 regmap.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2023 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __REGMAP_H__
```

```
#define __REGMAP_H__

#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>

struct regmap {
    unsigned int id;

    int reg_shift;
    int reg_stride;
    unsigned int reg_base;
    unsigned int reg_max;

    int (*reg_read)(struct regmap *rmap, unsigned int reg,
            unsigned int *val);

    int (*reg_write)(struct regmap *rmap, unsigned int reg,
             unsigned int val);

    int (*reg_update_bits)(struct regmap *rmap, unsigned int reg,
                    unsigned int mask, unsigned int val);

    struct sbi_dlist node;
};

static inline struct regmap *to_regmap(struct sbi_dlist *node)
{
    return container_of(node, struct regmap, node);
}

struct regmap *regmap_find(unsigned int id);

int regmap_add(struct regmap *rmap);

void regmap_remove(struct regmap *rmap);

int regmap_read(struct regmap *rmap, unsigned int reg, unsigned int *val);

int regmap_write(struct regmap *rmap, unsigned int reg, unsigned int val);

int regmap_update_bits(struct regmap *rmap, unsigned int reg,
               unsigned int mask, unsigned int val);

#endif
```

30.184 include/sbi_utils/reset/fdt_reset.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>
```

Include dependency graph for fdt_reset.h:

This graph shows which files directly or indirectly include this file:

Functions

- static void fdt_reset_init (const void *fdt)

30.184.1 Function Documentation

30.184.1.1 fdt_reset_init()

```
static void fdt_reset_init (
            const void * fdt )  [inline], [static]
```

30.185 fdt_reset.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __FDT_RESET_H__
#define __FDT_RESET_H__

#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>

#ifdef CONFIG_FDT_RESET

void fdt_reset_init(const void *fdt);

#else

static inline void fdt_reset_init(const void *fdt) { }

#endif

#endif
```

30.186 include/sbi_utils/serial/cadence-uart.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for cadence-uart.h:

This graph shows which files directly or indirectly include this file:

Functions

- int cadence_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

30.186.1 Function Documentation

30.186.1.1 cadence_uart_init()

```
int cadence_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.187 cadence-uart.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2022 StarFive Technology Co., Ltd.
00005  *
00006  * Author: Jun Liang Tan <junliang.tan@linux.starfivetech.com>
00007  */
00008
00009 #ifndef __SERIAL_CADENCE_UART_H__
00010 #define __SERIAL_CADENCE_UART_H__
00011
00012 #include <sbi/sbi_types.h>
00013
00014 int cadence_uart_init(unsigned long base, u32 in_freq, u32 baudrate);
00015
00016 #endif
```

30.188 include/sbi_utils/serial/fdt_serial.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>
```

Include dependency graph for fdt_serial.h:

This graph shows which files directly or indirectly include this file:

Functions

- static int fdt_serial_init (const void *fdt)

30.188.1 Function Documentation

30.188.1.1 fdt_serial_init()

```
static int fdt_serial_init (
            const void * fdt )  [inline], [static]
```

30.189 fdt_serial.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __FDT_SERIAL_H__
#define __FDT_SERIAL_H__

#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>

#ifdef CONFIG_FDT_SERIAL

int fdt_serial_init(const void *fdt);
```

```
#else

static inline int fdt_serial_init(const void *fdt) { return 0; }

#endif

#endif
```

30.190 include/sbi_utils/serial/gaisler-uart.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for gaisler-uart.h:

This graph shows which files directly or indirectly include this file:

Functions

- int gaisler_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

30.190.1 Function Documentation

30.190.1.1 gaisler_uart_init()

```
int gaisler_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.191 gaisler-uart.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Cobham Gaisler AB.
 *
 * Authors:
 *   Daniel Cederman <cederman@gaisler.com>
 */

#ifndef __SERIAL_GAISLER_APBUART_H__
#define __SERIAL_GAISLER_APBUART_H__

#include <sbi/sbi_types.h>

int gaisler_uart_init(unsigned long base, u32 in_freq, u32 baudrate);

#endif
```

30.192 include/sbi_utils/serial/litex-uart.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for litex-uart.h:

This graph shows which files directly or indirectly include this file:

Functions

- int litex_uart_init (unsigned long base)

30.192.1 Function Documentation

30.192.1.1 litex_uart_init()

```
int litex_uart_init (
            unsigned long base )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.193 litex-uart.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Gabriel Somlo
 *
 * Authors:
 *   Gabriel Somlo <gsomlo@gmail.com>
 */

#ifndef __SERIAL_LITEX_UART_H__
#define __SERIAL_LITEX_UART_H__

#include <sbi/sbi_types.h>

int litex_uart_init(unsigned long base);

#endif
```

30.194 include/sbi_utils/serial/renesas-scif.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- int renesas_scif_init (unsigned long base, u32 in_freq, u32 baudrate)

30.194.1 Function Documentation

30.194.1.1 renesas_scif_init()

```
int renesas_scif_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.195 renesas-scif.h

Go to the documentation of this file.

```
/* SPDX-License-Identifier: BSD-2-Clause */
/*
 * Copyright (C) 2022 Renesas Electronics Corporation
 */

#ifndef __SERIAL_RENESAS_SCIF_H__
#define __SERIAL_RENESAS_SCIF_H__

int renesas_scif_init(unsigned long base, u32 in_freq, u32 baudrate);

#endif /* __SERIAL_RENESAS_SCIF_H__ */
```

30.196 include/sbi_utils/serial/semihosting.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for semihosting.h:

This graph shows which files directly or indirectly include this file:

Enumerations

- enum semihosting_open_mode {
  MODE_READ = 0x0 , MODE_BINARY = 0x1 , MODE_PLUS = 0x2 , MODE_WRITE = 0x4 ,
  MODE_APPEND = 0x8 }

Functions

- static int semihosting_init (void)

- static bool semihosting_enabled (void)

30.196.1 Enumeration Type Documentation

30.196.1.1 semihosting_open_mode

enum semihosting_open_mode

enum semihosting_open_mode - Numeric file modes for use with semihosting_open() MODE_READ: 'r' MODE_↩ BINARY: 'b' MODE_PLUS: '+' MODE_WRITE: 'w' MODE_APPEND: 'a'

These modes represent the mode string used by fopen(3) in a form which can be passed to semihosting_open(). These do NOT correspond directly to O_RDONLY, O_CREAT, etc; see fopen(3) for details. In particular, @↩ MODE_PLUS effectively results in adding O_RDWR, and @MODE_WRITE will add O_TRUNC. For compatibility, @MODE_BINARY should be added when opening non-text files (such as images).

Enumerator

MODE_READ	
MODE_BINARY	
MODE_PLUS	
MODE_WRITE	
MODE_APPEND	

30.196.2 Function Documentation

30.196.2.1 semihosting_enabled()

```
static bool semihosting_enabled (
            void ) [inline], [static]
```

30.196.2.2 semihosting_init()

```
static int semihosting_init (
            void ) [inline], [static]
```

30.197 semihosting.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2022 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 *   Kautuk Consul <kconsul@ventanamicro.com>
 */

#ifndef __SERIAL_SEMIHOSTING_H__
#define __SERIAL_SEMIHOSTING_H__

#include <sbi/sbi_types.h>

enum semihosting_open_mode {
    MODE_READ   = 0x0,
    MODE_BINARY  = 0x1,
    MODE_PLUS   = 0x2,
    MODE_WRITE    = 0x4,
    MODE_APPEND  = 0x8,
};

#ifdef CONFIG_SERIAL_SEMIHOSTING
int semihosting_init(void);
bool semihosting_enabled(void);
#else
static inline int semihosting_init(void) { return SBI_ENODEV; }
static inline bool semihosting_enabled(void) { return false; }
#endif
```

```
00046
00047 #endif
```

30.198 include/sbi_utils/serial/shakti-uart.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for shakti-uart.h:

This graph shows which files directly or indirectly include this file:

Functions

- int shakti_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

30.198.1 Function Documentation

30.198.1.1 shakti_uart_init()

```
int shakti_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.199 shakti-uart.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Vijai Kumar K <vijai@behindbytes.com>
 */

#ifndef __SERIAL_SHAKTI_UART_H__
#define __SERIAL_SHAKTI_UART_H__

#include <sbi/sbi_types.h>

int shakti_uart_init(unsigned long base, u32 in_freq, u32 baudrate);

#endif
```

30.200 include/sbi_utils/serial/sifive-uart.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for sifive-uart.h:

This graph shows which files directly or indirectly include this file:

Functions

- int sifive_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

30.200.1 Function Documentation

30.200.1.1 sifive_uart_init()

```
int sifive_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.201 sifive-uart.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2019 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __SERIAL_SIFIVE_UART_H__
#define __SERIAL_SIFIVE_UART_H__

#include <sbi/sbi_types.h>

int sifive_uart_init(unsigned long base, u32 in_freq, u32 baudrate);

#endif
```

30.202 include/sbi_utils/serial/uart8250.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for uart8250.h:

This graph shows which files directly or indirectly include this file:

Functions

- int uart8250_init (unsigned long base, u32 in_freq, u32 baudrate, u32 reg_shift, u32 reg_width, u32 reg_↩ offset)

30.202.1 Function Documentation

30.202.1.1 uart8250_init()

```
int uart8250_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate,
            u32 reg_shift,
            u32 reg_width,
            u32 reg_offset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.203 uart8250.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
00003  *
00004  * Copyright (c) 2019 Western Digital Corporation or its affiliates.
00005  *
00006  * Authors:
00007  *   Anup Patel <anup.patel@wdc.com>
00008  */
00009
00010 #ifndef __SERIAL_UART8250_H__
00011 #define __SERIAL_UART8250_H__
00012
00013 #include <sbi/sbi_types.h>
00014
00015 int uart8250_init(unsigned long base, u32 in_freq, u32 baudrate, u32 reg_shift,
00016           u32 reg_width, u32 reg_offset);
00017
00018 #endif
```

30.204 include/sbi_utils/serial/xlnx_uartlite.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for xlnx_uartlite.h:

This graph shows which files directly or indirectly include this file:

Functions

- int xlnx_uartlite_init (unsigned long base)

30.204.1 Function Documentation

30.204.1.1 xlnx_uartlite_init()

```
int xlnx_uartlite_init (
            unsigned long base )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.205 xlnx_uartlite.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2022 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Alistair Francis <alistair.francis@wdc.com>
 */
#ifndef __SERIAL_XLNX_UARTLITE_H__
#define __SERIAL_XLNX_UARTLITE_H__

#include <sbi/sbi_types.h>

int xlnx_uartlite_init(unsigned long base);

#endif
```

30.206 include/sbi_utils/suspend/fdt_suspend.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>
```

Include dependency graph for fdt_suspend.h:

This graph shows which files directly or indirectly include this file:

Functions

- static void fdt_suspend_init (const void *fdt)

30.206.1 Function Documentation

30.206.1.1 fdt_suspend_init()

```
static void fdt_suspend_init (
            const void * fdt )  [inline], [static]
```

30.207 fdt_suspend.h

Go to the documentation of this file.

```
00001 /*
00002  * SPDX-License-Identifier: BSD-2-Clause
```

```
 *
 * Copyright (c) 2024 Ventana Micro Systems Inc.
 *
 * Authors:
 *   Anup Patel <apatel@ventanamicro.com>
 */

#ifndef __FDT_SUSPEND_H__
#define __FDT_SUSPEND_H__

#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>

#ifdef CONFIG_FDT_SUSPEND

void fdt_suspend_init(const void *fdt);

#else

static inline void fdt_suspend_init(const void *fdt) { }

#endif

#endif
```

30.208 include/sbi_utils/sys/atcsmu.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for atcsmu.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct smu_data

Macros

- #define PCS0_WE_OFFSET 0x90
- #define PCSm_WE_OFFSET(i) ((i + 3) ∗ 0x20 + PCS0_WE_OFFSET)
- #define PCS0_CTL_OFFSET 0x94
- #define PCSm_CTL_OFFSET(i) ((i + 3) ∗ 0x20 + PCS0_CTL_OFFSET)
- #define PCS_CTL_CMD_SHIFT 0
- #define PCS_CTL_PARAM_SHIFT 3
- #define SLEEP_CMD 0x3
- #define WAKEUP_CMD (0x0 | (1 << PCS_CTL_PARAM_SHIFT))
- #define LIGHTSLEEP_MODE 0
- #define DEEPSLEEP_MODE 1
- #define LIGHT_SLEEP_CMD (SLEEP_CMD | (LIGHTSLEEP_MODE << PCS_CTL_PARAM_SHIFT))
- #define DEEP_SLEEP_CMD (SLEEP_CMD | (DEEPSLEEP_MODE << PCS_CTL_PARAM_SHIFT))
- #define PCS0_CFG_OFFSET 0x80
- #define PCSm_CFG_OFFSET(i) ((i + 3) ∗ 0x20 + PCS0_CFG_OFFSET)
- #define PCS_CFG_LIGHT_SLEEP_SHIFT 2
- #define PCS_CFG_LIGHT_SLEEP (1 << PCS_CFG_LIGHT_SLEEP_SHIFT)
- #define PCS_CFG_DEEP_SLEEP_SHIFT 3
- #define PCS_CFG_DEEP_SLEEP (1 << PCS_CFG_DEEP_SLEEP_SHIFT)
- #define RESET_VEC_LO_OFFSET 0x50
- #define RESET_VEC_HI_OFFSET 0x60
- #define RESET_VEC_8CORE_OFFSET 0x1a0
- #define HARTn_RESET_VEC_LO(n)
- #define HARTn_RESET_VEC_HI(n)
- #define PCS_MAX_NR 8
- #define FLASH_BASE 0x80000000ULL

Functions

- int smu_set_wakeup_events (struct smu_data ∗smu, u32 events, u32 hartid)
- bool smu_support_sleep_mode (struct smu_data ∗smu, u32 sleep_mode, u32 hartid)
- int smu_set_command (struct smu_data ∗smu, u32 pcs_ctl, u32 hartid)
- int smu_set_reset_vector (struct smu_data ∗smu, ulong wakeup_addr, u32 hartid)

30.208.1 Macro Definition Documentation

30.208.1.1 DEEP_SLEEP_CMD

```
#define DEEP_SLEEP_CMD (SLEEP_CMD | (DEEPSLEEP_MODE << PCS_CTL_PARAM_SHIFT))
```

30.208.1.2 DEEPSLEEP_MODE

```
#define DEEPSLEEP_MODE 1
```

30.208.1.3 FLASH_BASE

```
#define FLASH_BASE 0x80000000ULL
```

30.208.1.4 HARTn_RESET_VEC_HI

```
#define HARTn_RESET_VEC_HI(
            n )
```

Value:

```
(RESET_VEC_HI_OFFSET + \
((n) < 4 ? 0 : RESET_VEC_8CORE_OFFSET) + \
((n) * 0x4))
```

30.208.1.5 HARTn_RESET_VEC_LO

```
#define HARTn_RESET_VEC_LO(
            n )
```

Value:

```
(RESET_VEC_LO_OFFSET + \
((n) < 4 ? 0 : RESET_VEC_8CORE_OFFSET) + \
((n) * 0x4))
```

30.208.1.6 LIGHT_SLEEP_CMD

```
#define LIGHT_SLEEP_CMD (SLEEP_CMD | (LIGHTSLEEP_MODE << PCS_CTL_PARAM_SHIFT))
```

30.208.1.7 LIGHTSLEEP_MODE

```
#define LIGHTSLEEP_MODE 0
```

30.208.1.8 PCS0_CFG_OFFSET

```
#define PCS0_CFG_OFFSET 0x80
```

30.208.1.9 PCS0_CTL_OFFSET

```
#define PCS0_CTL_OFFSET 0x94
```

30.208.1.10 PCS0_WE_OFFSET

```
#define PCS0_WE_OFFSET 0x90
```

30.208.1.11 PCS_CFG_DEEP_SLEEP

```
#define PCS_CFG_DEEP_SLEEP (1 << PCS_CFG_DEEP_SLEEP_SHIFT)
```

30.208.1.12 PCS_CFG_DEEP_SLEEP_SHIFT

```
#define PCS_CFG_DEEP_SLEEP_SHIFT 3
```

30.208.1.13 PCS_CFG_LIGHT_SLEEP

```
#define PCS_CFG_LIGHT_SLEEP (1 << PCS_CFG_LIGHT_SLEEP_SHIFT)
```

30.208.1.14 PCS_CFG_LIGHT_SLEEP_SHIFT

```
#define PCS_CFG_LIGHT_SLEEP_SHIFT 2
```

30.208.1.15 PCS_CTL_CMD_SHIFT

```
#define PCS_CTL_CMD_SHIFT 0
```

30.208.1.16 PCS_CTL_PARAM_SHIFT

```
#define PCS_CTL_PARAM_SHIFT 3
```

30.208.1.17 PCS_MAX_NR

```
#define PCS_MAX_NR 8
```

30.208.1.18 PCSm_CFG_OFFSET

```
#define PCSm_CFG_OFFSET(
            i ) ((i + 3) * 0x20 + PCS0_CFG_OFFSET)
```

30.208.1.19 PCSm_CTL_OFFSET

```
#define PCSm_CTL_OFFSET(
            i ) ((i + 3) * 0x20 + PCS0_CTL_OFFSET)
```

30.208.1.20 PCSm_WE_OFFSET

```
#define PCSm_WE_OFFSET(
            i ) ((i + 3) * 0x20 + PCS0_WE_OFFSET)
```

30.208.1.21 RESET_VEC_8CORE_OFFSET

```
#define RESET_VEC_8CORE_OFFSET 0x1a0
```

30.208.1.22 RESET_VEC_HI_OFFSET

```
#define RESET_VEC_HI_OFFSET 0x60
```

30.208.1.23 RESET_VEC_LO_OFFSET

```
#define RESET_VEC_LO_OFFSET 0x50
```

30.208.1.24 SLEEP_CMD

```
#define SLEEP_CMD 0x3
```

30.208.1.25 WAKEUP_CMD

```
#define WAKEUP_CMD (0x0 | (1 << PCS_CTL_PARAM_SHIFT))
```

30.208.2 Function Documentation

30.208.2.1 smu_set_command()

```
int smu_set_command (
            struct smu_data * smu,
            u32 pcs_ctl,
            u32 hartid )  [inline]
```

30.208.2.2 smu_set_reset_vector()

```
int smu_set_reset_vector (
            struct smu_data * smu,
            ulong wakeup_addr,
            u32 hartid )  [inline]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.208.2.3 smu_set_wakeup_events()

```
int smu_set_wakeup_events (
            struct smu_data * smu,
            u32 events,
            u32 hartid )  [inline]
```

30.208.2.4 smu_support_sleep_mode()

```
bool smu_support_sleep_mode (
            struct smu_data * smu,
            u32 sleep_mode,
            u32 hartid )  [inline]
```

Here is the call graph for this function:

30.209 atcsmu.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-3-Clause
 *
 * Copyright (c) 2023 Andes Technology Corporation
 */

#ifndef _SYS_ATCSMU_H
#define _SYS_ATCSMU_H

#include <sbi/sbi_types.h>

/* clang-format off */

#define PCS0_WE_OFFSET      0x90
#define PCSm_WE_OFFSET(i)   ((i + 3) * 0x20 + PCS0_WE_OFFSET)

#define PCS0_CTL_OFFSET     0x94
#define PCSm_CTL_OFFSET(i)  ((i + 3) * 0x20 + PCS0_CTL_OFFSET)
#define PCS_CTL_CMD_SHIFT   0
#define PCS_CTL_PARAM_SHIFT 3
#define SLEEP_CMD        0x3
#define WAKEUP_CMD       (0x0 | (1 « PCS_CTL_PARAM_SHIFT))
#define LIGHTSLEEP_MODE     0
#define DEEPSLEEP_MODE      1
#define LIGHT_SLEEP_CMD     (SLEEP_CMD | (LIGHTSLEEP_MODE « PCS_CTL_PARAM_SHIFT))
#define DEEP_SLEEP_CMD      (SLEEP_CMD | (DEEPSLEEP_MODE « PCS_CTL_PARAM_SHIFT))

#define PCS0_CFG_OFFSET         0x80
```

```
#define PCSm_CFG_OFFSET(i)       ((i + 3) * 0x20 + PCS0_CFG_OFFSET)
#define PCS_CFG_LIGHT_SLEEP_SHIFT   2
#define PCS_CFG_LIGHT_SLEEP      (1 « PCS_CFG_LIGHT_SLEEP_SHIFT)
#define PCS_CFG_DEEP_SLEEP_SHIFT    3
#define PCS_CFG_DEEP_SLEEP       (1 « PCS_CFG_DEEP_SLEEP_SHIFT)

#define RESET_VEC_LO_OFFSET 0x50
#define RESET_VEC_HI_OFFSET 0x60
#define RESET_VEC_8CORE_OFFSET  0x1a0
#define HARTn_RESET_VEC_LO(n)    (RESET_VEC_LO_OFFSET + \
                    ((n) < 4 ? 0 : RESET_VEC_8CORE_OFFSET) + \
                    ((n) * 0x4))
#define HARTn_RESET_VEC_HI(n)    (RESET_VEC_HI_OFFSET + \
                    ((n) < 4 ? 0 : RESET_VEC_8CORE_OFFSET) + \
                    ((n) * 0x4))

#define PCS_MAX_NR  8
#define FLASH_BASE  0x80000000ULL

/* clang-format on */

struct smu_data {
    unsigned long addr;
};

int smu_set_wakeup_events(struct smu_data *smu, u32 events, u32 hartid);
bool smu_support_sleep_mode(struct smu_data *smu, u32 sleep_mode, u32 hartid);
int smu_set_command(struct smu_data *smu, u32 pcs_ctl, u32 hartid);
int smu_set_reset_vector(struct smu_data *smu, ulong wakeup_addr, u32 hartid);

#endif /* _SYS_ATCSMU_H */
```

30.210 include/sbi_utils/sys/htif.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for htif.h:

This graph shows which files directly or indirectly include this file:

Functions

- int htif_serial_init (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom_tohost↩ _addr)
- int htif_system_reset_init (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom↩ _tohost_addr)

30.210.1 Function Documentation

30.210.1.1 htif_serial_init()

```
int htif_serial_init (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.210.1.2 htif_system_reset_init()

```
int htif_system_reset_init (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.211 htif.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-3-Clause
 *
 * Copyright (c) 2010-2020, The Regents of the University of California
 * (Regents).  All Rights Reserved.
 */

#ifndef __SYS_HTIF_H__
#define __SYS_HTIF_H__

#include <sbi/sbi_types.h>

int htif_serial_init(bool custom_addr,
             unsigned long custom_fromhost_addr,
             unsigned long custom_tohost_addr);

int htif_system_reset_init(bool custom_addr,
               unsigned long custom_fromhost_addr,
               unsigned long custom_tohost_addr);

#endif
```

30.212 include/sbi_utils/timer/aclint_mtimer.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for aclint_mtimer.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct aclint_mtimer_data

Macros

- #define ACLINT_MTIMER_ALIGN 0x8
- #define ACLINT_MTIMER_MAX_HARTS 4095
- #define ACLINT_DEFAULT_MTIME_OFFSET 0x7ff8
- #define ACLINT_DEFAULT_MTIME_SIZE 0x8
- #define ACLINT_DEFAULT_MTIMECMP_OFFSET 0x0000
- #define ACLINT_DEFAULT_MTIMECMP_SIZE 0x7ff8
- #define CLINT_MTIMER_OFFSET 0x4000
- #define MTIMER_REGION_ALIGN 0x1000

Functions

- void aclint_mtimer_sync (struct aclint_mtimer_data *mt)
- void aclint_mtimer_set_reference (struct aclint_mtimer_data *mt, struct aclint_mtimer_data *ref)
- int aclint_mtimer_cold_init (struct aclint_mtimer_data *mt, struct aclint_mtimer_data *reference)

30.212.1 Macro Definition Documentation

30.212.1.1 ACLINT_DEFAULT_MTIME_OFFSET

```
#define ACLINT_DEFAULT_MTIME_OFFSET 0x7ff8
```

30.212.1.2 ACLINT_DEFAULT_MTIME_SIZE

```
#define ACLINT_DEFAULT_MTIME_SIZE 0x8
```

30.212.1.3 ACLINT_DEFAULT_MTIMECMP_OFFSET

```
#define ACLINT_DEFAULT_MTIMECMP_OFFSET 0x0000
```

30.212.1.4 ACLINT_DEFAULT_MTIMECMP_SIZE

```
#define ACLINT_DEFAULT_MTIMECMP_SIZE 0x7ff8
```

30.212.1.5 ACLINT_MTIMER_ALIGN

```
#define ACLINT_MTIMER_ALIGN 0x8
```

30.212.1.6 ACLINT_MTIMER_MAX_HARTS

```
#define ACLINT_MTIMER_MAX_HARTS 4095
```

30.212.1.7 CLINT_MTIMER_OFFSET

```
#define CLINT_MTIMER_OFFSET 0x4000
```

30.212.1.8 MTIMER_REGION_ALIGN

```
#define MTIMER_REGION_ALIGN 0x1000
```

30.212.2 Function Documentation

30.212.2.1 aclint_mtimer_cold_init()

```
int aclint_mtimer_cold_init (
            struct aclint_mtimer_data * mt,
            struct aclint_mtimer_data * reference )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.212.2.2 aclint_mtimer_set_reference()

```
void aclint_mtimer_set_reference (
            struct aclint_mtimer_data * mt,
            struct aclint_mtimer_data * ref )
```

Here is the caller graph for this function:

30.212.2.3 aclint_mtimer_sync()

```
void aclint_mtimer_sync (
            struct aclint_mtimer_data * mt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.213 aclint_mtimer.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __TIMER_ACLINT_MTIMER_H__
#define __TIMER_ACLINT_MTIMER_H__

#include <sbi/sbi_types.h>

#define ACLINT_MTIMER_ALIGN     0x8
#define ACLINT_MTIMER_MAX_HARTS     4095

#define ACLINT_DEFAULT_MTIME_OFFSET 0x7ff8
#define ACLINT_DEFAULT_MTIME_SIZE   0x8
#define ACLINT_DEFAULT_MTIMECMP_OFFSET  0x0000
#define ACLINT_DEFAULT_MTIMECMP_SIZE    0x7ff8

#define CLINT_MTIMER_OFFSET     0x4000

#define MTIMER_REGION_ALIGN     0x1000

struct aclint_mtimer_data {
    /* Public details */
```

```
    unsigned long mtime_freq;
    unsigned long mtime_addr;
    unsigned long mtime_size;
    unsigned long mtimecmp_addr;
    unsigned long mtimecmp_size;
    u32 first_hartid;
    u32 hart_count;
    bool has_64bit_mmio;
    bool has_shared_mtime;
    /* Private details (initialized and used by ACLINT MTIMER library) */
    struct aclint_mtimer_data *time_delta_reference;
    unsigned long time_delta_computed;
    u64 (*time_rd)(volatile u64 *addr);
    void (*time_wr)(bool timecmp, u64 value, volatile u64 *addr);
};

void aclint_mtimer_sync(struct aclint_mtimer_data *mt);

void aclint_mtimer_set_reference(struct aclint_mtimer_data *mt,
                 struct aclint_mtimer_data *ref);

int aclint_mtimer_cold_init(struct aclint_mtimer_data *mt,
                struct aclint_mtimer_data *reference);

#endif
```

30.214 include/sbi_utils/timer/andes_plmt.h File Reference

This graph shows which files directly or indirectly include this file:

Data Structures

- struct plmt_data

Macros

- #define DEFAULT_AE350_PLMT_FREQ 60000000
- #define PLMT_REGION_ALIGN 0x1000

Functions

- int plmt_cold_timer_init (struct plmt_data *plmt)

30.214.1 Macro Definition Documentation

30.214.1.1 DEFAULT_AE350_PLMT_FREQ

```
#define DEFAULT_AE350_PLMT_FREQ 60000000
```

30.214.1.2 PLMT_REGION_ALIGN

```
#define PLMT_REGION_ALIGN 0x1000
```

30.214.2 Function Documentation

30.214.2.1 plmt_cold_timer_init()

```
int plmt_cold_timer_init (
            struct plmt_data * plmt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.215 andes_plmt.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2022 Andes Technology Corporation
 *
 * Authors:
 *   Zong Li <zong@andestech.com>
 *   Nylon Chen <nylon7@andestech.com>
 *   Yu Chien Peter Lin <peterlin@andestech.com>
 */

#ifndef __TIMER_ANDES_PLMT_H__
#define __TIMER_ANDES_PLMT_H__

#define DEFAULT_AE350_PLMT_FREQ 60000000
#define PLMT_REGION_ALIGN 0x1000

struct plmt_data {
    u32 hart_count;
    unsigned long size;
    unsigned long timer_freq;
    volatile u64 *time_val;
    volatile u64 *time_cmp;
};

int plmt_cold_timer_init(struct plmt_data *plmt);

#endif /* __TIMER_ANDES_PLMT_H__ */
```

30.216 include/sbi_utils/timer/fdt_timer.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>
```

Include dependency graph for fdt_timer.h:

This graph shows which files directly or indirectly include this file:

Functions

- static int fdt_timer_init (void)

30.216.1 Function Documentation

30.216.1.1 fdt_timer_init()

```
static int fdt_timer_init (
            void ) [inline], [static]
```

30.217 fdt_timer.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2020 Western Digital Corporation or its affiliates.
 *
 * Authors:
 *   Anup Patel <anup.patel@wdc.com>
 */

#ifndef __FDT_TIMER_H__
#define __FDT_TIMER_H__

#include <sbi/sbi_types.h>
#include <sbi_utils/fdt/fdt_driver.h>

#ifdef CONFIG_FDT_TIMER

int fdt_timer_init(void);

#else

static inline int fdt_timer_init(void) { return 0; }

#endif

#endif
```

30.218 lib/sbi/riscv_asm.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for riscv_asm.c:

Macros

- #define switchcase_csr_read(__csr_num, __val)
- #define switchcase_csr_read_2(__csr_num, __val)

- #define switchcase_csr_read_4(__csr_num, __val)
- #define switchcase_csr_read_8(__csr_num, __val)
- #define switchcase_csr_read_16(__csr_num, __val)
- #define switchcase_csr_read_32(__csr_num, __val)
- #define switchcase_csr_read_64(__csr_num, __val)
- #define switchcase_csr_write(__csr_num, __val)
- #define switchcase_csr_write_2(__csr_num, __val)
- #define switchcase_csr_write_4(__csr_num, __val)
- #define switchcase_csr_write_8(__csr_num, __val)
- #define switchcase_csr_write_16(__csr_num, __val)
- #define switchcase_csr_write_32(__csr_num, __val)
- #define switchcase_csr_write_64(__csr_num, __val)

Functions

- int misa_extension_imp (char ext)
- int misa_xlen (void)
- void misa_string (int xlen, char *out, unsigned int out_sz)
- unsigned long csr_read_num (int csr_num)
- void csr_write_num (int csr_num, unsigned long val)
- static unsigned long ctz (unsigned long x)
- int pmp_disable (unsigned int n)
- int is_pmp_entry_mapped (unsigned long entry)
- int pmp_set (unsigned int n, unsigned long prot, unsigned long addr, unsigned long log2len)
- int pmp_get (unsigned int n, unsigned long *prot_out, unsigned long *addr_out, unsigned long *log2len)

30.218.1 Macro Definition Documentation

30.218.1.1 switchcase_csr_read

```
#define switchcase_csr_read(
            __csr_num,
            __val )
```

Value:

```
case __csr_num:                 \
    __val = csr_read(__csr_num);        \
    break;
```

30.218.1.2 switchcase_csr_read_16

```
#define switchcase_csr_read_16(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_read_8(__csr_num + 0, __val)     \
switchcase_csr_read_8(__csr_num + 8, __val)
```

30.218.1.3 switchcase_csr_read_2

```
#define switchcase_csr_read_2(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_read(__csr_num + 0, __val)     \
switchcase_csr_read(__csr_num + 1, __val)
```

30.218.1.4 switchcase_csr_read_32

```
#define switchcase_csr_read_32(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_read_16(__csr_num + 0, __val)  \
switchcase_csr_read_16(__csr_num + 16, __val)
```

30.218.1.5 switchcase_csr_read_4

```
#define switchcase_csr_read_4(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_read_2(__csr_num + 0, __val)    \
switchcase_csr_read_2(__csr_num + 2, __val)
```

30.218.1.6 switchcase_csr_read_64

```
#define switchcase_csr_read_64(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_read_32(__csr_num + 0, __val)  \
switchcase_csr_read_32(__csr_num + 32, __val)
```

30.218.1.7 switchcase_csr_read_8

```
#define switchcase_csr_read_8(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_read_4(__csr_num + 0, __val)    \
switchcase_csr_read_4(__csr_num + 4, __val)
```

30.218.1.8 switchcase_csr_write

```
#define switchcase_csr_write(
            __csr_num,
            __val )
```

Value:

```
case __csr_num:                 \
    csr_write(__csr_num, __val);        \
    break;
```

30.218.1.9 switchcase_csr_write_16

```
#define switchcase_csr_write_16(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_write_8(__csr_num + 0, __val)  \
switchcase_csr_write_8(__csr_num + 8, __val)
```

30.218.1.10 switchcase_csr_write_2

```
#define switchcase_csr_write_2(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_write(__csr_num + 0, __val)  \
switchcase_csr_write(__csr_num + 1, __val)
```

30.218.1.11 switchcase_csr_write_32

```
#define switchcase_csr_write_32(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_write_16(__csr_num + 0, __val)     \
switchcase_csr_write_16(__csr_num + 16, __val)
```

30.218.1.12 switchcase_csr_write_4

```
#define switchcase_csr_write_4(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_write_2(__csr_num + 0, __val)  \
switchcase_csr_write_2(__csr_num + 2, __val)
```

30.218.1.13 switchcase_csr_write_64

```
#define switchcase_csr_write_64(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_write_32(__csr_num + 0, __val)     \
switchcase_csr_write_32(__csr_num + 32, __val)
```

30.218.1.14 switchcase_csr_write_8

```
#define switchcase_csr_write_8(
            __csr_num,
            __val )
```

Value:

```
switchcase_csr_write_4(__csr_num + 0, __val)  \
switchcase_csr_write_4(__csr_num + 4, __val)
```

30.218.2 Function Documentation

30.218.2.1 csr_read_num()

```
unsigned long csr_read_num (
            int csr_num )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.218.2.2 csr_write_num()

```
void csr_write_num (
            int csr_num,
            unsigned long val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.218.2.3 ctz()

```
static unsigned long ctz (
            unsigned long x )  [static]
```

Here is the caller graph for this function:

30.218.2.4 is_pmp_entry_mapped()

```
int is_pmp_entry_mapped (
            unsigned long entry )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.218.2.5 misa_extension_imp()

```
int misa_extension_imp (
            char ext )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.218.2.6 misa_string()

```
void misa_string (
            int xlen,
            char * out,
            unsigned int out_sz )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.218.2.7 misa_xlen()

```
int misa_xlen (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.218.2.8 pmp_disable()

```
int pmp_disable (
            unsigned int n )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.218.2.9 pmp_get()

```
int pmp_get (
            unsigned int n,
            unsigned long * prot_out,
            unsigned long * addr_out,
            unsigned long * log2len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.218.2.10 pmp_set()

```
int pmp_set (
            unsigned int n,
            unsigned long prot,
            unsigned long addr,
            unsigned long log2len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.219 lib/sbi/riscv_atomic.c File Reference

```
#include <sbi/sbi_bitops.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_barrier.h>
```

Include dependency graph for riscv_atomic.c:

Macros

- #define __axchg(ptr, new, size)
- #define axchg(ptr, x)

Functions

- long atomic_read (atomic_t *atom)
- void atomic_write (atomic_t *atom, long value)
- long atomic_add_return (atomic_t *atom, long value)
- long atomic_sub_return (atomic_t *atom, long value)
- long atomic_cmpxchg (atomic_t *atom, long oldval, long newval)
- long atomic_xchg (atomic_t *atom, long newval)
- unsigned int atomic_raw_xchg_uint (volatile unsigned int *ptr, unsigned int newval)
- unsigned long atomic_raw_xchg_ulong (volatile unsigned long *ptr, unsigned long newval)
- int atomic_raw_set_bit (int nr, volatile unsigned long *addr)
- int atomic_raw_clear_bit (int nr, volatile unsigned long *addr)
- int atomic_set_bit (int nr, atomic_t *atom)
- int atomic_clear_bit (int nr, atomic_t *atom)

30.219.1 Macro Definition Documentation

30.219.1.1 __axchg

```
#define __axchg(
            ptr,
            new,
            size )
```

Value:

```
({                                          \
    __typeof__(ptr) __ptr = (ptr);                  \
    __typeof__(new) __new = (new);                  \
    __typeof__(*(ptr)) __ret;                   \
    switch (size) {                         \
    case 4:                             \
        __asm__ __volatile__ (                  \
```

```
            "   amoswap.w.aqrl %0, %2, %1\n"        \
            : "=r" (__ret), "+A" (*__ptr)           \
            : "r" (__new)                           \
            : "memory");                            \
        break;                                      \
    case 8:                                         \
        __asm__ __volatile__ (                      \
            "   amoswap.d.aqrl %0, %2, %1\n"        \
            : "=r" (__ret), "+A" (*__ptr)           \
            : "r" (__new)                           \
            : "memory");                            \
        break;                                      \
    default:                                        \
        break;                                      \
    }                                               \
    __ret;                                          \
})
```

30.219.1.2 axchg

```
#define axchg(
            ptr,
            x )
```

Value:

```
({                                                  \
    __typeof__(*(ptr)) _x_ = (x);                   \
    (__typeof__(*(ptr))) __axchg((ptr), _x_, sizeof(*(ptr))); \
})
```

30.219.2 Function Documentation

30.219.2.1 atomic_add_return()

```
long atomic_add_return (
            atomic_t * atom,
            long value )
```

Here is the caller graph for this function:

30.219.2.2 atomic_clear_bit()

```
int atomic_clear_bit (
            int nr,
```

```
            atomic_t * atom )
```

Clear a bit in an atomic variable and return the value of bit before modify. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:

Here is the caller graph for this function:

30.219.2.3 atomic_cmpxchg()

```
long atomic_cmpxchg (
            atomic_t * atom,
            long oldval,
            long newval )
```

Here is the caller graph for this function:

30.219.2.4 atomic_raw_clear_bit()

```
int atomic_raw_clear_bit (
            int nr,
            volatile unsigned long * addr )
```

Clear a bit in any address and return the value of bit before modify. @nr : Bit to set. @addr: Address to modify

Here is the caller graph for this function:

30.219.2.5 atomic_raw_set_bit()

```
int atomic_raw_set_bit (
            int nr,
            volatile unsigned long * addr )
```

Set a bit in any address and return the value of bit before modify. @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:

30.219.2.6 atomic_raw_xchg_uint()

```
unsigned int atomic_raw_xchg_uint (
            volatile unsigned int * ptr,
            unsigned int newval )
```

30.219.2.7 atomic_raw_xchg_ulong()

```
unsigned long atomic_raw_xchg_ulong (
            volatile unsigned long * ptr,
            unsigned long newval )
```

Here is the caller graph for this function:

30.219.2.8 atomic_read()

```
long atomic_read (
            atomic_t * atom )
```

Here is the caller graph for this function:

30.219.2.9 atomic_set_bit()

```
int atomic_set_bit (
            int nr,
            atomic_t * atom )
```

Set a bit in an atomic variable and return the value of bit before modify. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:

Here is the caller graph for this function:

30.219.2.10 atomic_sub_return()

```
long atomic_sub_return (
            atomic_t * atom,
            long value )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.219.2.11 atomic_write()

```
void atomic_write (
            atomic_t * atom,
            long value )
```

Here is the caller graph for this function:

30.219.2.12 atomic_xchg()

```
long atomic_xchg (
            atomic_t * atom,
            long newval )
```

Here is the caller graph for this function:

30.220 lib/sbi/riscv_locks.c File Reference

```
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_locks.h>
```

Include dependency graph for riscv_locks.c:

Functions

- static bool spin_lock_unlocked (spinlock_t lock)
- bool spin_lock_check (spinlock_t *lock)
- bool spin_trylock (spinlock_t *lock)
- void spin_lock (spinlock_t *lock)
- void spin_unlock (spinlock_t *lock)

30.220.1 Function Documentation

30.220.1.1 spin_lock()

```
void spin_lock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

30.220.1.2 spin_lock_check()

```
bool spin_lock_check (
            spinlock_t * lock )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.220.1.3 spin_lock_unlocked()

```
static bool spin_lock_unlocked (
            spinlock_t lock )  [inline], [static]
```

Here is the caller graph for this function:

30.220.1.4 spin_trylock()

```
bool spin_trylock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

30.220.1.5 spin_unlock()

```
void spin_unlock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

30.221 lib/sbi/sbi_bitmap.c File Reference

```
#include <sbi/sbi_bitmap.h>
```

Include dependency graph for sbi_bitmap.c:

Functions

- void __bitmap_and (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- void __bitmap_or (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- void __bitmap_xor (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)

30.221.1 Function Documentation

30.221.1.1 __bitmap_and()

```
void __bitmap_and (
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

Here is the caller graph for this function:

30.221.1.2 __bitmap_or()

```
void __bitmap_or (
```

```
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

Here is the caller graph for this function:

30.221.1.3 __bitmap_xor()

```
void __bitmap_xor (
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

Here is the caller graph for this function:

30.222 lib/sbi/sbi_bitops.c File Reference

```
#include <sbi/sbi_bitops.h>
```

Include dependency graph for sbi_bitops.c:

Macros

- #define BITOP_WORD(nr) ((nr) / BITS_PER_LONG)

Functions

- unsigned long find_first_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_first_zero_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_last_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_next_bit (const unsigned long *addr, unsigned long size, unsigned long offset)
- unsigned long find_next_zero_bit (const unsigned long *addr, unsigned long size, unsigned long offset)

30.222.1 Macro Definition Documentation

30.222.1.1 BITOP_WORD

```
#define BITOP_WORD(
            nr ) ((nr) / BITS_PER_LONG)
```

30.222.2 Function Documentation

30.222.2.1 find_first_bit()

```
unsigned long find_first_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_first_bit - find the first set bit in a memory region @addr: The address to start the search at @size: The maximum size to search

Returns the bit number of the first set bit. Here is the call graph for this function:

30.222.2.2 find_first_zero_bit()

```
unsigned long find_first_zero_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_first_zero_bit - find the first cleared bit in a memory region @addr: The address to start the search at @size: The maximum size to search

Returns the bit number of the first cleared bit.

30.222.2.3 find_last_bit()

```
unsigned long find_last_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_last_bit - find the last set bit in a memory region @addr: The address to start the search at @size: The maximum size to search

Returns the bit number of the first set bit, or size. Here is the call graph for this function:

30.222.2.4 find_next_bit()

```
unsigned long find_next_bit (
            const unsigned long * addr,
            unsigned long size,
            unsigned long offset )
```

find_next_bit - find the next set bit in a memory region @addr: The address to base the search on @offset: The bitnumber to start searching at @size: The bitmap size in bits Here is the call graph for this function:

30.222.2.5 find_next_zero_bit()

```
unsigned long find_next_zero_bit (
            const unsigned long * addr,
            unsigned long size,
            unsigned long offset )
```

find_next_zero_bit - find the next cleared bit in a memory region @addr: The address to base the search on @offset: The bitnumber to start searching at @size: The bitmap size in bits

30.223 lib/sbi/sbi_console.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_fifo.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_console.c:

Macros

- #define CONSOLE_TBUF_MAX 256
- #define CONSOLE_EARLY_BUFFER_SIZE 256
- #define PAD_RIGHT 1
- #define PAD_ZERO 2
- #define PAD_ALTERNATE 4
- #define PAD_SIGN 8
- #define USE_TBUF 16
- #define PRINT_BUF_LEN 64
- #define va_start(v, l) __builtin_va_start((v), l)
- #define va_end __builtin_va_end
- #define va_arg __builtin_va_arg

Typedefs

- typedef __builtin_va_list va_list

Functions

- static SBI_FIFO_DEFINE (console_early_fifo, console_early_buffer, CONSOLE_EARLY_BUFFER_SIZE, sizeof(char))
- bool sbi_isprintable (char c)
- int sbi_getc (void)
- static unsigned long nputs (const char *str, unsigned long len)
- static void nputs_all (const char *str, unsigned long len)
- void sbi_putc (char ch)
- void sbi_puts (const char *str)
- unsigned long sbi_nputs (const char *str, unsigned long len)
- void sbi_gets (char *s, int maxwidth, char endchar)
- unsigned long sbi_ngets (char *str, unsigned long len)
- static void printc (char **out, u32 *out_len, char ch, int flags)
- static int prints (char **out, u32 *out_len, const char *string, int width, int flags)
- static int printi (char **out, u32 *out_len, long long i, int width, int flags, int type)
- static int print (char **out, u32 *out_len, const char *format, va_list args)
- int sbi_sprintf (char *out, const char *format,...)
- int sbi_snprintf (char *out, u32 out_sz, const char *format,...)

- int sbi_printf (const char *format,...)
- int sbi_dprintf (const char *format,...)
- void sbi_panic (const char *format,...)
- const struct sbi_console_device * sbi_console_get_device (void)
- void sbi_console_set_device (const struct sbi_console_device *dev)

Variables

- static const struct sbi_console_device * console_dev = NULL
- static char console_tbuf [CONSOLE_TBUF_MAX]
- static u32 console_tbuf_len
- static spinlock_t console_out_lock = SPIN_LOCK_INITIALIZER
- static char console_early_buffer [CONSOLE_EARLY_BUFFER_SIZE] = { 0 }

30.223.1 Macro Definition Documentation

30.223.1.1 CONSOLE_EARLY_BUFFER_SIZE

```
#define CONSOLE_EARLY_BUFFER_SIZE 256
```

30.223.1.2 CONSOLE_TBUF_MAX

```
#define CONSOLE_TBUF_MAX 256
```

30.223.1.3 PAD_ALTERNATE

```
#define PAD_ALTERNATE 4
```

30.223.1.4 PAD_RIGHT

```
#define PAD_RIGHT 1
```

30.223.1.5 PAD_SIGN

```
#define PAD_SIGN 8
```

30.223.1.6 PAD_ZERO

```
#define PAD_ZERO 2
```

30.223.1.7 PRINT_BUF_LEN

```
#define PRINT_BUF_LEN 64
```

30.223.1.8 USE_TBUF

```
#define USE_TBUF 16
```

30.223.1.9 va_arg

```
#define va_arg __builtin_va_arg
```

30.223.1.10 va_end

```
#define va_end __builtin_va_end
```

30.223.1.11 va_start

```
#define va_start(
            v,
            l ) __builtin_va_start((v), l)
```

30.223.2 Typedef Documentation

30.223.2.1 va_list

```
typedef __builtin_va_list va_list
```

30.223.3 Function Documentation

30.223.3.1 nputs()

```
static unsigned long nputs (
            const char * str,
            unsigned long len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.2 nputs_all()

```
static void nputs_all (
            const char * str,
            unsigned long len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.3 print()

```
static int print (
            char ** out,
            u32 * out_len,
            const char * format,
            va_list args )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.4 printc()

```
static void printc (
            char ** out,
            u32 * out_len,
            char ch,
            int flags )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.5 printi()

```
static int printi (
            char ** out,
            u32 * out_len,
            long long i,
            int width,
            int flags,
```

```
int type )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.6 prints()

```
static int prints (
            char ** out,
            u32 * out_len,
            const char * string,
            int width,
            int flags )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.7 sbi_console_get_device()

```
const struct sbi_console_device * sbi_console_get_device (
            void  )
```

Here is the caller graph for this function:

30.223.3.8 sbi_console_set_device()

```
void sbi_console_set_device (
            const struct sbi_console_device * dev )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.9 sbi_dprintf()

```
int sbi_dprintf (
            const char * format,
             ...  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.10 SBI_FIFO_DEFINE()

```
static SBI_FIFO_DEFINE (
            console_early_fifo ,
            console_early_buffer ,
            CONSOLE_EARLY_BUFFER_SIZE ,
            sizeof(char)  )  [static]
```

30.223.3.11 sbi_getc()

```
int sbi_getc (
            void  )
```

Here is the caller graph for this function:

30.223.3.12 sbi_gets()

```
void sbi_gets (
            char * s,
            int maxwidth,
            char endchar )
```

Here is the call graph for this function:

30.223.3.13 sbi_isprintable()

```
bool sbi_isprintable (
            char c )
```

30.223.3.14 sbi_ngets()

```
unsigned long sbi_ngets (
            char * str,
            unsigned long len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.15 sbi_nputs()

```
unsigned long sbi_nputs (
            const char * str,
            unsigned long len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.16 sbi_panic()

```
void sbi_panic (
            const char * format,
             ...  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.17 sbi_printf()

```
int sbi_printf (
            const char * format,
             ...  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.18 sbi_putc()

```
void sbi_putc (
            char ch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.19 sbi_puts()

```
void sbi_puts (
            const char * str )
```

Here is the call graph for this function:

30.223.3.20 sbi_snprintf()

```
int sbi_snprintf (
            char * out,
            u32 out_sz,
            const char * format,
             ...  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.223.3.21 sbi_sprintf()

```
int sbi_sprintf (
            char * out,
            const char * format,
             ...  )
```

Here is the call graph for this function:

30.223.4 Variable Documentation

30.223.4.1 console_dev

```
const struct sbi_console_device* console_dev = NULL  [static]
```

30.223.4.2 console_early_buffer

```
char console_early_buffer[CONSOLE_EARLY_BUFFER_SIZE] = { 0 }  [static]
```

30.223.4.3 console_out_lock

```
spinlock_t console_out_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.223.4.4 console_tbuf

```
char console_tbuf[CONSOLE_TBUF_MAX]  [static]
```

30.223.4.5 console_tbuf_len

```
u32 console_tbuf_len  [static]
```

30.224 lib/sbi/sbi_cppc.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_cppc.h>
```

Include dependency graph for sbi_cppc.c:

Functions

- const struct sbi_cppc_device * sbi_cppc_get_device (void)
- void sbi_cppc_set_device (const struct sbi_cppc_device *dev)
- static bool sbi_cppc_is_reserved (unsigned long reg)
- static bool sbi_cppc_readable (unsigned long reg)
- static bool sbi_cppc_writable (unsigned long reg)
- int sbi_cppc_probe (unsigned long reg)
- int sbi_cppc_read (unsigned long reg, uint64_t *val)
- int sbi_cppc_write (unsigned long reg, uint64_t val)

Variables

- static const struct sbi_cppc_device * cppc_dev = NULL

30.224.1 Function Documentation

30.224.1.1 sbi_cppc_get_device()

```
const struct sbi_cppc_device * sbi_cppc_get_device (
            void  )
```

Here is the caller graph for this function:

30.224.1.2 sbi_cppc_is_reserved()

```
static bool sbi_cppc_is_reserved (
            unsigned long reg )  [static]
```

Here is the caller graph for this function:

30.224.1.3 sbi_cppc_probe()

```
int sbi_cppc_probe (
            unsigned long reg )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.224.1.4 sbi_cppc_read()

```
int sbi_cppc_read (
            unsigned long reg,
            uint64_t * val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.224.1.5 sbi_cppc_readable()

```
static bool sbi_cppc_readable (
            unsigned long reg )  [static]
```

Here is the caller graph for this function:

30.224.1.6 sbi_cppc_set_device()

```
void sbi_cppc_set_device (
            const struct sbi_cppc_device * dev )
```

Here is the caller graph for this function:

30.224.1.7 sbi_cppc_writable()

```
static bool sbi_cppc_writable (
            unsigned long reg )  [static]
```

Here is the caller graph for this function:

30.224.1.8 sbi_cppc_write()

```
int sbi_cppc_write (
            unsigned long reg,
            uint64_t val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.224.2 Variable Documentation

30.224.2.1 cppc_dev

```
const struct sbi_cppc_device* cppc_dev = NULL  [static]
```

30.225 lib/sbi/sbi_dbtr.c File Reference

```
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_csr_detect.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_byteorder.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
```

```
#include <sbi/sbi_trap.h>
#include <sbi/sbi_dbtr.h>
#include <sbi/sbi_heap.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_dbtr.c:

Macros

- #define dbtr_get_hart_state_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, hart_state_ptr_offset)
- #define dbtr_thishart_state_ptr() dbtr_get_hart_state_ptr(sbi_scratch_thishart_ptr())
- #define dbtr_set_hart_state_ptr(__scratch, __hart_state)
- #define INDEX_TO_TRIGGER(_index)
- #define for_each_trig_entry(_base, _max, _etype, _entry)
- #define DBTR_SHMEM_MAKE_PHYS(_p_hi, _p_lo) (_p_lo)

Functions

- static bool sbi_dbtr_shmem_disabled (struct sbi_dbtr_hart_triggers_state *hs)
- static void sbi_dbtr_disable_shmem (struct sbi_dbtr_hart_triggers_state *hs)
- static void * hart_shmem_base (struct sbi_dbtr_hart_triggers_state *hs)
- static void sbi_trigger_init (struct sbi_dbtr_trigger *trig, unsigned long type_mask, unsigned long idx)
- static struct sbi_dbtr_trigger * sbi_alloc_trigger (void)
- static void sbi_free_trigger (struct sbi_dbtr_trigger *trig)
- int sbi_dbtr_init (struct sbi_scratch *scratch, bool coldboot)
- int sbi_dbtr_get_total_triggers (void)
- int sbi_dbtr_setup_shmem (const struct sbi_domain *dom, unsigned long smode, unsigned long shmem_↩ phys_lo, unsigned long shmem_phys_hi)
- static void dbtr_trigger_setup (struct sbi_dbtr_trigger *trig, struct sbi_dbtr_data_msg *recv)
- static void update_bit (unsigned long new, int nr, volatile unsigned long *addr)
- static void dbtr_trigger_enable (struct sbi_dbtr_trigger *trig)
- static void dbtr_trigger_disable (struct sbi_dbtr_trigger *trig)
- static void dbtr_trigger_clear (struct sbi_dbtr_trigger *trig)
- static int dbtr_trigger_supported (unsigned long type)
- static int dbtr_trigger_valid (unsigned long type, unsigned long tdata)
- int sbi_dbtr_num_trig (unsigned long data, unsigned long *out)
- int sbi_dbtr_read_trig (unsigned long smode, unsigned long trig_idx_base, unsigned long trig_count)
- int sbi_dbtr_install_trig (unsigned long smode, unsigned long trig_count, unsigned long *out)
- int sbi_dbtr_uninstall_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_enable_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_update_trig (unsigned long smode, unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_disable_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)

Variables

- static unsigned long hart_state_ptr_offset

30.225.1 Macro Definition Documentation

30.225.1.1 dbtr_get_hart_state_ptr

```
#define dbtr_get_hart_state_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, hart_state_ptr_offset)
```

30.225.1.2 dbtr_set_hart_state_ptr

```
#define dbtr_set_hart_state_ptr(
            __scratch,
            __hart_state )
```

Value:

```
sbi_scratch_write_type((__scratch), void *, hart_state_ptr_offset, \
                (__hart_state))
```

30.225.1.3 DBTR_SHMEM_MAKE_PHYS

```
#define DBTR_SHMEM_MAKE_PHYS(
            _p_hi,
            _p_lo ) (_p_lo)
```

30.225.1.4 dbtr_thishart_state_ptr

```
#define dbtr_thishart_state_ptr( )  dbtr_get_hart_state_ptr(sbi_scratch_thishart_ptr())
```

30.225.1.5 for_each_trig_entry

```
#define for_each_trig_entry(
            _base,
            _max,
            _etype,
            _entry )
```

Value:

```
for (int _idx = 0; _entry = ((_etype *)_base + _idx),       \
     _idx < _max;                                           \
     _idx++, _entry = ((_etype *)_base + _idx))
```

30.225.1.6 INDEX_TO_TRIGGER

```
#define INDEX_TO_TRIGGER(
            _index )
```

Value:

```
({                                          \
    struct sbi_dbtr_trigger *__trg = NULL;          \
    struct sbi_dbtr_hart_triggers_state *__hs = NULL;   \
    __hs = dbtr_get_hart_state_ptr(sbi_scratch_thishart_ptr()); \
    __trg = &__hs->triggers[_index];            \
    (__trg);                                    \
})
```

30.225.2 Function Documentation

30.225.2.1 dbtr_trigger_clear()

```
static void dbtr_trigger_clear (
            struct sbi_dbtr_trigger * trig )  [static]
```

Here is the caller graph for this function:

30.225.2.2 dbtr_trigger_disable()

```
static void dbtr_trigger_disable (
            struct sbi_dbtr_trigger * trig )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.3 dbtr_trigger_enable()

```
static void dbtr_trigger_enable (
            struct sbi_dbtr_trigger * trig )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.4 dbtr_trigger_setup()

```
static void dbtr_trigger_setup (
            struct sbi_dbtr_trigger * trig,
            struct sbi_dbtr_data_msg * recv )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.5 dbtr_trigger_supported()

```
static int dbtr_trigger_supported (
            unsigned long type )  [static]
```

Here is the caller graph for this function:

30.225.2.6 dbtr_trigger_valid()

```
static int dbtr_trigger_valid (
```

```
            unsigned long type,
            unsigned long tdata )  [static]
```

Here is the caller graph for this function:

30.225.2.7 hart_shmem_base()

```
static void * hart_shmem_base (
            struct sbi_dbtr_hart_triggers_state * hs )  [inline], [static]
```

Here is the caller graph for this function:

30.225.2.8 sbi_alloc_trigger()

```
static struct sbi_dbtr_trigger * sbi_alloc_trigger (
            void  )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.9 sbi_dbtr_disable_shmem()

```
static void sbi_dbtr_disable_shmem (
            struct sbi_dbtr_hart_triggers_state * hs )  [inline], [static]
```

Here is the caller graph for this function:

30.225.2.10 sbi_dbtr_disable_trig()

```
int sbi_dbtr_disable_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.11 sbi_dbtr_enable_trig()

```
int sbi_dbtr_enable_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.12 sbi_dbtr_get_total_triggers()

```
int sbi_dbtr_get_total_triggers (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.13 sbi_dbtr_init()

```
int sbi_dbtr_init (
            struct sbi_scratch * scratch,
            bool coldboot )
```

Initialize debug triggers Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.14 sbi_dbtr_install_trig()

```
int sbi_dbtr_install_trig (
            unsigned long smode,
            unsigned long trig_count,
            unsigned long * out )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.15 sbi_dbtr_num_trig()

```
int sbi_dbtr_num_trig (
            unsigned long data,
            unsigned long * out )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.16 sbi_dbtr_read_trig()

```
int sbi_dbtr_read_trig (
            unsigned long smode,
            unsigned long trig_idx_base,
            unsigned long trig_count )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.17 sbi_dbtr_setup_shmem()

```
int sbi_dbtr_setup_shmem (
            const struct sbi_domain * dom,
            unsigned long smode,
            unsigned long shmem_phys_lo,
            unsigned long shmem_phys_hi )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.18 sbi_dbtr_shmem_disabled()

```
static bool sbi_dbtr_shmem_disabled (
            struct sbi_dbtr_hart_triggers_state * hs )  [inline], [static]
```

Here is the caller graph for this function:

30.225.2.19 sbi_dbtr_uninstall_trig()

```
int sbi_dbtr_uninstall_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.20 sbi_dbtr_update_trig()

```
int sbi_dbtr_update_trig (
            unsigned long smode,
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.2.21 sbi_free_trigger()

```
static void sbi_free_trigger (
            struct sbi_dbtr_trigger * trig )  [inline], [static]
```

Here is the caller graph for this function:

30.225.2.22 sbi_trigger_init()

```
static void sbi_trigger_init (
            struct sbi_dbtr_trigger * trig,
            unsigned long type_mask,
            unsigned long idx )  [static]
```

Here is the caller graph for this function:

30.225.2.23 update_bit()

```
static void update_bit (
            unsigned long new,
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.225.3 Variable Documentation

30.225.3.1 hart_state_ptr_offset

```
unsigned long hart_state_ptr_offset [static]
```

Offset of pointer to HART's debug triggers info in scratch space

30.226 lib/sbi/sbi_domain.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_list.h>
#include <sbi/sbi_math.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_domain.c:

Macros

- #define ROOT_REGION_MAX 32

Functions

- SBI_LIST_HEAD (domain_list)
- struct sbi_domain ∗ sbi_hartindex_to_domain (u32 hartindex)
- void sbi_update_hartindex_to_domain (u32 hartindex, struct sbi_domain ∗dom)
- bool sbi_domain_is_assigned_hart (const struct sbi_domain ∗dom, u32 hartindex)
- int sbi_domain_get_assigned_hartmask (const struct sbi_domain ∗dom, struct sbi_hartmask ∗mask)

- void sbi_domain_memregion_init (unsigned long addr, unsigned long size, unsigned long flags, struct sbi_domain_memregion *reg)
- bool sbi_domain_check_addr (const struct sbi_domain *dom, unsigned long addr, unsigned long mode, unsigned long access_flags)
- static bool is_region_valid (const struct sbi_domain_memregion *reg)
- static bool is_region_subset (const struct sbi_domain_memregion *regA, const struct sbi_domain_memregion *regB)
- static bool is_region_compatible (const struct sbi_domain_memregion *regA, const struct sbi_domain_memregion *regB)
- static bool is_region_before (const struct sbi_domain_memregion *regA, const struct sbi_domain_memregion *regB)
- static const struct sbi_domain_memregion * find_region (const struct sbi_domain *dom, unsigned long addr)
- static const struct sbi_domain_memregion * find_next_subset_region (const struct sbi_domain *dom, const struct sbi_domain_memregion *reg, unsigned long addr)
- static void swap_region (struct sbi_domain_memregion *reg1, struct sbi_domain_memregion *reg2)
- static void clear_region (struct sbi_domain_memregion *reg)
- static int sanitize_domain (struct sbi_domain *dom)
- bool sbi_domain_check_addr_range (const struct sbi_domain *dom, unsigned long addr, unsigned long size, unsigned long mode, unsigned long access_flags)
- void sbi_domain_dump (const struct sbi_domain *dom, const char *suffix)
- void sbi_domain_dump_all (const char *suffix)
- int sbi_domain_register (struct sbi_domain *dom, const struct sbi_hartmask *assign_mask)
- static int root_add_memregion (const struct sbi_domain_memregion *reg)
- int sbi_domain_root_add_memrange (unsigned long addr, unsigned long size, unsigned long align, unsigned long region_flags)
- int sbi_domain_finalize (struct sbi_scratch *scratch, u32 cold_hartid)
- int sbi_domain_init (struct sbi_scratch *scratch, u32 cold_hartid)

Variables

- static u32 domain_count = 0
- static bool domain_finalized = false
- static u32 root_memregs_count = 0
- struct sbi_domain root
- static unsigned long domain_hart_ptr_offset

30.226.1 Macro Definition Documentation

30.226.1.1 ROOT_REGION_MAX

```
#define ROOT_REGION_MAX 32
```

30.226.2 Function Documentation

30.226.2.1 clear_region()

```
static void clear_region (
            struct sbi_domain_memregion * reg )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.2 find_next_subset_region()

```
static const struct sbi_domain_memregion * find_next_subset_region (
            const struct sbi_domain * dom,
            const struct sbi_domain_memregion * reg,
            unsigned long addr )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.3 find_region()

```
static const struct sbi_domain_memregion * find_region (
```

```
const struct sbi_domain * dom,
unsigned long addr )  [static]
```

Here is the caller graph for this function:

30.226.2.4 is_region_before()

```
static bool is_region_before (
            const struct sbi_domain_memregion * regA,
            const struct sbi_domain_memregion * regB )  [static]
```

Check if regionA should be placed before regionB Here is the caller graph for this function:

30.226.2.5 is_region_compatible()

```
static bool is_region_compatible (
            const struct sbi_domain_memregion * regA,
            const struct sbi_domain_memregion * regB )  [static]
```

Check if regionA can be replaced by regionB Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.6 is_region_subset()

```
static bool is_region_subset (
            const struct sbi_domain_memregion * regA,
            const struct sbi_domain_memregion * regB )  [static]
```

Check if regionA is sub-region of regionB Here is the caller graph for this function:

30.226.2.7 is_region_valid()

```
static bool is_region_valid (
            const struct sbi_domain_memregion * reg )  [static]
```

Here is the caller graph for this function:

30.226.2.8 root_add_memregion()

```
static int root_add_memregion (
            const struct sbi_domain_memregion * reg )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.9 sanitize_domain()

```
static int sanitize_domain (
            struct sbi_domain * dom )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.10 sbi_domain_check_addr()

```
bool sbi_domain_check_addr (
            const struct sbi_domain * dom,
            unsigned long addr,
            unsigned long mode,
            unsigned long access_flags )
```

Check whether we can access specified address for given mode and memory region flags under a domain

Parameters

dom	pointer to domain
addr	the address to be checked
mode	the privilege mode of access
access_flags	bitmask of domain access types (enum sbi_domain_access)

Returns

true if access allowed otherwise false

Here is the caller graph for this function:

30.226.2.11 sbi_domain_check_addr_range()

```
bool sbi_domain_check_addr_range (
            const struct sbi_domain * dom,
            unsigned long addr,
            unsigned long size,
            unsigned long mode,
            unsigned long access_flags )
```

Check whether we can access specified address range for given mode and memory region flags under a domain

Parameters

dom	pointer to domain
addr	the start of the address range to be checked
size	the size of the address range to be checked
mode	the privilege mode of access
access_flags	bitmask of domain access types (enum sbi_domain_access)

Returns

TRUE if access allowed otherwise FALSE

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.12 sbi_domain_dump()

```
void sbi_domain_dump (
            const struct sbi_domain * dom,
            const char * suffix )
```

Dump domain details on the console Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.13 sbi_domain_dump_all()

```
void sbi_domain_dump_all (
            const char * suffix )
```

Dump all domain details on the console Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.14 sbi_domain_finalize()

```
int sbi_domain_finalize (
            struct sbi_scratch * scratch,
            u32 cold_hartid )
```

Finalize domain tables and startup non-root domains Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.15 sbi_domain_get_assigned_hartmask()

```
int sbi_domain_get_assigned_hartmask (
            const struct sbi_domain * dom,
            struct sbi_hartmask * mask )
```

Get the assigned HART mask for given domain

Parameters

dom	pointer to domain
mask	the output hartmask to fill

Returns

0 on success and SBI_Exxx ($<$ 0) on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.16 sbi_domain_init()

```
int sbi_domain_init (
            struct sbi_scratch * scratch,
            u32 cold_hartid )
```

Initialize domains Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.17 sbi_domain_is_assigned_hart()

```
bool sbi_domain_is_assigned_hart (
            const struct sbi_domain * dom,
            u32 hartindex )
```

Check whether given HART is assigned to specified domain

Parameters

dom	pointer to domain
hartindex	the HART index

Returns

true if HART is assigned to domain otherwise false

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.18 sbi_domain_memregion_init()

```
void sbi_domain_memregion_init (
            unsigned long addr,
            unsigned long size,
            unsigned long flags,
            struct sbi_domain_memregion * reg )
```

Initialize a domain memory region based on it's physical address and size.

Parameters

addr	start physical address of memory region
size	physical size of memory region
flags	memory region flags
reg	pointer to memory region being initialized

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.19 sbi_domain_register()

```
int sbi_domain_register (
            struct sbi_domain * dom,
            const struct sbi_hartmask * assign_mask )
```

Register a new domain

Parameters

dom	pointer to domain
assign_mask	pointer to HART mask of HARTs assigned to the domain

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.20 sbi_domain_root_add_memrange()

```
int sbi_domain_root_add_memrange (
            unsigned long addr,
            unsigned long size,
            unsigned long align,
            unsigned long region_flags )
```

Add a memory range with its flags to the root domain

Parameters

addr	start physical address of memory range
size	physical size of memory range
align	alignment of memory region
region_flags	memory range flags

Returns

0 on success

SBI_EALREADY if memory region conflicts with the existing one

SBI_EINVAL otherwise

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.2.21 sbi_hartindex_to_domain()

```
struct sbi_domain * sbi_hartindex_to_domain (
            u32 hartindex )
```

Get pointer to sbi_domain from HART index Here is the caller graph for this function:

30.226.2.22 SBI_LIST_HEAD()

```
SBI_LIST_HEAD (
            domain_list  )
```

30.226.2.23 sbi_update_hartindex_to_domain()

```
void sbi_update_hartindex_to_domain (
            u32 hartindex,
            struct sbi_domain * dom )
```

Update HART local pointer to point to specified domain Here is the caller graph for this function:

30.226.2.24 swap_region()

```
static void swap_region (
            struct sbi_domain_memregion * reg1,
            struct sbi_domain_memregion * reg2 )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.226.3 Variable Documentation

30.226.3.1 domain_count

```
u32 domain_count = 0  [static]
```

30.226.3.2 domain_finalized

```
bool domain_finalized = false  [static]
```

30.226.3.3 domain_hart_ptr_offset

```
unsigned long domain_hart_ptr_offset  [static]
```

30.226.3.4 root

```
struct sbi_domain root
```

Initial value:

```
= {
    .name = "root",
    .possible_harts = NULL,
    .regions = NULL,
    .system_reset_allowed = true,
    .system_suspend_allowed = true,
    .fw_region_inited = false,
}
```

The root domain instance

30.226.3.5 root_memregs_count

```
u32 root_memregs_count = 0  [static]
```

30.227 lib/sbi/sbi_domain_context.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/riscv_locks.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
```

```
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_domain_context.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_domain_context.c:

Macros

- #define hart_context_thishart_get()

Functions

- static struct hart_context ∗ hart_context_get (struct sbi_domain ∗dom, u32 hartindex)
- static void hart_context_set (struct sbi_domain ∗dom, u32 hartindex, struct hart_context ∗hc)
- static void switch_to_next_domain_context (struct hart_context ∗ctx, struct hart_context ∗dom_ctx)
- int sbi_domain_context_enter (struct sbi_domain ∗dom)
- int sbi_domain_context_exit (void)
- int sbi_domain_context_init (void)
- void sbi_domain_context_deinit (void)

Variables

- static struct sbi_domain_data dcpriv

30.227.1 Macro Definition Documentation

30.227.1.1 hart_context_thishart_get

```
#define hart_context_thishart_get( )
```

Value:

```
hart_context_get(sbi_domain_thishart_ptr(),          \
        current_hartindex())
```

Macro to obtain the current hart's context pointer

30.227.2 Function Documentation

30.227.2.1 hart_context_get()

```
static struct hart_context * hart_context_get (
            struct sbi_domain * dom,
            u32 hartindex )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.227.2.2 hart_context_set()

```
static void hart_context_set (
            struct sbi_domain * dom,
            u32 hartindex,
            struct hart_context * hc )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.227.2.3 sbi_domain_context_deinit()

```
void sbi_domain_context_deinit (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.227.2.4 sbi_domain_context_enter()

```
int sbi_domain_context_enter (
            struct sbi_domain * dom )
```

Enter a specific domain context synchronously

Parameters

dom	pointer to domain

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

30.227.2.5 sbi_domain_context_exit()

```
int sbi_domain_context_exit (
            void  )
```

Exit the current domain context, and then return to the caller of sbi_domain_context_enter or attempt to start the next domain context to be initialized

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

30.227.2.6 sbi_domain_context_init()

```
int sbi_domain_context_init (
            void  )
```

Initialize domain context support

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.227.2.7 switch_to_next_domain_context()

```
static void switch_to_next_domain_context (
            struct hart_context * ctx,
            struct hart_context * dom_ctx )  [static]
```

Switches the HART context from the current domain to the target domain. This includes changing domain assignments and reconfiguring PMP, as well as saving and restoring CSRs and trap states.

Parameters

ctx	pointer to the current HART context
dom_ctx	pointer to the target domain context

Here is the call graph for this function:

Here is the caller graph for this function:

30.227.3 Variable Documentation

30.227.3.1 dcpriv

```
struct sbi_domain_data dcpriv  [static]
```

Initial value:

```
= {
    .data_size = sizeof(struct domain_context_priv),
}
```

30.228 lib/sbi/sbi_domain_data.c File Reference

```
#include <sbi/sbi_bitmap.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
```

Include dependency graph for sbi_domain_data.c:

Functions

- static SBI_LIST_HEAD (data_list)
- static DECLARE_BITMAP (data_idx_bmap, SBI_DOMAIN_MAX_DATA_PTRS)
- void ∗ sbi_domain_data_ptr (struct sbi_domain ∗dom, struct sbi_domain_data ∗data)
- static int domain_setup_data_one (struct sbi_domain ∗dom, struct sbi_domain_data ∗data)
- static void domain_cleanup_data_one (struct sbi_domain ∗dom, struct sbi_domain_data ∗data)
- int sbi_domain_setup_data (struct sbi_domain ∗dom)
- void sbi_domain_cleanup_data (struct sbi_domain ∗dom)
- int sbi_domain_register_data (struct sbi_domain_data ∗data)
- void sbi_domain_unregister_data (struct sbi_domain_data ∗data)

30.228.1 Function Documentation

30.228.1.1 DECLARE_BITMAP()

```
static DECLARE_BITMAP (
            data_idx_bmap ,
            SBI_DOMAIN_MAX_DATA_PTRS  )  [static]
```

30.228.1.2 domain_cleanup_data_one()

```
static void domain_cleanup_data_one (
            struct sbi_domain * dom,
            struct sbi_domain_data * data )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.228.1.3 domain_setup_data_one()

```
static int domain_setup_data_one (
            struct sbi_domain * dom,
            struct sbi_domain_data * data )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.228.1.4 sbi_domain_cleanup_data()

```
void sbi_domain_cleanup_data (
            struct sbi_domain * dom )
```

Cleanup all domain data for a domain

Parameters

dom	pointer to domain

Note: This function is used internally within domain framework. Here is the call graph for this function:

Here is the caller graph for this function:

30.228.1.5 sbi_domain_data_ptr()

```
void * sbi_domain_data_ptr (
            struct sbi_domain * dom,
            struct sbi_domain_data * data )
```

Get per-domain data pointer for a given domain

Parameters

dom	pointer to domain
data	pointer to domain data

Returns

per-domain data pointer

Here is the caller graph for this function:

30.228.1.6 sbi_domain_register_data()

```
int sbi_domain_register_data (
            struct sbi_domain_data * data )
```

Register a domain data

Parameters

hndl	pointer to domain data

Returns

0 on success and negative error code on failure

Note: This function must be used only in cold boot path. Here is the call graph for this function:

Here is the caller graph for this function:

30.228.1.7 sbi_domain_setup_data()

```
int sbi_domain_setup_data (
            struct sbi_domain * dom )
```

Setup all domain data for a domain

Parameters

dom	pointer to domain

Returns

0 on success and negative error code on failure

Note: This function is used internally within domain framework. Here is the call graph for this function:

Here is the caller graph for this function:

30.228.1.8 sbi_domain_unregister_data()

```
void sbi_domain_unregister_data (
            struct sbi_domain_data * data )
```

Unregister a domain data

Parameters

hndl	pointer to domain data

Note: This function must be used only in cold boot path. Here is the call graph for this function:

Here is the caller graph for this function:

30.228.1.9 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            data_list  )  [static]
```

30.229 lib/sbi/sbi_double_trap.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_sse.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_double_trap.c:

Functions

- int sbi_double_trap_handler (struct sbi_trap_context *tcntx)

30.229.1 Function Documentation

30.229.1.1 sbi_double_trap_handler()

```
int sbi_double_trap_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.230 lib/sbi/sbi_ecall.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_ecall.c:

Functions

- u16 sbi_ecall_version_major (void)
- u16 sbi_ecall_version_minor (void)
- unsigned long sbi_ecall_get_impid (void)
- void sbi_ecall_set_impid (unsigned long impid)
- static SBI_LIST_HEAD (ecall_exts_list)
- struct sbi_ecall_extension * sbi_ecall_find_extension (unsigned long extid)
- void sbi_ecall_get_extensions_str (char *exts_str, int exts_str_size, bool experimental)
- int sbi_ecall_register_extension (struct sbi_ecall_extension *ext)
- void sbi_ecall_unregister_extension (struct sbi_ecall_extension *ext)
- int sbi_ecall_handler (struct sbi_trap_context *tcntx)
- int sbi_ecall_init (void)

Variables

- struct sbi_ecall_extension *const sbi_ecall_exts [ ]
- static unsigned long ecall_impid = SBI_OPENSBI_IMPID

30.230.1 Function Documentation

30.230.1.1 sbi_ecall_find_extension()

```
struct sbi_ecall_extension * sbi_ecall_find_extension (
            unsigned long extid )
```

Here is the caller graph for this function:

30.230.1.2 sbi_ecall_get_extensions_str()

```
void sbi_ecall_get_extensions_str (
            char * exts_str,
            int exts_str_size,
            bool experimental )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.230.1.3 sbi_ecall_get_impid()

```
unsigned long sbi_ecall_get_impid (
            void )
```

Here is the caller graph for this function:

30.230.1.4 sbi_ecall_handler()

```
int sbi_ecall_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.230.1.5 sbi_ecall_init()

```
int sbi_ecall_init (
            void  )
```

Here is the caller graph for this function:

30.230.1.6 sbi_ecall_register_extension()

```
int sbi_ecall_register_extension (
            struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.230.1.7 sbi_ecall_set_impid()

```
void sbi_ecall_set_impid (
            unsigned long impid )
```

30.230.1.8 sbi_ecall_unregister_extension()

```
void sbi_ecall_unregister_extension (
```

```
            struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:

30.230.1.9 sbi_ecall_version_major()

```
u16 sbi_ecall_version_major (
            void  )
```

Here is the caller graph for this function:

30.230.1.10 sbi_ecall_version_minor()

```
u16 sbi_ecall_version_minor (
            void  )
```

Here is the caller graph for this function:

30.230.1.11 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            ecall_exts_list  )  [static]
```

30.230.2 Variable Documentation

30.230.2.1 ecall_impid

```
unsigned long ecall_impid = SBI_OPENSBI_IMPID  [static]
```

30.230.2.2 sbi_ecall_exts

```
struct sbi_ecall_extension* const sbi_ecall_exts[]  [extern]
```

30.231 lib/sbi/sbi_ecall_base.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
```

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_version.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_ecall_base.c:

Functions

- static int sbi_ecall_base_probe (unsigned long extid, unsigned long *out_val)
- static int sbi_ecall_base_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_base_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_base

30.231.1 Function Documentation

30.231.1.1 sbi_ecall_base_handler()

```
static int sbi_ecall_base_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.231.1.2 sbi_ecall_base_probe()

```
static int sbi_ecall_base_probe (
```

```
            unsigned long extid,
            unsigned long * out_val )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.231.1.3 sbi_ecall_base_register_extensions()

```
static int sbi_ecall_base_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.231.2 Variable Documentation

30.231.2.1 ecall_base

```
struct sbi_ecall_extension ecall_base
```

Initial value:

```
= {
    .name            = "base",
    .extid_start         = SBI_EXT_BASE,
    .extid_end       = SBI_EXT_BASE,
    .register_extensions     = sbi_ecall_base_register_extensions,
    .handle          = sbi_ecall_base_handler,
}
```

30.232 lib/sbi/sbi_ecall_cppc.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
```

```
#include <sbi/sbi_cppc.h>
```

Include dependency graph for sbi_ecall_cppc.c:

Functions

- static int sbi_ecall_cppc_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_cppc_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_cppc

30.232.1 Function Documentation

30.232.1.1 sbi_ecall_cppc_handler()

```
static int sbi_ecall_cppc_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.232.1.2 sbi_ecall_cppc_register_extensions()

```
static int sbi_ecall_cppc_register_extensions (
            void )  [static]
```

Here is the call graph for this function:

30.232.2 Variable Documentation

30.232.2.1 ecall_cppc

```
struct sbi_ecall_extension ecall_cppc
```

Initial value:

```
= {
    .name           = "cppc",
    .extid_start        = SBI_EXT_CPPC,
    .extid_end      = SBI_EXT_CPPC,
    .register_extensions    = sbi_ecall_cppc_register_extensions,
    .handle         = sbi_ecall_cppc_handler,
}
```

30.233 lib/sbi/sbi_ecall_dbcn.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_hart.h>
```

Include dependency graph for sbi_ecall_dbcn.c:

Functions

- static int sbi_ecall_dbcn_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_dbcn_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_dbcn

30.233.1 Function Documentation

30.233.1.1 sbi_ecall_dbcn_handler()

```
static int sbi_ecall_dbcn_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.233.1.2 sbi_ecall_dbcn_register_extensions()

```
static int sbi_ecall_dbcn_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.233.2 Variable Documentation

30.233.2.1 ecall_dbcn

```
struct sbi_ecall_extension ecall_dbcn
```

Initial value:

```
= {
    .name            = "dbcn",
    .extid_start         = SBI_EXT_DBCN,
    .extid_end       = SBI_EXT_DBCN,
    .register_extensions     = sbi_ecall_dbcn_register_extensions,
    .handle          = sbi_ecall_dbcn_handler,
}
```

30.234 lib/sbi/sbi_ecall_dbtr.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_dbtr.h>
```

Include dependency graph for sbi_ecall_dbtr.c:

Functions

- static int sbi_ecall_dbtr_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_dbtr_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_dbtr

30.234.1 Function Documentation

30.234.1.1 sbi_ecall_dbtr_handler()

```
static int sbi_ecall_dbtr_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.234.1.2 sbi_ecall_dbtr_register_extensions()

```
static int sbi_ecall_dbtr_register_extensions (
            void )  [static]
```

Here is the call graph for this function:

30.234.2 Variable Documentation

30.234.2.1 ecall_dbtr

```
struct sbi_ecall_extension ecall_dbtr
```

Initial value:

```
= {
    .name            = "dbtr",
    .extid_start         = SBI_EXT_DBTR,
    .extid_end       = SBI_EXT_DBTR,
    .experimental        = true,
    .handle          = sbi_ecall_dbtr_handler,
    .register_extensions     = sbi_ecall_dbtr_register_extensions,
}
```

30.235 lib/sbi/sbi_ecall_fwft.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_fwft.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_ecall_fwft.c:

Functions

- static int sbi_ecall_fwft_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_fwft_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_fwft

30.235.1 Function Documentation

30.235.1.1 sbi_ecall_fwft_handler()

```
static int sbi_ecall_fwft_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.235.1.2 sbi_ecall_fwft_register_extensions()

```
static int sbi_ecall_fwft_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.235.2 Variable Documentation

30.235.2.1 ecall_fwft

```
struct sbi_ecall_extension ecall_fwft
```

Initial value:

```
= {
    .name           = "fwft",
    .extid_start        = SBI_EXT_FWFT,
    .extid_end      = SBI_EXT_FWFT,
    .experimental       = true,
    .register_extensions    = sbi_ecall_fwft_register_extensions,
    .handle         = sbi_ecall_fwft_handler,
}
```

30.236 lib/sbi/sbi_ecall_hsm.c File Reference

```
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_scratch.h>
```

```
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_ecall_hsm.c:

Functions

- static int sbi_ecall_hsm_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_hsm_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_hsm

30.236.1 Function Documentation

30.236.1.1 sbi_ecall_hsm_handler()

```
static int sbi_ecall_hsm_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.236.1.2 sbi_ecall_hsm_register_extensions()

```
static int sbi_ecall_hsm_register_extensions (
            void ) [static]
```

Here is the call graph for this function:

30.236.2 Variable Documentation

30.236.2.1 ecall_hsm

```
struct sbi_ecall_extension ecall_hsm
```

Initial value:

```
= {
    .name            = "hsm",
    .extid_start         = SBI_EXT_HSM,
    .extid_end       = SBI_EXT_HSM,
    .register_extensions     = sbi_ecall_hsm_register_extensions,
    .handle          = sbi_ecall_hsm_handler,
}
```

30.237 lib/sbi/sbi_ecall_ipi.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_ipi.h>
```

Include dependency graph for sbi_ecall_ipi.c:

Functions

- static int sbi_ecall_ipi_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_ipi_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_ipi

30.237.1 Function Documentation

30.237.1.1 sbi_ecall_ipi_handler()

```
static int sbi_ecall_ipi_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.237.1.2 sbi_ecall_ipi_register_extensions()

```
static int sbi_ecall_ipi_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.237.2 Variable Documentation

30.237.2.1 ecall_ipi

```
struct sbi_ecall_extension ecall_ipi
```

Initial value:

```
= {
    .name           = "ipi",
    .extid_start        = SBI_EXT_IPI,
    .extid_end      = SBI_EXT_IPI,
    .register_extensions    = sbi_ecall_ipi_register_extensions,
    .handle         = sbi_ecall_ipi_handler,
}
```

30.238 lib/sbi/sbi_ecall_legacy.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_tlb.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
#include <sbi/sbi_hart.h>
```

Include dependency graph for sbi_ecall_legacy.c:

Functions

- static bool sbi_load_hart_mask_unpriv (ulong *pmask, ulong *hmask, ulong *hbase, struct sbi_trap_info *uptrap)
- static int sbi_ecall_legacy_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_legacy_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_legacy

30.238.1 Function Documentation

30.238.1.1 sbi_ecall_legacy_handler()

```
static int sbi_ecall_legacy_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.238.1.2 sbi_ecall_legacy_register_extensions()

```
static int sbi_ecall_legacy_register_extensions (
            void ) [static]
```

Here is the call graph for this function:

30.238.1.3 sbi_load_hart_mask_unpriv()

```
static bool sbi_load_hart_mask_unpriv (
            ulong * pmask,
            ulong * hmask,
            ulong * hbase,
            struct sbi_trap_info * uptrap ) [static]
```

Here is the caller graph for this function:

30.238.2 Variable Documentation

30.238.2.1 ecall_legacy

```
struct sbi_ecall_extension ecall_legacy
```

Initial value:

```
= {
    .name           = "legacy",
    .extid_start        = SBI_EXT_0_1_SET_TIMER,
    .extid_end      = SBI_EXT_0_1_SHUTDOWN,
    .register_extensions    = sbi_ecall_legacy_register_extensions,
    .handle         = sbi_ecall_legacy_handler,
}
```

30.239 lib/sbi/sbi_ecall_mpxy.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_mpxy.h>
```

Include dependency graph for sbi_ecall_mpxy.c:

Functions

- static int sbi_ecall_mpxy_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_mpxy_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_mpxy

30.239.1 Function Documentation

30.239.1.1 sbi_ecall_mpxy_handler()

```
static int sbi_ecall_mpxy_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.239.1.2 sbi_ecall_mpxy_register_extensions()

```
static int sbi_ecall_mpxy_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.239.2 Variable Documentation

30.239.2.1 ecall_mpxy

```
struct sbi_ecall_extension ecall_mpxy
```

Initial value:

```
= {
    .name           = "mpxy",
    .extid_start        = SBI_EXT_MPXY,
    .extid_end      = SBI_EXT_MPXY,
    .experimental       = true,
    .register_extensions    = sbi_ecall_mpxy_register_extensions,
    .handle         = sbi_ecall_mpxy_handler,
}
```

30.240 lib/sbi/sbi_ecall_pmu.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_version.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_ecall_pmu.c:

Functions

- static int sbi_ecall_pmu_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_pmu_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_pmu

30.240.1 Function Documentation

30.240.1.1 sbi_ecall_pmu_handler()

```
static int sbi_ecall_pmu_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.240.1.2 sbi_ecall_pmu_register_extensions()

```
static int sbi_ecall_pmu_register_extensions (
            void )  [static]
```

Here is the call graph for this function:

30.240.2 Variable Documentation

30.240.2.1 ecall_pmu

```
struct sbi_ecall_extension ecall_pmu
```

Initial value:

```
= {
    .name           = "pmu",
    .extid_start        = SBI_EXT_PMU,
    .extid_end      = SBI_EXT_PMU,
    .register_extensions    = sbi_ecall_pmu_register_extensions,
    .handle         = sbi_ecall_pmu_handler,
}
```

30.241 lib/sbi/sbi_ecall_rfence.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_tlb.h>
```

Include dependency graph for sbi_ecall_rfence.c:

Functions

- static int sbi_ecall_rfence_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_rfence_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_rfence

30.241.1 Function Documentation

30.241.1.1 sbi_ecall_rfence_handler()

```
static int sbi_ecall_rfence_handler (
            unsigned long extid,
```

```
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.241.1.2 sbi_ecall_rfence_register_extensions()

```
static int sbi_ecall_rfence_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.241.2 Variable Documentation

30.241.2.1 ecall_rfence

```
struct sbi_ecall_extension ecall_rfence
```

Initial value:

```
= {
    .name           = "rfnc",
    .extid_start        = SBI_EXT_RFENCE,
    .extid_end      = SBI_EXT_RFENCE,
    .register_extensions    = sbi_ecall_rfence_register_extensions,
    .handle         = sbi_ecall_rfence_handler,
}
```

30.242 lib/sbi/sbi_ecall_srst.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_system.h>
```

Include dependency graph for sbi_ecall_srst.c:

Functions

- static int sbi_ecall_srst_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static bool srst_available (void)
- static int sbi_ecall_srst_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_srst

30.242.1 Function Documentation

30.242.1.1 sbi_ecall_srst_handler()

```
static int sbi_ecall_srst_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.242.1.2 sbi_ecall_srst_register_extensions()

```
static int sbi_ecall_srst_register_extensions (
            void ) [static]
```

Here is the call graph for this function:

30.242.1.3 srst_available()

```
static bool srst_available (
            void ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.242.2 Variable Documentation

30.242.2.1 ecall_srst

```
struct sbi_ecall_extension ecall_srst
```

Initial value:

```
= {
    .name           = "srst",
    .extid_start        = SBI_EXT_SRST,
    .extid_end      = SBI_EXT_SRST,
    .register_extensions    = sbi_ecall_srst_register_extensions,
    .handle         = sbi_ecall_srst_handler,
}
```

30.243 lib/sbi/sbi_ecall_sse.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_sse.h>
```

Include dependency graph for sbi_ecall_sse.c:

Functions

- static int sbi_ecall_sse_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_sse_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_sse

30.243.1 Function Documentation

30.243.1.1 sbi_ecall_sse_handler()

```
static int sbi_ecall_sse_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.243.1.2 sbi_ecall_sse_register_extensions()

```
static int sbi_ecall_sse_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.243.2 Variable Documentation

30.243.2.1 ecall_sse

```
struct sbi_ecall_extension ecall_sse
```

Initial value:

```
= {
    .name           = "sse",
    .extid_start        = SBI_EXT_SSE,
    .extid_end      = SBI_EXT_SSE,
    .experimental       = true,
    .register_extensions    = sbi_ecall_sse_register_extensions,
    .handle         = sbi_ecall_sse_handler,
}
```

30.244 lib/sbi/sbi_ecall_susp.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_system.h>
```

Include dependency graph for sbi_ecall_susp.c:

Functions

- static int sbi_ecall_susp_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static bool susp_available (void)
- static int sbi_ecall_susp_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_susp

30.244.1 Function Documentation

30.244.1.1 sbi_ecall_susp_handler()

```
static int sbi_ecall_susp_handler (
            unsigned long extid,
            unsigned long funcid,
```

```
struct sbi_trap_regs * regs,
struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.244.1.2 sbi_ecall_susp_register_extensions()

```
static int sbi_ecall_susp_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.244.1.3 susp_available()

```
static bool susp_available (
            void  )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.244.2 Variable Documentation

30.244.2.1 ecall_susp

```
struct sbi_ecall_extension ecall_susp
```

Initial value:

```
= {
    .name           = "susp",
    .extid_start        = SBI_EXT_SUSP,
    .extid_end      = SBI_EXT_SUSP,
    .register_extensions    = sbi_ecall_susp_register_extensions,
    .handle         = sbi_ecall_susp_handler,
}
```

30.245 lib/sbi/sbi_ecall_time.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_ecall_time.c:

Functions

- static int sbi_ecall_time_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_time_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_time

30.245.1 Function Documentation

30.245.1.1 sbi_ecall_time_handler()

```
static int sbi_ecall_time_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.245.1.2 sbi_ecall_time_register_extensions()

```
static int sbi_ecall_time_register_extensions (
            void )  [static]
```

Here is the call graph for this function:

30.245.2 Variable Documentation

30.245.2.1 ecall_time

```
struct sbi_ecall_extension ecall_time
```

Initial value:

```
= {
    .name           = "time",
    .extid_start        = SBI_EXT_TIME,
    .extid_end      = SBI_EXT_TIME,
    .register_extensions    = sbi_ecall_time_register_extensions,
    .handle         = sbi_ecall_time_handler,
}
```

30.246 lib/sbi/sbi_ecall_vendor.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_trap.h>
```

```
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_ecall_vendor.c:

Functions

- static unsigned long sbi_ecall_vendor_id (void)
- static int sbi_ecall_vendor_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_vendor_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_vendor

30.246.1 Function Documentation

30.246.1.1 sbi_ecall_vendor_handler()

```
static int sbi_ecall_vendor_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

30.246.1.2 sbi_ecall_vendor_id()

```
static unsigned long sbi_ecall_vendor_id (
            void  )  [inline], [static]
```

Here is the caller graph for this function:

30.246.1.3 sbi_ecall_vendor_register_extensions()

```
static int sbi_ecall_vendor_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.246.2 Variable Documentation

30.246.2.1 ecall_vendor

```
struct sbi_ecall_extension ecall_vendor
```

Initial value:

```
= {
    .name           = "vendor",
    .extid_start        = SBI_EXT_VENDOR_START,
    .extid_end      = SBI_EXT_VENDOR_END,
    .register_extensions    = sbi_ecall_vendor_register_extensions,
    .handle         = sbi_ecall_vendor_handler,
}
```

30.247 lib/sbi/sbi_emulate_csr.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_emulate_csr.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_emulate_csr.c:

Macros

- #define switchcase_hpm(__uref, __mref, __csr)
- #define switchcase_hpm_2(__uref, __mref, __csr)
- #define switchcase_hpm_4(__uref, __mref, __csr)
- #define switchcase_hpm_8(__uref, __mref, __csr)
- #define switchcase_hpm_16(__uref, __mref, __csr)

Functions

- static bool hpm_allowed (int hpm_num, ulong prev_mode, bool virt)
- int sbi_emulate_csr_read (int csr_num, struct sbi_trap_regs *regs, ulong *csr_val)
- int sbi_emulate_csr_write (int csr_num, struct sbi_trap_regs *regs, ulong csr_val)

30.247.1 Macro Definition Documentation

30.247.1.1 switchcase_hpm

```
#define switchcase_hpm(
            __uref,
            __mref,
            __csr )
```

Value:

```
    case __csr:                                 \
        if (sbi_hart_mhpm_mask(scratch) & (1 « (__csr - __uref)))\
            return SBI_ENOTSUPP;                \
        if (!hpm_allowed(__csr - __uref, prev_mode, virt))    \
            return SBI_ENOTSUPP;                \
        *csr_val = csr_read(__mref + __csr - __uref);        \
        break;
```

30.247.1.2 switchcase_hpm_16

```
#define switchcase_hpm_16(
            __uref,
            __mref,
            __csr )
```

Value:

```
    switchcase_hpm_8(__uref, __mref, __csr + 0)          \
    switchcase_hpm_8(__uref, __mref, __csr + 8)
```

30.247.1.3 switchcase_hpm_2

```
#define switchcase_hpm_2(
            __uref,
            __mref,
            __csr )
```

Value:

```
    switchcase_hpm(__uref, __mref, __csr + 0)          \
    switchcase_hpm(__uref, __mref, __csr + 1)
```

30.247.1.4 switchcase_hpm_4

```
#define switchcase_hpm_4(
            __uref,
            __mref,
            __csr )
```

Value:

```
    switchcase_hpm_2(__uref, __mref, __csr + 0)          \
    switchcase_hpm_2(__uref, __mref, __csr + 2)
```

30.247.1.5 switchcase_hpm_8

```
#define switchcase_hpm_8(
            __uref,
            __mref,
            __csr )
```

Value:

```
    switchcase_hpm_4(__uref, __mref, __csr + 0)          \
    switchcase_hpm_4(__uref, __mref, __csr + 4)
```

30.247.2 Function Documentation

30.247.2.1 hpm_allowed()

```
static bool hpm_allowed (
            int hpm_num,
            ulong prev_mode,
            bool virt )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.247.2.2 sbi_emulate_csr_read()

```
int sbi_emulate_csr_read (
            int csr_num,
            struct sbi_trap_regs * regs,
            ulong * csr_val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.247.2.3 sbi_emulate_csr_write()

```
int sbi_emulate_csr_write (
            int csr_num,
            struct sbi_trap_regs * regs,
            ulong csr_val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.248 lib/sbi/sbi_fifo.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_fifo.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_fifo.c:

Functions

- void sbi_fifo_init (struct sbi_fifo *fifo, void *queue_mem, u16 entries, u16 entry_size)

- static bool __sbi_fifo_is_full (struct sbi_fifo *fifo)
- u16 sbi_fifo_avail (struct sbi_fifo *fifo)
- int sbi_fifo_is_full (struct sbi_fifo *fifo)
- static void __sbi_fifo_enqueue (struct sbi_fifo *fifo, void *data)
- static void __sbi_fifo_dequeue (struct sbi_fifo *fifo, void *data)
- static bool __sbi_fifo_is_empty (struct sbi_fifo *fifo)
- int sbi_fifo_is_empty (struct sbi_fifo *fifo)
- static void __sbi_fifo_reset (struct sbi_fifo *fifo)
- bool sbi_fifo_reset (struct sbi_fifo *fifo)
- int sbi_fifo_inplace_update (struct sbi_fifo *fifo, void *in, int(*fptr)(void *in, void *data))
- int sbi_fifo_enqueue (struct sbi_fifo *fifo, void *data, bool force)
- int sbi_fifo_dequeue (struct sbi_fifo *fifo, void *data)

30.248.1 Function Documentation

30.248.1.1 __sbi_fifo_dequeue()

```
static void __sbi_fifo_dequeue (
            struct sbi_fifo * fifo,
            void * data )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.248.1.2 __sbi_fifo_enqueue()

```
static void __sbi_fifo_enqueue (
            struct sbi_fifo * fifo,
            void * data )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.248.1.3 __sbi_fifo_is_empty()

```
static bool __sbi_fifo_is_empty (
            struct sbi_fifo * fifo )  [inline], [static]
```

Here is the caller graph for this function:

30.248.1.4 __sbi_fifo_is_full()

```
static bool __sbi_fifo_is_full (
            struct sbi_fifo * fifo )  [inline], [static]
```

Here is the caller graph for this function:

30.248.1.5 __sbi_fifo_reset()

```
static void __sbi_fifo_reset (
            struct sbi_fifo * fifo )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.248.1.6 sbi_fifo_avail()

```
u16 sbi_fifo_avail (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

30.248.1.7 sbi_fifo_dequeue()

```
int sbi_fifo_dequeue (
            struct sbi_fifo * fifo,
            void * data )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.248.1.8 sbi_fifo_enqueue()

```
int sbi_fifo_enqueue (
            struct sbi_fifo * fifo,
            void * data,
            bool force )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.248.1.9 sbi_fifo_init()

```
void sbi_fifo_init (
            struct sbi_fifo * fifo,
            void * queue_mem,
            u16 entries,
            u16 entry_size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.248.1.10 sbi_fifo_inplace_update()

```
int sbi_fifo_inplace_update (
            struct sbi_fifo * fifo,
            void * in,
            int(*)(void *in, void *data) fptr )
```

Provide a helper function to do inplace update to the fifo. Note: The callback function is called with lock being held. **Do not** invoke any other fifo function from callback. Otherwise, it will lead to deadlock. Here is the call graph for this function:

Here is the caller graph for this function:

30.248.1.11 sbi_fifo_is_empty()

```
int sbi_fifo_is_empty (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

30.248.1.12 sbi_fifo_is_full()

```
int sbi_fifo_is_full (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

30.248.1.13 sbi_fifo_reset()

```
bool sbi_fifo_reset (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

30.249 lib/sbi/sbi_fwft.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_bitmap.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_types.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
```

Include dependency graph for sbi_fwft.c:

Macros

- #define fwft_get_hart_state_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, fwft_ptr_offset)
- #define fwft_thishart_state_ptr() fwft_get_hart_state_ptr(sbi_scratch_thishart_ptr())
- #define fwft_set_hart_state_ptr(__scratch, __phs) sbi_scratch_write_type((__scratch), void *, fwft_ptr_offset, (__phs))
- #define MIS_DELEG (1UL << CAUSE_MISALIGNED_LOAD | 1UL << CAUSE_MISALIGNED_STORE)

Functions

- static bool fwft_is_defined_feature (enum sbi_fwft_feature_t feature)
- static int fwft_menvcfg_set_bit (unsigned long value, unsigned long bit)
- static int fwft_menvcfg_read_bit (unsigned long *value, unsigned long bit)

- static int fwft_misaligned_delegation_supported (struct fwft_config ∗conf)
- static int fwft_set_misaligned_delegation (struct fwft_config ∗conf, unsigned long value)
- static int fwft_get_misaligned_delegation (struct fwft_config ∗conf, unsigned long ∗value)
- static int fwft_double_trap_supported (struct fwft_config ∗conf)
- static int fwft_set_double_trap (struct fwft_config ∗conf, unsigned long value)
- static int fwft_get_double_trap (struct fwft_config ∗conf, unsigned long ∗value)
- static int fwft_adue_supported (struct fwft_config ∗conf)
- static int fwft_set_adue (struct fwft_config ∗conf, unsigned long value)
- static int fwft_get_adue (struct fwft_config ∗conf, unsigned long ∗value)
- static int fwft_lpad_supported (struct fwft_config ∗conf)
- static int fwft_enable_lpad (struct fwft_config ∗conf, unsigned long value)
- static int fwft_get_lpad (struct fwft_config ∗conf, unsigned long ∗value)
- static int fwft_sstack_supported (struct fwft_config ∗conf)
- static int fwft_enable_sstack (struct fwft_config ∗conf, unsigned long value)
- static int fwft_get_sstack (struct fwft_config ∗conf, unsigned long ∗value)
- static struct fwft_config ∗ get_feature_config (enum sbi_fwft_feature_t feature)
- static int fwft_get_feature (enum sbi_fwft_feature_t feature, struct fwft_config ∗∗conf)
- static void fwft_clear_config_lock (enum sbi_fwft_feature_t feature)
- int sbi_fwft_set (enum sbi_fwft_feature_t feature, unsigned long value, unsigned long flags)
- int sbi_fwft_get (enum sbi_fwft_feature_t feature, unsigned long ∗out_val)
- int sbi_fwft_init (struct sbi_scratch ∗scratch, bool cold_boot)

Variables

- static unsigned long fwft_ptr_offset
- static const unsigned long fwft_defined_features [ ]
- static const struct fwft_feature features [ ]

30.249.1 Macro Definition Documentation

30.249.1.1 fwft_get_hart_state_ptr

```
#define fwft_get_hart_state_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, fwft_ptr_offset)
```

30.249.1.2 fwft_set_hart_state_ptr

```
#define fwft_set_hart_state_ptr(
            __scratch,
            __phs )  sbi_scratch_write_type((__scratch), void *, fwft_ptr_offset, (__phs))
```

30.249.1.3 fwft_thishart_state_ptr

```
#define fwft_thishart_state_ptr( )  fwft_get_hart_state_ptr(sbi_scratch_thishart_ptr())
```

30.249.1.4 MIS_DELEG

```
#define MIS_DELEG (1UL << CAUSE_MISALIGNED_LOAD | 1UL << CAUSE_MISALIGNED_STORE)
```

30.249.2 Function Documentation

30.249.2.1 fwft_adue_supported()

```
static int fwft_adue_supported (
            struct fwft_config * conf )  [static]
```

Here is the call graph for this function:

30.249.2.2 fwft_clear_config_lock()

```
static void fwft_clear_config_lock (
            enum sbi_fwft_feature_t feature )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.249.2.3 fwft_double_trap_supported()

```
static int fwft_double_trap_supported (
            struct fwft_config * conf )  [static]
```

Here is the call graph for this function:

30.249.2.4 fwft_enable_lpad()

```
static int fwft_enable_lpad (
            struct fwft_config * conf,
            unsigned long value )  [static]
```

Here is the call graph for this function:

30.249.2.5 fwft_enable_sstack()

```
static int fwft_enable_sstack (
            struct fwft_config * conf,
            unsigned long value )  [static]
```

Here is the call graph for this function:

30.249.2.6 fwft_get_adue()

```
static int fwft_get_adue (
            struct fwft_config * conf,
            unsigned long * value )  [static]
```

Here is the call graph for this function:

30.249.2.7 fwft_get_double_trap()

```
static int fwft_get_double_trap (
            struct fwft_config * conf,
            unsigned long * value )  [static]
```

Here is the call graph for this function:

30.249.2.8 fwft_get_feature()

```
static int fwft_get_feature (
            enum sbi_fwft_feature_t feature,
            struct fwft_config ** conf )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.249.2.9 fwft_get_lpad()

```
static int fwft_get_lpad (
            struct fwft_config * conf,
            unsigned long * value )  [static]
```

Here is the call graph for this function:

30.249.2.10 fwft_get_misaligned_delegation()

```
static int fwft_get_misaligned_delegation (
            struct fwft_config * conf,
            unsigned long * value )  [static]
```

30.249.2.11 fwft_get_sstack()

```
static int fwft_get_sstack (
            struct fwft_config * conf,
            unsigned long * value )  [static]
```

Here is the call graph for this function:

30.249.2.12 fwft_is_defined_feature()

```
static bool fwft_is_defined_feature (
            enum sbi_fwft_feature_t feature )  [static]
```

Here is the caller graph for this function:

30.249.2.13 fwft_lpad_supported()

```
static int fwft_lpad_supported (
            struct fwft_config * conf )  [static]
```

Here is the call graph for this function:

30.249.2.14 fwft_menvcfg_read_bit()

```
static int fwft_menvcfg_read_bit (
            unsigned long * value,
            unsigned long bit )  [static]
```

Here is the caller graph for this function:

30.249.2.15 fwft_menvcfg_set_bit()

```
static int fwft_menvcfg_set_bit (
            unsigned long value,
            unsigned long bit )  [static]
```

Here is the caller graph for this function:

30.249.2.16 fwft_misaligned_delegation_supported()

```
static int fwft_misaligned_delegation_supported (
            struct fwft_config * conf )  [static]
```

30.249.2.17 fwft_set_adue()

```
static int fwft_set_adue (
            struct fwft_config * conf,
            unsigned long value )  [static]
```

Here is the call graph for this function:

30.249.2.18 fwft_set_double_trap()

```
static int fwft_set_double_trap (
            struct fwft_config * conf,
            unsigned long value )  [static]
```

Here is the call graph for this function:

30.249.2.19 fwft_set_misaligned_delegation()

```
static int fwft_set_misaligned_delegation (
            struct fwft_config * conf,
            unsigned long value )  [static]
```

30.249.2.20 fwft_sstack_supported()

```
static int fwft_sstack_supported (
            struct fwft_config * conf )  [static]
```

Here is the call graph for this function:

30.249.2.21 get_feature_config()

```
static struct fwft_config * get_feature_config (
            enum sbi_fwft_feature_t feature )  [static]
```

Here is the caller graph for this function:

30.249.2.22 sbi_fwft_get()

```
int sbi_fwft_get (
            enum sbi_fwft_feature_t feature,
            unsigned long * out_val )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.249.2.23 sbi_fwft_init()

```
int sbi_fwft_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.249.2.24 sbi_fwft_set()

```
int sbi_fwft_set (
            enum sbi_fwft_feature_t feature,
```

```
            unsigned long value,
            unsigned long flags )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.249.3 Variable Documentation

30.249.3.1 features

```
const struct fwft_feature features[]  [static]
```

30.249.3.2 fwft_defined_features

```
const unsigned long fwft_defined_features[]  [static]
```

Initial value:

```
= {
    SBI_FWFT_MISALIGNED_EXC_DELEG,
    SBI_FWFT_LANDING_PAD,
    SBI_FWFT_SHADOW_STACK,
    SBI_FWFT_DOUBLE_TRAP,
    SBI_FWFT_PTE_AD_HW_UPDATING,
    SBI_FWFT_POINTER_MASKING_PMLEN,
}
```

30.249.3.3 fwft_ptr_offset

```
unsigned long fwft_ptr_offset  [static]
```

Offset of pointer to FWFT HART state in scratch space

30.250 lib/sbi/sbi_hart.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_fp.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_csr_detect.h>
```

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_math.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_hfence.h>
```

Include dependency graph for sbi_hart.c:

Macros

- #define __set_menvcfg_ext(__ext, __bits)
- #define __SBI_HART_EXT_DATA(_name, _id)
- #define __check_hpm_csr(__csr, __mask)
- #define __check_hpm_csr_2(__csr, __mask)
- #define __check_hpm_csr_4(__csr, __mask)
- #define __check_hpm_csr_8(__csr, __mask)
- #define __check_hpm_csr_16(__csr, __mask)
- #define __check_csr(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_2(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_4(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_8(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_16(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_32(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_64(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_priv(__csr, __base_priv, __priv)
- #define __check_ext_csr(__base_priv, __csr, __ext)

Functions

- void __sbi_expected_trap (void)
- void __sbi_expected_trap_hext (void)
- static void mstatus_init (struct sbi_scratch *scratch)
- static int fp_init (struct sbi_scratch *scratch)
- static int delegate_traps (struct sbi_scratch *scratch)
- void sbi_hart_delegation_dump (struct sbi_scratch *scratch, const char *prefix, const char *suffix)
- unsigned int sbi_hart_mhpm_mask (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_count (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_log2gran (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_addrbits (struct sbi_scratch *scratch)
- unsigned int sbi_hart_mhpm_bits (struct sbi_scratch *scratch)
- static unsigned int sbi_hart_get_smepmp_flags (struct sbi_scratch *scratch, struct sbi_domain *dom, struct sbi_domain_memregion *reg)

- static void sbi_hart_smepmp_set (struct sbi_scratch *scratch, struct sbi_domain *dom, struct sbi_domain_memregion *reg, unsigned int pmp_idx, unsigned int pmp_flags, unsigned int pmp_log2gran, unsigned long pmp_addr↩_max)
- static int sbi_hart_smepmp_configure (struct sbi_scratch *scratch, unsigned int pmp_count, unsigned int pmp_log2gran, unsigned long pmp_addr_max)
- static int sbi_hart_oldpmp_configure (struct sbi_scratch *scratch, unsigned int pmp_count, unsigned int pmp_log2gran, unsigned long pmp_addr_max)
- int sbi_hart_map_saddr (unsigned long addr, unsigned long size)
- int sbi_hart_unmap_saddr (void)
- int sbi_hart_pmp_configure (struct sbi_scratch *scratch)
- int sbi_hart_priv_version (struct sbi_scratch *scratch)
- void sbi_hart_get_priv_version_str (struct sbi_scratch *scratch, char *version_str, int nvstr)
- static void __sbi_hart_update_extension (struct sbi_hart_features *hfeatures, enum sbi_hart_extensions ext, bool enable)
- void sbi_hart_update_extension (struct sbi_scratch *scratch, enum sbi_hart_extensions ext, bool enable)
- bool sbi_hart_has_extension (struct sbi_scratch *scratch, enum sbi_hart_extensions ext)
- void sbi_hart_get_extensions_str (struct sbi_scratch *scratch, char *extensions_str, int nestr)
- static unsigned long hart_pmp_get_allowed_addr (void)
- static int hart_mhpm_get_allowed_bits (void)
- static int hart_detect_features (struct sbi_scratch *scratch)
- int sbi_hart_reinit (struct sbi_scratch *scratch)
- int sbi_hart_init (struct sbi_scratch *scratch, bool cold_boot)
- void __attribute__ ((noreturn))

Variables

- void(* sbi_hart_expected_trap)(void) = &__sbi_expected_trap
- static unsigned long hart_features_offset
- const struct sbi_hart_ext_data sbi_hart_ext [ ]

30.250.1 Macro Definition Documentation

30.250.1.1 __check_csr

```
#define __check_csr(
            __csr,
            __rdonly,
            __wrval,
            __field,
            __skip )
```

Value:

```
oldval = csr_read_allowed(__csr, &trap);            \
if (!trap.cause) {                                  \
    if (__rdonly) {                                 \
        (hfeatures->__field)++;                     \
    } else {                                        \
        csr_write_allowed(__csr, &trap, __wrval);   \
        if (!trap.cause) {                          \
            if ((csr_swap(__csr, oldval) & __wrval) == __wrval) \
                (hfeatures->__field)++;             \
            else                                    \
                goto __skip;                        \
        } else {                                    \
            goto __skip;                            \
        }                                           \
    }                                               \
} else {                                            \
    goto __skip;                                    \
}
```

30.250.1.2 __check_csr_16

```
#define __check_csr_16(
            __csr,
            __rdonly,
            __wrval,
            __field,
            __skip )
```

Value:

```
__check_csr_8(__csr + 0, __rdonly, __wrval, __field, __skip)   \
__check_csr_8(__csr + 8, __rdonly, __wrval, __field, __skip)
```

30.250.1.3 __check_csr_2

```
#define __check_csr_2(
            __csr,
            __rdonly,
            __wrval,
            __field,
            __skip )
```

Value:

```
__check_csr(__csr + 0, __rdonly, __wrval, __field, __skip)   \
__check_csr(__csr + 1, __rdonly, __wrval, __field, __skip)
```

30.250.1.4 __check_csr_32

```
#define __check_csr_32(
            __csr,
            __rdonly,
            __wrval,
            __field,
            __skip )
```

Value:

```
__check_csr_16(__csr + 0, __rdonly, __wrval, __field, __skip) \
__check_csr_16(__csr + 16, __rdonly, __wrval, __field, __skip)
```

30.250.1.5 __check_csr_4

```
#define __check_csr_4(
            __csr,
            __rdonly,
            __wrval,
            __field,
            __skip )
```

Value:

```
__check_csr_2(__csr + 0, __rdonly, __wrval, __field, __skip)   \
__check_csr_2(__csr + 2, __rdonly, __wrval, __field, __skip)
```

30.250.1.6 __check_csr_64

```
#define __check_csr_64(
            __csr,
            __rdonly,
            __wrval,
            __field,
            __skip )
```

Value:

```
__check_csr_32(__csr + 0, __rdonly, __wrval, __field, __skip) \
__check_csr_32(__csr + 32, __rdonly, __wrval, __field, __skip)
```

30.250.1.7 __check_csr_8

```
#define __check_csr_8(
```

```
                __csr,
                __rdonly,
                __wrval,
                __field,
                __skip )
```

Value:

```
__check_csr_4(__csr + 0, __rdonly, __wrval, __field, __skip)    \
__check_csr_4(__csr + 4, __rdonly, __wrval, __field, __skip)
```

30.250.1.8 __check_ext_csr

```
#define __check_ext_csr(
                __base_priv,
                __csr,
                __ext )
```

Value:

```
if (hfeatures->priv_version >= __base_priv) {            \
    csr_read_allowed(__csr, &trap);              \
    if (!trap.cause)                    \
        __sbi_hart_update_extension(hfeatures,      \
                        __ext, true);    \
}
```

30.250.1.9 __check_hpm_csr

```
#define __check_hpm_csr(
                __csr,
                __mask )
```

Value:

```
oldval = csr_read_allowed(__csr, &trap);              \
if (!trap.cause) {                          \
    csr_write_allowed(__csr, &trap, 1UL);             \
    if (!trap.cause && csr_swap(__csr, oldval) == 1UL) {      \
        (hfeatures->__mask) |= 1 « (__csr - CSR_MCYCLE); \
    }                                   \
}
```

30.250.1.10 __check_hpm_csr_16

```
#define __check_hpm_csr_16(
                __csr,
                __mask )
```

Value:

```
__check_hpm_csr_8(__csr + 0, __mask)              \
__check_hpm_csr_8(__csr + 8, __mask)
```

30.250.1.11 __check_hpm_csr_2

```
#define __check_hpm_csr_2(
                __csr,
                __mask )
```

Value:

```
__check_hpm_csr(__csr + 0, __mask)              \
__check_hpm_csr(__csr + 1, __mask)
```

30.250.1.12 __check_hpm_csr_4

```
#define __check_hpm_csr_4(
                __csr,
                __mask )
```

Value:

```
__check_hpm_csr_2(__csr + 0, __mask)              \
__check_hpm_csr_2(__csr + 2, __mask)
```

30.250.1.13 __check_hpm_csr_8

```
#define __check_hpm_csr_8(
            __csr,
            __mask )
```

Value:

```
__check_hpm_csr_4(__csr + 0, __mask)            \
__check_hpm_csr_4(__csr + 4, __mask)
```

30.250.1.14 __check_priv

```
#define __check_priv(
            __csr,
            __base_priv,
            __priv )
```

Value:

```
val = csr_read_allowed(__csr, &trap);               \
if (!trap.cause && (hfeatures->priv_version >= __base_priv)) {  \
    hfeatures->priv_version = __priv;           \
}
```

30.250.1.15 __SBI_HART_EXT_DATA

```
#define __SBI_HART_EXT_DATA(
            _name,
            _id )
```

Value:

```
{   \
.name = #_name,             \
.id = _id,              \
}
```

30.250.1.16 __set_menvcfg_ext

```
#define __set_menvcfg_ext(
            __ext,
            __bits )
```

Value:

```
    if (sbi_hart_has_extension(scratch, __ext))       \
        menvcfg_val |= __bits;
```

30.250.2 Function Documentation

30.250.2.1 __attribute__()

```
void __attribute__ (
            (noreturn)  )
```

30.250.2.2 __sbi_expected_trap()

```
void __sbi_expected_trap (
            void  )  [extern]
```

30.250.2.3 __sbi_expected_trap_hext()

```
void __sbi_expected_trap_hext (
            void  )  [extern]
```

Here is the caller graph for this function:

30.250.2.4 __sbi_hart_update_extension()

```
static void __sbi_hart_update_extension (
            struct sbi_hart_features * hfeatures,
            enum sbi_hart_extensions ext,
            bool enable )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.5 delegate_traps()

```
static int delegate_traps (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.6 fp_init()

```
static int fp_init (
            struct sbi_scratch * scratch )  [static]
```

Here is the caller graph for this function:

30.250.2.7 hart_detect_features()

```
static int hart_detect_features (
            struct sbi_scratch * scratch )  [static]
```

Detect the allowed address bits & granularity. At least PMPADDR0 should be implemented.
No need to check for MHPMCOUNTERH for RV32 as they are expected to be implemented if MHPMCOUNTER is implemented.Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.8 hart_mhpm_get_allowed_bits()

```
static int hart_mhpm_get_allowed_bits (
            void ) [static]
```

It is assumed that platforms will implement same number of bits for all the performance counters including mcycle/minstret.Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.9 hart_pmp_get_allowed_addr()

```
static unsigned long hart_pmp_get_allowed_addr (
            void ) [static]
```

Here is the caller graph for this function:

30.250.2.10 mstatus_init()

```
static void mstatus_init (
            struct sbi_scratch * scratch ) [static]
```

OpenSBI doesn't use any PMU counters in M-mode. Supervisor mode usage for all counters are enabled by default But counters will not run until mcountinhibit is set.

The mhpmeventn[h] CSR should be initialized with interrupt disabled and inhibited running in M-mode during init.Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.11 sbi_hart_delegation_dump()

```
void sbi_hart_delegation_dump (
            struct sbi_scratch * scratch,
            const char * prefix,
            const char * suffix )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.12 sbi_hart_get_extensions_str()

```
void sbi_hart_get_extensions_str (
            struct sbi_scratch * scratch,
            char * extensions_str,
            int nestr )
```

Get the hart extensions in string format

Parameters

scratch	pointer to the HART scratch space
extensions_str	pointer to a char array where the extensions string will be updated
nestr	length of the features_str. The feature string will be truncated if nestr is not long enough.

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.13 sbi_hart_get_priv_version_str()

```
void sbi_hart_get_priv_version_str (
            struct sbi_scratch * scratch,
            char * version_str,
            int nvstr )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.14 sbi_hart_get_smepmp_flags()

```
static unsigned int sbi_hart_get_smepmp_flags (
            struct sbi_scratch * scratch,
```

```
struct sbi_domain * dom,
struct sbi_domain_memregion * reg )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.15 sbi_hart_has_extension()

```
bool sbi_hart_has_extension (
            struct sbi_scratch * scratch,
            enum sbi_hart_extensions ext )
```

Check whether a particular hart extension is available

Parameters

scratch	pointer to the HART scratch space
ext	the extension number to check

Returns

true (available) or false (not available)

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.16 sbi_hart_init()

```
int sbi_hart_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.17 sbi_hart_map_saddr()

```
int sbi_hart_map_saddr (
            unsigned long addr,
            unsigned long size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.18 sbi_hart_mhpm_bits()

```
unsigned int sbi_hart_mhpm_bits (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.250.2.19 sbi_hart_mhpm_mask()

```
unsigned int sbi_hart_mhpm_mask (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.250.2.20 sbi_hart_oldpmp_configure()

```
static int sbi_hart_oldpmp_configure (
            struct sbi_scratch * scratch,
            unsigned int pmp_count,
            unsigned int pmp_log2gran,
            unsigned long pmp_addr_max )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.21 sbi_hart_pmp_addrbits()

```
unsigned int sbi_hart_pmp_addrbits (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.250.2.22 sbi_hart_pmp_configure()

```
int sbi_hart_pmp_configure (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.23 sbi_hart_pmp_count()

```
unsigned int sbi_hart_pmp_count (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.250.2.24 sbi_hart_pmp_log2gran()

```
unsigned int sbi_hart_pmp_log2gran (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.250.2.25 sbi_hart_priv_version()

```
int sbi_hart_priv_version (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.250.2.26 sbi_hart_reinit()

```
int sbi_hart_reinit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.27 sbi_hart_smepmp_configure()

```
static int sbi_hart_smepmp_configure (
            struct sbi_scratch * scratch,
            unsigned int pmp_count,
            unsigned int pmp_log2gran,
            unsigned long pmp_addr_max )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.28 sbi_hart_smepmp_set()

```
static void sbi_hart_smepmp_set (
            struct sbi_scratch * scratch,
```

```
struct sbi_domain * dom,
struct sbi_domain_memregion * reg,
unsigned int pmp_idx,
unsigned int pmp_flags,
unsigned int pmp_log2gran,
unsigned long pmp_addr_max )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.29 sbi_hart_unmap_saddr()

```
int sbi_hart_unmap_saddr (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.250.2.30 sbi_hart_update_extension()

```
void sbi_hart_update_extension (
            struct sbi_scratch * scratch,
            enum sbi_hart_extensions ext,
            bool enable )
```

Enable/Disable a particular hart extension

Parameters

scratch	pointer to the HART scratch space
ext	the extension number to check
enable	new state of hart extension

Here is the call graph for this function:

30.250.3 Variable Documentation

30.250.3.1 hart_features_offset

```
unsigned long hart_features_offset [static]
```

30.250.3.2 sbi_hart_expected_trap

```
void(* sbi_hart_expected_trap) (void) (
            void  ) = &__sbi_expected_trap
```

30.250.3.3 sbi_hart_ext

```
const struct sbi_hart_ext_data sbi_hart_ext[]
```

Initial value:

```
= {
    __SBI_HART_EXT_DATA(smaia, SBI_HART_EXT_SMAIA),
    __SBI_HART_EXT_DATA(smepmp, SBI_HART_EXT_SMEPMP),
    __SBI_HART_EXT_DATA(smstateen, SBI_HART_EXT_SMSTATEEN),
    __SBI_HART_EXT_DATA(sscofpmf, SBI_HART_EXT_SSCOFPMF),
    __SBI_HART_EXT_DATA(sstc, SBI_HART_EXT_SSTC),
    __SBI_HART_EXT_DATA(zicntr, SBI_HART_EXT_ZICNTR),
    __SBI_HART_EXT_DATA(zihpm, SBI_HART_EXT_ZIHPM),
    __SBI_HART_EXT_DATA(zkr, SBI_HART_EXT_ZKR),
    __SBI_HART_EXT_DATA(smcntrpmf, SBI_HART_EXT_SMCNTRPMF),
    __SBI_HART_EXT_DATA(xandespmu, SBI_HART_EXT_XANDESPMU),
    __SBI_HART_EXT_DATA(zicboz, SBI_HART_EXT_ZICBOZ),
    __SBI_HART_EXT_DATA(zicbom, SBI_HART_EXT_ZICBOM),
    __SBI_HART_EXT_DATA(svpbmt, SBI_HART_EXT_SVPBMT),
    __SBI_HART_EXT_DATA(sdtrig, SBI_HART_EXT_SDTRIG),
    __SBI_HART_EXT_DATA(smcsrind, SBI_HART_EXT_SMCSRIND),
    __SBI_HART_EXT_DATA(smcdeleg, SBI_HART_EXT_SMCDELEG),
    __SBI_HART_EXT_DATA(sscsrind, SBI_HART_EXT_SSCSRIND),
    __SBI_HART_EXT_DATA(ssccfg, SBI_HART_EXT_SSCCFG),
    __SBI_HART_EXT_DATA(svade, SBI_HART_EXT_SVADE),
    __SBI_HART_EXT_DATA(svadu, SBI_HART_EXT_SVADU),
    __SBI_HART_EXT_DATA(smnpm, SBI_HART_EXT_SMNPM),
    __SBI_HART_EXT_DATA(zicfilp, SBI_HART_EXT_ZICFILP),
    __SBI_HART_EXT_DATA(zicfiss, SBI_HART_EXT_ZICFISS),
    __SBI_HART_EXT_DATA(ssdbltrp, SBI_HART_EXT_SSDBLTRP),
}
```

30.251 lib/sbi/sbi_heap.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_list.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_heap.c:

Macros

- #define HEAP_ALLOC_ALIGN 64
- #define HEAP_HOUSEKEEPING_FACTOR 16

Functions

- static void * alloc_with_align (struct sbi_heap_control *hpctrl, size_t align, size_t size)
- void * sbi_malloc_from (struct sbi_heap_control *hpctrl, size_t size)
- void * sbi_aligned_alloc_from (struct sbi_heap_control *hpctrl, size_t alignment, size_t size)
- void * sbi_zalloc_from (struct sbi_heap_control *hpctrl, size_t size)
- void sbi_free_from (struct sbi_heap_control *hpctrl, void *ptr)
- unsigned long sbi_heap_free_space_from (struct sbi_heap_control *hpctrl)
- unsigned long sbi_heap_used_space_from (struct sbi_heap_control *hpctrl)
- unsigned long sbi_heap_reserved_space_from (struct sbi_heap_control *hpctrl)
- int sbi_heap_init_new (struct sbi_heap_control *hpctrl, unsigned long base, unsigned long size)
- int sbi_heap_init (struct sbi_scratch *scratch)
- int sbi_heap_alloc_new (struct sbi_heap_control **hpctrl)

Variables

- struct sbi_heap_control global_hpctrl

30.251.1 Macro Definition Documentation

30.251.1.1 HEAP_ALLOC_ALIGN

```
#define HEAP_ALLOC_ALIGN 64
```

30.251.1.2 HEAP_HOUSEKEEPING_FACTOR

```
#define HEAP_HOUSEKEEPING_FACTOR 16
```

30.251.2 Function Documentation

30.251.2.1 alloc_with_align()

```
static void * alloc_with_align (
            struct sbi_heap_control * hpctrl,
            size_t align,
            size_t size )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.251.2.2 sbi_aligned_alloc_from()

```
void * sbi_aligned_alloc_from (
            struct sbi_heap_control * hpctrl,
            size_t alignment,
            size_t size )
```

Allocate aligned from heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.251.2.3 sbi_free_from()

```
void sbi_free_from (
            struct sbi_heap_control * hpctrl,
            void * ptr )
```

Free-up to heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.251.2.4 sbi_heap_alloc_new()

```
int sbi_heap_alloc_new (
            struct sbi_heap_control ** hpctrl )
```

Here is the call graph for this function:

30.251.2.5 sbi_heap_free_space_from()

```
unsigned long sbi_heap_free_space_from (
            struct sbi_heap_control * hpctrl )
```

Amount (in bytes) of free space in the heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.251.2.6 sbi_heap_init()

```
int sbi_heap_init (
            struct sbi_scratch * scratch )
```

Initialize heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.251.2.7 sbi_heap_init_new()

```
int sbi_heap_init_new (
            struct sbi_heap_control * hpctrl,
            unsigned long base,
            unsigned long size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.251.2.8 sbi_heap_reserved_space_from()

```
unsigned long sbi_heap_reserved_space_from (
            struct sbi_heap_control * hpctrl )
```

Amount (in bytes) of reserved space in the heap area Here is the caller graph for this function:

30.251.2.9 sbi_heap_used_space_from()

```
unsigned long sbi_heap_used_space_from (
            struct sbi_heap_control * hpctrl )
```

Amount (in bytes) of used space in the heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.251.2.10 sbi_malloc_from()

```
void * sbi_malloc_from (
            struct sbi_heap_control * hpctrl,
            size_t size )
```

Allocate from heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.251.2.11 sbi_zalloc_from()

```
void * sbi_zalloc_from (
            struct sbi_heap_control * hpctrl,
            size_t size )
```

Zero allocate from heap area Here is the call graph for this function:

Here is the caller graph for this function:

30.251.3 Variable Documentation

30.251.3.1 global_hpctrl

```
struct sbi_heap_control global_hpctrl
```

30.252 lib/sbi/sbi_hsm.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_atomic.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_init.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_hsm.c:

Macros

- #define __sbi_hsm_hart_change_state(hdata, oldstate, newstate)

Functions

- bool sbi_hsm_hart_change_state (struct sbi_scratch ∗scratch, long oldstate, long newstate)
- int __sbi_hsm_hart_get_state (u32 hartindex)
- int sbi_hsm_hart_get_state (const struct sbi_domain ∗dom, u32 hartid)
- static bool hsm_start_ticket_acquire (struct sbi_hsm_data ∗hdata)
- static void hsm_start_ticket_release (struct sbi_hsm_data ∗hdata)
- int sbi_hsm_hart_interruptible_mask (const struct sbi_domain ∗dom, struct sbi_hartmask ∗mask)
- void __noreturn sbi_hsm_hart_start_finish (struct sbi_scratch ∗scratch, u32 hartid)
- static void sbi_hsm_hart_wait (struct sbi_scratch ∗scratch)
- const struct sbi_hsm_device ∗ sbi_hsm_get_device (void)
- void sbi_hsm_set_device (const struct sbi_hsm_device ∗dev)
- static bool hsm_device_has_hart_hotplug (void)
- static bool hsm_device_has_hart_secondary_boot (void)
- static int hsm_device_hart_start (u32 hartid, ulong saddr)
- static int hsm_device_hart_stop (void)
- static int hsm_device_hart_suspend (u32 suspend_type, ulong mmode_resume_addr)
- static void hsm_device_hart_resume (void)
- int sbi_hsm_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void __noreturn sbi_hsm_exit (struct sbi_scratch ∗scratch)
- int sbi_hsm_hart_start (struct sbi_scratch ∗scratch, const struct sbi_domain ∗dom, u32 hartid, ulong saddr, ulong smode, ulong arg1)
- int sbi_hsm_hart_stop (struct sbi_scratch ∗scratch, bool exitnow)
- static int __sbi_hsm_suspend_default (struct sbi_scratch ∗scratch)
- void __sbi_hsm_suspend_non_ret_save (struct sbi_scratch ∗scratch)
- static void __sbi_hsm_suspend_non_ret_restore (struct sbi_scratch ∗scratch)
- void sbi_hsm_hart_resume_start (struct sbi_scratch ∗scratch)
- void __noreturn sbi_hsm_hart_resume_finish (struct sbi_scratch ∗scratch, u32 hartid)
- int sbi_hsm_hart_suspend (struct sbi_scratch ∗scratch, u32 suspend_type, ulong raddr, ulong rmode, ulong arg1)

Variables

- static const struct sbi_hsm_device ∗ hsm_dev = NULL
- static unsigned long hart_data_offset

30.252.1 Macro Definition Documentation

30.252.1.1 __sbi_hsm_hart_change_state

```
#define __sbi_hsm_hart_change_state(
            hdata,
            oldstate,
            newstate )
```

Value:

```
({                                          \
    long state = atomic_cmpxchg(&(hdata)->state, oldstate, newstate); \
    if (state != (oldstate))                                          \
        sbi_printf("%s: ERR: The hart is in invalid state [%lu]\n", \
                __func__, state);                                     \
    state == (oldstate);                                              \
})
```

30.252.2 Function Documentation

30.252.2.1 __sbi_hsm_hart_get_state()

```
int __sbi_hsm_hart_get_state (
            u32 hartindex )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.2 __sbi_hsm_suspend_default()

```
static int __sbi_hsm_suspend_default (
            struct sbi_scratch * scratch )  [static]
```

Here is the caller graph for this function:

30.252.2.3 __sbi_hsm_suspend_non_ret_restore()

```
static void __sbi_hsm_suspend_non_ret_restore (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.4 __sbi_hsm_suspend_non_ret_save()

```
void __sbi_hsm_suspend_non_ret_save (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.5 hsm_device_hart_resume()

```
static void hsm_device_hart_resume (
            void  )  [static]
```

Here is the caller graph for this function:

30.252.2.6 hsm_device_hart_start()

```
static int hsm_device_hart_start (
            u32 hartid,
            ulong saddr )  [static]
```

Here is the caller graph for this function:

30.252.2.7 hsm_device_hart_stop()

```
static int hsm_device_hart_stop (
            void  )  [static]
```

Here is the caller graph for this function:

30.252.2.8 hsm_device_hart_suspend()

```
static int hsm_device_hart_suspend (
            u32 suspend_type,
            ulong mmode_resume_addr )  [static]
```

Here is the caller graph for this function:

30.252.2.9 hsm_device_has_hart_hotplug()

```
static bool hsm_device_has_hart_hotplug (
            void ) [static]
```

Here is the caller graph for this function:

30.252.2.10 hsm_device_has_hart_secondary_boot()

```
static bool hsm_device_has_hart_secondary_boot (
            void ) [static]
```

Here is the caller graph for this function:

30.252.2.11 hsm_start_ticket_acquire()

```
static bool hsm_start_ticket_acquire (
            struct sbi_hsm_data * hdata ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.12 hsm_start_ticket_release()

```
static void hsm_start_ticket_release (
            struct sbi_hsm_data * hdata )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.13 sbi_hsm_exit()

```
void __noreturn sbi_hsm_exit (
            struct sbi_scratch * scratch )
```

As platform is lacking support for hotplug, directly jump to warmboot and wait for interrupts in warmboot. We do it preemptively in order preserve the hart states and reuse the code path for hotplug.Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.14 sbi_hsm_get_device()

```
const struct sbi_hsm_device * sbi_hsm_get_device (
            void  )
```

Here is the caller graph for this function:

30.252.2.15 sbi_hsm_hart_change_state()

```
bool sbi_hsm_hart_change_state (
            struct sbi_scratch * scratch,
            long oldstate,
            long newstate )
```

Here is the caller graph for this function:

30.252.2.16 sbi_hsm_hart_get_state()

```
int sbi_hsm_hart_get_state (
            const struct sbi_domain * dom,
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.17 sbi_hsm_hart_interruptible_mask()

```
int sbi_hsm_hart_interruptible_mask (
            const struct sbi_domain * dom,
            struct sbi_hartmask * mask )
```

Get the mask of harts which are valid IPI targets

Parameters

dom	the domain to be used for output HART mask
mask	the output hartmask to fill

Returns

0 on success and SBI_Exxx (< 0) on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.18 sbi_hsm_hart_resume_finish()

```
void __noreturn sbi_hsm_hart_resume_finish (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.19 sbi_hsm_hart_resume_start()

```
void sbi_hsm_hart_resume_start (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.20 sbi_hsm_hart_start()

```
int sbi_hsm_hart_start (
            struct sbi_scratch * scratch,
            const struct sbi_domain * dom,
            u32 hartid,
            ulong saddr,
            ulong smode,
            ulong arg1 )
```

if a hart is already transition to start or stop, another start call is considered as invalid request.Here is the call graph

for this function:

Here is the caller graph for this function:

30.252.2.21 sbi_hsm_hart_start_finish()

```
void __noreturn sbi_hsm_hart_start_finish (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.22 sbi_hsm_hart_stop()

```
int sbi_hsm_hart_stop (
            struct sbi_scratch * scratch,
            bool exitnow )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.23 sbi_hsm_hart_suspend()

```
int sbi_hsm_hart_suspend (
            struct sbi_scratch * scratch,
            u32 suspend_type,
            ulong raddr,
```

```
            ulong rmode,
            ulong arg1 )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.24 sbi_hsm_hart_wait()

```
static void sbi_hsm_hart_wait (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.25 sbi_hsm_init()

```
int sbi_hsm_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.252.2.26 sbi_hsm_set_device()

```
void sbi_hsm_set_device (
            const struct sbi_hsm_device * dev )
```

Here is the caller graph for this function:

30.252.3 Variable Documentation

30.252.3.1 hart_data_offset

```
unsigned long hart_data_offset  [static]
```

30.252.3.2 hsm_dev

```
const struct sbi_hsm_device* hsm_dev = NULL  [static]
```

30.253 lib/sbi/sbi_illegal_insn.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_emulate_csr.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_illegal_insn.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for sbi_illegal_insn.c:

Typedefs

- typedef int(∗ illegal_insn_func) (ulong insn, struct sbi_trap_regs ∗regs)

Functions

- static int truly_illegal_insn (ulong insn, struct sbi_trap_regs ∗regs)
- static int misc_mem_opcode_insn (ulong insn, struct sbi_trap_regs ∗regs)
- static int system_opcode_insn (ulong insn, struct sbi_trap_regs ∗regs)
- int sbi_illegal_insn_handler (struct sbi_trap_context ∗tcntx)

Variables

- static const illegal_insn_func illegal_insn_table [32]

30.253.1 Typedef Documentation

30.253.1.1 illegal_insn_func

```
typedef int(* illegal_insn_func) (ulong insn, struct sbi_trap_regs *regs)
```

30.253.2 Function Documentation

30.253.2.1 misc_mem_opcode_insn()

```
static int misc_mem_opcode_insn (
            ulong insn,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

30.253.2.2 sbi_illegal_insn_handler()

```
int sbi_illegal_insn_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.253.2.3 system_opcode_insn()

```
static int system_opcode_insn (
            ulong insn,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

30.253.2.4 truly_illegal_insn()

```
static int truly_illegal_insn (
            ulong insn,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.253.3 Variable Documentation

30.253.3.1 illegal_insn_table

```
const illegal_insn_func illegal_insn_table[32]  [static]
```

30.254 lib/sbi/sbi_init.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_barrier.h>
```

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_cppc.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_fwft.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_irqchip.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_dbtr.h>
#include <sbi/sbi_mpxy.h>
#include <sbi/sbi_sse.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_tlb.h>
#include <sbi/sbi_version.h>
#include <sbi/sbi_unit_test.h>
```

Include dependency graph for sbi_init.c:

Macros

- #define BANNER

Functions

- static void sbi_boot_print_banner (struct sbi_scratch *scratch)
- static void sbi_boot_print_general (struct sbi_scratch *scratch)
- static void sbi_boot_print_domains (struct sbi_scratch *scratch)
- static void sbi_boot_print_hart (struct sbi_scratch *scratch, u32 hartid)
- static void wait_for_coldboot (struct sbi_scratch *scratch)
- static void wake_coldboot_harts (struct sbi_scratch *scratch)
- static void __noreturn init_coldboot (struct sbi_scratch *scratch, u32 hartid)
- static void __noreturn init_warm_startup (struct sbi_scratch *scratch, u32 hartid)
- static void __noreturn init_warm_resume (struct sbi_scratch *scratch, u32 hartid)
- static void __noreturn init_warmboot (struct sbi_scratch *scratch, u32 hartid)
- void __noreturn sbi_init (struct sbi_scratch *scratch)
- unsigned long sbi_entry_count (u32 hartindex)
- unsigned long sbi_init_count (u32 hartindex)
- void __noreturn sbi_exit (struct sbi_scratch *scratch)

Variables

- static unsigned long coldboot_done
- static unsigned long entry_count_offset
- static unsigned long init_count_offset
- static atomic_t coldboot_lottery = ATOMIC_INITIALIZER(0)

30.254.1 Macro Definition Documentation

30.254.1.1 BANNER

```
#define BANNER
```

Value:

```
"   ____                    _____ ____ _____\n"     \
"  / __ \\                  / ____|  _ \\_   _|\n"  \
" | |  | |_ __   ___ _ __ | (___ | |_) || |\n"      \
" | |  | | '_ \\ / _ \\ '_ \\ \\___ \\|  _ < | |\n" \
" | |__| | |_) |  __/ | | |____) | |_) || |_\n"     \
"  \\____/| .__/ \\___|_| |_|_____/|____/_____|\n"  \
"        | |\n"                                    \
"        |_|\n\n"
```

30.254.2 Function Documentation

30.254.2.1 init_coldboot()

```
static void __noreturn init_coldboot (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.2 init_warm_resume()

```
static void __noreturn init_warm_resume (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.3 init_warm_startup()

```
static void __noreturn init_warm_startup (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.4 init_warmboot()

```
static void __noreturn init_warmboot (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.5 sbi_boot_print_banner()

```
static void sbi_boot_print_banner (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.6 sbi_boot_print_domains()

```
static void sbi_boot_print_domains (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.7 sbi_boot_print_general()

```
static void sbi_boot_print_general (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.8 sbi_boot_print_hart()

```
static void sbi_boot_print_hart (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.9 sbi_entry_count()

```
unsigned long sbi_entry_count (
            u32 hartindex )
```

Here is the caller graph for this function:

30.254.2.10 sbi_exit()

```
void __noreturn sbi_exit (
            struct sbi_scratch * scratch )
```

Exit OpenSBI library for current HART and stop HART

The function expects following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. Stack pointer (SP) is setup for current HART

Parameters

scratch	pointer to sbi_scratch of current HART

Here is the call graph for this function:

Here is the caller graph for this function:

30.254.2.11 sbi_init()

```
void __noreturn sbi_init (
            struct sbi_scratch * scratch )
```

Initialize OpenSBI library for current HART and jump to next booting stage.
The function expects following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. Stack pointer (SP) is setup for current HART
3. Interrupts are disabled in MSTATUS CSR
4. All interrupts are disabled in MIE CSR

Parameters

scratch	pointer to sbi_scratch of current HART

Here is the call graph for this function:

30.254.2.12 sbi_init_count()

```
unsigned long sbi_init_count (
            u32 hartindex )
```

Here is the caller graph for this function:

30.254.2.13 wait_for_coldboot()

```
static void wait_for_coldboot (
            struct sbi_scratch * scratch )  [static]
```

Here is the caller graph for this function:

30.254.2.14 wake_coldboot_harts()

```
static void wake_coldboot_harts (
            struct sbi_scratch * scratch )  [static]
```

Here is the caller graph for this function:

30.254.3 Variable Documentation

30.254.3.1 coldboot_done

```
unsigned long coldboot_done  [static]
```

30.254.3.2 coldboot_lottery

```
atomic_t coldboot_lottery = ATOMIC_INITIALIZER(0)  [static]
```

30.254.3.3 entry_count_offset

```
unsigned long entry_count_offset  [static]
```

30.254.3.4 init_count_offset

```
unsigned long init_count_offset  [static]
```

30.255 lib/sbi/sbi_ipi.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_barrier.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
```

```
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_init.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_tlb.h>
```

Include dependency graph for sbi_ipi.c:

Functions

- static int sbi_ipi_send (struct sbi_scratch ∗scratch, u32 remote_hartindex, u32 event, void ∗data)
- static int sbi_ipi_sync (struct sbi_scratch ∗scratch, u32 event)
- int sbi_ipi_send_many (ulong hmask, ulong hbase, u32 event, void ∗data)
- int sbi_ipi_event_create (const struct sbi_ipi_event_ops ∗ops)
- void sbi_ipi_event_destroy (u32 event)
- static void sbi_ipi_process_smode (struct sbi_scratch ∗scratch)
- int sbi_ipi_send_smode (ulong hmask, ulong hbase)
- void sbi_ipi_clear_smode (void)
- static int sbi_ipi_update_halt (struct sbi_scratch ∗scratch, struct sbi_scratch ∗remote_scratch, u32 remote↩_hartindex, void ∗data)
- static void sbi_ipi_process_halt (struct sbi_scratch ∗scratch)
- int sbi_ipi_send_halt (ulong hmask, ulong hbase)
- void sbi_ipi_process (void)
- int sbi_ipi_raw_send (u32 hartindex)
- void sbi_ipi_raw_clear (void)
- const struct sbi_ipi_device ∗ sbi_ipi_get_device (void)
- void sbi_ipi_set_device (const struct sbi_ipi_device ∗dev)
- int sbi_ipi_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_ipi_exit (struct sbi_scratch ∗scratch)

Variables

- static unsigned long ipi_data_off
- static const struct sbi_ipi_device ∗ ipi_dev = NULL
- static const struct sbi_ipi_event_ops ∗ ipi_ops_array [SBI_IPI_EVENT_MAX]
- static struct sbi_ipi_event_ops ipi_smode_ops
- static u32 ipi_smode_event = SBI_IPI_EVENT_MAX
- static struct sbi_ipi_event_ops ipi_halt_ops
- static u32 ipi_halt_event = SBI_IPI_EVENT_MAX

30.255.1 Function Documentation

30.255.1.1 sbi_ipi_clear_smode()

```
void sbi_ipi_clear_smode (
            void  )
```

Here is the caller graph for this function:

30.255.1.2 sbi_ipi_event_create()

```
int sbi_ipi_event_create (
            const struct sbi_ipi_event_ops * ops )
```

Here is the caller graph for this function:

30.255.1.3 sbi_ipi_event_destroy()

```
void sbi_ipi_event_destroy (
            u32 event )
```

30.255.1.4 sbi_ipi_exit()

```
void sbi_ipi_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.255.1.5 sbi_ipi_get_device()

```
const struct sbi_ipi_device * sbi_ipi_get_device (
            void  )
```

Here is the caller graph for this function:

30.255.1.6 sbi_ipi_init()

```
int sbi_ipi_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.255.1.7 sbi_ipi_process()

```
void sbi_ipi_process (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.255.1.8 sbi_ipi_process_halt()

```
static void sbi_ipi_process_halt (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

30.255.1.9 sbi_ipi_process_smode()

```
static void sbi_ipi_process_smode (
            struct sbi_scratch * scratch )  [static]
```

30.255.1.10 sbi_ipi_raw_clear()

```
void sbi_ipi_raw_clear (
            void  )
```

Here is the caller graph for this function:

30.255.1.11 sbi_ipi_raw_send()

```
int sbi_ipi_raw_send (
            u32 hartindex )
```

Here is the caller graph for this function:

30.255.1.12 sbi_ipi_send()

```
static int sbi_ipi_send (
            struct sbi_scratch * scratch,
```

```
            u32 remote_hartindex,
            u32 event,
            void * data )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.255.1.13 sbi_ipi_send_halt()

```
int sbi_ipi_send_halt (
            ulong hmask,
            ulong hbase )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.255.1.14 sbi_ipi_send_many()

```
int sbi_ipi_send_many (
            ulong hmask,
            ulong hbase,
            u32 event,
            void * data )
```

As this this function only handlers scalar values of hart mask, it must be set to all online harts if the intention is to send IPIs to all the harts. If hmask is zero, no IPIs will be sent. Here is the call graph for this function:

Here is the caller graph for this function:

30.255.1.15 sbi_ipi_send_smode()

```
int sbi_ipi_send_smode (
            ulong hmask,
            ulong hbase )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.255.1.16 sbi_ipi_set_device()

```
void sbi_ipi_set_device (
            const struct sbi_ipi_device * dev )
```

Here is the caller graph for this function:

30.255.1.17 sbi_ipi_sync()

```
static int sbi_ipi_sync (
            struct sbi_scratch * scratch,
            u32 event )  [static]
```

Here is the caller graph for this function:

30.255.1.18 sbi_ipi_update_halt()

```
static int sbi_ipi_update_halt (
            struct sbi_scratch * scratch,
            struct sbi_scratch * remote_scratch,
            u32 remote_hartindex,
            void * data )  [static]
```

30.255.2 Variable Documentation

30.255.2.1 ipi_data_off

```
unsigned long ipi_data_off  [static]
```

30.255.2.2 ipi_dev

```
const struct sbi_ipi_device* ipi_dev = NULL  [static]
```

30.255.2.3 ipi_halt_event

```
u32 ipi_halt_event = SBI_IPI_EVENT_MAX  [static]
```

30.255.2.4 ipi_halt_ops

```
struct sbi_ipi_event_ops ipi_halt_ops  [static]
```

Initial value:

```
= {
    .name = "IPI_HALT",
    .update = sbi_ipi_update_halt,
    .process = sbi_ipi_process_halt,
}
```

30.255.2.5 ipi_ops_array

```
const struct sbi_ipi_event_ops* ipi_ops_array[SBI_IPI_EVENT_MAX]  [static]
```

30.255.2.6 ipi_smode_event

```
u32 ipi_smode_event = SBI_IPI_EVENT_MAX  [static]
```

30.255.2.7 ipi_smode_ops

```
struct sbi_ipi_event_ops ipi_smode_ops  [static]
```

Initial value:

```
= {
    .name = "IPI_SMODE",
    .process = sbi_ipi_process_smode,
}
```

30.256 lib/sbi/sbi_irqchip.c File Reference

```
#include <sbi/sbi_irqchip.h>
#include <sbi/sbi_list.h>
#include <sbi/sbi_platform.h>
```

Include dependency graph for sbi_irqchip.c:

Functions

- static SBI_LIST_HEAD (irqchip_list)
- static int default_irqfn (void)
- void sbi_irqchip_add_device (struct sbi_irqchip_device ∗dev)
- int sbi_irqchip_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_irqchip_exit (struct sbi_scratch ∗scratch)

Variables

- static int(∗ ext_irqfn)(void)

30.256.1 Function Documentation

30.256.1.1 default_irqfn()

```
static int default_irqfn (
            void ) [static]
```

Here is the caller graph for this function:

30.256.1.2 sbi_irqchip_add_device()

```
void sbi_irqchip_add_device (
            struct sbi_irqchip_device * dev )
```

Register an irqchip device to receive callbacks Here is the call graph for this function:

Here is the caller graph for this function:

30.256.1.3 sbi_irqchip_exit()

```
void sbi_irqchip_exit (
            struct sbi_scratch * scratch )
```

Exit interrupt controllers Here is the call graph for this function:

Here is the caller graph for this function:

30.256.1.4 sbi_irqchip_init()

```
int sbi_irqchip_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Initialize interrupt controllers Here is the call graph for this function:

Here is the caller graph for this function:

30.256.1.5 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            irqchip_list ) [static]
```

30.256.2 Variable Documentation

30.256.2.1 ext_irqfn

```
int(* ext_irqfn) (void) (
            void ) [static], [default]
```

30.257 lib/sbi/sbi_math.c File Reference

Functions

- unsigned long log2roundup (unsigned long x)

30.257.1 Function Documentation

30.257.1.1 log2roundup()

```
unsigned long log2roundup (
            unsigned long x )
```

Here is the caller graph for this function:

30.258 lib/sbi/sbi_mpxy.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_mpxy.h>
```

```
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_byteorder.h>
```

Include dependency graph for sbi_mpxy.c:

Macros

- #define INVALID_ADDR (-1U)
- #define ATTR_SIZE (4)
- #define CAP_MSI_POS 0
- #define CAP_MSI_MASK (1U << CAP_MSI_POS)
- #define CAP_SSE_POS 1
- #define CAP_SSE_MASK (1U << CAP_SSE_POS)
- #define CAP_EVENTSSTATE_POS 2
- #define CAP_EVENTSSTATE_MASK (1U << CAP_EVENTSSTATE_POS)
- #define CAP_GET_NOTIFICATIONS_POS 3
- #define CAP_GET_NOTIFICATIONS_MASK (1U << CAP_GET_NOTIFICATIONS_POS)
- #define CAP_SEND_MSG_WITHOUT_RESP_POS 4
- #define CAP_SEND_MSG_WITHOUT_RESP_MASK (1U << CAP_SEND_MSG_WITHOUT_RESP_POS)
- #define CAP_SEND_MSG_WITH_RESP_POS 5
- #define CAP_SEND_MSG_WITH_RESP_MASK (1U << CAP_SEND_MSG_WITH_RESP_POS)
- #define CAP_ENABLE(_c, _m) INSERT_FIELD(_c, _m, 1)
- #define CAP_DISABLE(_c, _m) INSERT_FIELD(_c, _m, 0)
- #define CAP_GET(_c, _m) EXTRACT_FIELD(_c, _m)
- #define assert_field_offset(field, attr_offset)

Functions

- static SBI_LIST_HEAD (mpxy_channel_list)
- static void sbi_mpxy_shmem_disable (struct mpxy_state *ms)
- static bool mpxy_shmem_enabled (struct mpxy_state *ms)
- static void * hart_shmem_base (struct mpxy_state *ms)
- assert_field_offset (msg_proto_id, SBI_MPXY_ATTR_MSG_PROT_ID)
- assert_field_offset (msg_proto_version, SBI_MPXY_ATTR_MSG_PROT_VER)
- assert_field_offset (msg_data_maxlen, SBI_MPXY_ATTR_MSG_MAX_LEN)
- assert_field_offset (msg_send_timeout, SBI_MPXY_ATTR_MSG_SEND_TIMEOUT)
- assert_field_offset (msg_completion_timeout, SBI_MPXY_ATTR_MSG_COMPLETION_TIMEOUT)
- assert_field_offset (capability, SBI_MPXY_ATTR_CHANNEL_CAPABILITY)
- assert_field_offset (sse_event_id, SBI_MPXY_ATTR_SSE_EVENT_ID)
- assert_field_offset (msi_control, SBI_MPXY_ATTR_MSI_CONTROL)
- assert_field_offset (msi_info.msi_addr_lo, SBI_MPXY_ATTR_MSI_ADDR_LO)
- assert_field_offset (msi_info.msi_addr_hi, SBI_MPXY_ATTR_MSI_ADDR_HI)

- assert_field_offset (msi_info.msi_data, SBI_MPXY_ATTR_MSI_DATA)
- assert_field_offset (eventsstate_ctrl, SBI_MPXY_ATTR_EVENTS_STATE_CONTROL)
- static bool mpxy_is_std_attr (u32 attr_id)
- static struct sbi_mpxy_channel ∗ mpxy_find_channel (u32 channel_id)
- static void mpxy_copy_std_attrs (u32 ∗outmem, u32 ∗inmem, u32 count)
- bool sbi_mpxy_channel_available (void)
- static void mpxy_std_attrs_init (struct sbi_mpxy_channel ∗channel)
- int sbi_mpxy_register_channel (struct sbi_mpxy_channel ∗channel)
- int sbi_mpxy_init (struct sbi_scratch ∗scratch)
- int sbi_mpxy_set_shmem (unsigned long shmem_size, unsigned long shmem_phys_lo, unsigned long shmem_phys_hi, unsigned long flags)
- int sbi_mpxy_get_channel_ids (u32 start_index)
- int sbi_mpxy_read_attrs (u32 channel_id, u32 base_attr_id, u32 attr_count)
- static int mpxy_check_write_std_attr (struct sbi_mpxy_channel ∗channel, u32 attr_id, u32 attr_val)
- static void mpxy_write_std_attr (struct sbi_mpxy_channel ∗channel, u32 attr_id, u32 attr_val)
- int sbi_mpxy_write_attrs (u32 channel_id, u32 base_attr_id, u32 attr_count)
- int sbi_mpxy_send_message (u32 channel_id, u8 msg_id, unsigned long msg_data_len, unsigned long ∗resp_data_len)
- int sbi_mpxy_get_notification_events (u32 channel_id, unsigned long ∗events_len)

Variables

- static unsigned long mpxy_state_offset

30.258.1 Macro Definition Documentation

30.258.1.1 assert_field_offset

```
#define assert_field_offset(
            field,
            attr_offset )
```

Value:

```
_Static_assert(                                 \
    ((offsetof(struct sbi_mpxy_channel_attrs, field)) /   \
     sizeof(u32)) == attr_offset,                 \
    "field " #field                             \
    " from struct sbi_mpxy_channel_attrs invalid offset, expected " #attr_offset)
```

Make sure all attributes are packed for direct memcpy in ATTR_READ

30.258.1.2 ATTR_SIZE

```
#define ATTR_SIZE (4)
```

MPXY Attribute size in bytes

30.258.1.3 CAP_DISABLE

```
#define CAP_DISABLE(
            _c,
            _m ) INSERT_FIELD(_c, _m, 0)
```

30.258.1.4 CAP_ENABLE

```
#define CAP_ENABLE(
            _c,
            _m ) INSERT_FIELD(_c, _m, 1)
```

Helpers to enable/disable channel capability bits _c: capability variable _m: capability mask

30.258.1.5 CAP_EVENTSSTATE_MASK

```
#define CAP_EVENTSSTATE_MASK (1U << CAP_EVENTSSTATE_POS)
```

30.258.1.6 CAP_EVENTSSTATE_POS

```
#define CAP_EVENTSSTATE_POS 2
```

Channel Capability - Events State

30.258.1.7 CAP_GET

```
#define CAP_GET(
            _c,
            _m ) EXTRACT_FIELD(_c, _m)
```

30.258.1.8 CAP_GET_NOTIFICATIONS_MASK

```
#define CAP_GET_NOTIFICATIONS_MASK (1U << CAP_GET_NOTIFICATIONS_POS)
```

30.258.1.9 CAP_GET_NOTIFICATIONS_POS

```
#define CAP_GET_NOTIFICATIONS_POS 3
```

Channel Capability - Get Notification function support

30.258.1.10 CAP_MSI_MASK

```
#define CAP_MSI_MASK (1U << CAP_MSI_POS)
```

30.258.1.11 CAP_MSI_POS

```
#define CAP_MSI_POS 0
```

Channel Capability - MSI

30.258.1.12 CAP_SEND_MSG_WITH_RESP_MASK

```
#define CAP_SEND_MSG_WITH_RESP_MASK (1U << CAP_SEND_MSG_WITH_RESP_POS)
```

30.258.1.13 CAP_SEND_MSG_WITH_RESP_POS

```
#define CAP_SEND_MSG_WITH_RESP_POS 5
```

Channel Capability - Send Message With Response function support

30.258.1.14 CAP_SEND_MSG_WITHOUT_RESP_MASK

```
#define CAP_SEND_MSG_WITHOUT_RESP_MASK (1U << CAP_SEND_MSG_WITHOUT_RESP_POS)
```

30.258.1.15 CAP_SEND_MSG_WITHOUT_RESP_POS

```
#define CAP_SEND_MSG_WITHOUT_RESP_POS 4
```

Channel Capability - Send Message Without Response function support

30.258.1.16 CAP_SSE_MASK

```
#define CAP_SSE_MASK (1U << CAP_SSE_POS)
```

30.258.1.17 CAP_SSE_POS

```
#define CAP_SSE_POS 1
```

Channel Capability - SSE

30.258.1.18 INVALID_ADDR

```
#define INVALID_ADDR (-1U)
```

Invalid Physical Address(all bits 1)

30.258.2 Function Documentation

30.258.2.1 assert_field_offset() [1/12]

```
assert_field_offset (
            capability ,
            SBI_MPXY_ATTR_CHANNEL_CAPABILITY  )
```

30.258.2.2 assert_field_offset() [2/12]

```
assert_field_offset (
            eventsstate_ctrl ,
            SBI_MPXY_ATTR_EVENTS_STATE_CONTROL  )
```

30.258.2.3 assert_field_offset() [3/12]

```
assert_field_offset (
            msg_completion_timeout ,
            SBI_MPXY_ATTR_MSG_COMPLETION_TIMEOUT  )
```

30.258.2.4 assert_field_offset() [4/12]

```
assert_field_offset (
            msg_data_maxlen ,
            SBI_MPXY_ATTR_MSG_MAX_LEN  )
```

30.258.2.5 assert_field_offset() [5/12]

```
assert_field_offset (
            msg_proto_id ,
            SBI_MPXY_ATTR_MSG_PROT_ID  )
```

30.258.2.6 assert_field_offset() [6/12]

```
assert_field_offset (
            msg_proto_version ,
            SBI_MPXY_ATTR_MSG_PROT_VER  )
```

30.258.2.7 assert_field_offset() [7/12]

```
assert_field_offset (
            msg_send_timeout ,
            SBI_MPXY_ATTR_MSG_SEND_TIMEOUT  )
```

30.258.2.8 assert_field_offset() [8/12]

```
assert_field_offset (
            msi_control ,
            SBI_MPXY_ATTR_MSI_CONTROL  )
```

30.258.2.9 assert_field_offset() [9/12]

```
assert_field_offset (
            msi_info.  msi_addr_hi,
            SBI_MPXY_ATTR_MSI_ADDR_HI  )
```

30.258.2.10 assert_field_offset() [10/12]

```
assert_field_offset (
            msi_info.  msi_addr_lo,
            SBI_MPXY_ATTR_MSI_ADDR_LO  )
```

30.258.2.11 assert_field_offset() [11/12]

```
assert_field_offset (
            msi_info.  msi_data,
            SBI_MPXY_ATTR_MSI_DATA  )
```

30.258.2.12 assert_field_offset() [12/12]

```
assert_field_offset (
            sse_event_id ,
            SBI_MPXY_ATTR_SSE_EVENT_ID  )
```

30.258.2.13 hart_shmem_base()

```
static void * hart_shmem_base (
            struct mpxy_state * ms )  [inline], [static]
```

Get hart shared memory base address Here is the caller graph for this function:

30.258.2.14 mpxy_check_write_std_attr()

```
static int mpxy_check_write_std_attr (
            struct sbi_mpxy_channel * channel,
            u32 attr_id,
            u32 attr_val )  [static]
```

Verify the channel standard attribute wrt to write permission and the value to be set if valid or not. Only attributes needs to be checked which are defined Read/Write permission. Other with Readonly permission will result in error. Attributes values to be written must also be checked because before writing a range of attributes, we need to make sure that either complete range of attributes is written successfully or not at all. All RO access attributes falls under defaultHere is the caller graph for this function:

30.258.2.15 mpxy_copy_std_attrs()

```
static void mpxy_copy_std_attrs (
            u32 * outmem,
```

```
            u32 * inmem,
            u32 count )  [static]
```

Copy attributes word size Here is the caller graph for this function:

30.258.2.16 mpxy_find_channel()

```
static struct sbi_mpxy_channel * mpxy_find_channel (
            u32 channel_id )  [static]
```

Find channel_id in registered channels list Here is the caller graph for this function:

30.258.2.17 mpxy_is_std_attr()

```
static bool mpxy_is_std_attr (
            u32 attr_id )  [inline], [static]
```

Check if the attribute is a standard attribute or a message protocol specific attribute attr_id[31] = 0 for standard attr_id[31] = 1 for message protocol specific Here is the caller graph for this function:

30.258.2.18 mpxy_shmem_enabled()

```
static bool mpxy_shmem_enabled (
            struct mpxy_state * ms )  [inline], [static]
```

Check if shared memory is already setup on hart Here is the caller graph for this function:

30.258.2.19 mpxy_std_attrs_init()

```
static void mpxy_std_attrs_init (
            struct sbi_mpxy_channel * channel )  [static]
```

Check if MSI or SSE available for notification interrrupt. Priority given to MSI if both MSI and SSE are avaialble. switch_eventstate callback support means support for events state reporting supoprt. Enable events state reporting in channel capability.Here is the caller graph for this function:

30.258.2.20 mpxy_write_std_attr()

```
static void mpxy_write_std_attr (
            struct sbi_mpxy_channel * channel,
            u32 attr_id,
            u32 attr_val )  [static]
```

Write the attribute value Here is the caller graph for this function:

30.258.2.21 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            mpxy_channel_list  )  [static]
```

List of MPXY proxy channels

30.258.2.22 sbi_mpxy_channel_available()

```
bool sbi_mpxy_channel_available (
            void  )
```

Check if any channel is registered with mpxy framework Here is the call graph for this function:

Here is the caller graph for this function:

30.258.2.23 sbi_mpxy_get_channel_ids()

```
int sbi_mpxy_get_channel_ids (
            u32 start_index )
```

Get channel IDs list number of channel ids which can be stored in shmem adjusting for remaining and returned fieldsHere is the call graph for this function:

Here is the caller graph for this function:

30.258.2.24 sbi_mpxy_get_notification_events()

```
int sbi_mpxy_get_notification_events (
```

```
u32 channel_id,
unsigned long * events_len )
```

Get Message proxy notification events Here is the call graph for this function:

Here is the caller graph for this function:

30.258.2.25 sbi_mpxy_init()

```
int sbi_mpxy_init (
            struct sbi_scratch * scratch )
```

Initialize Message proxy subsystem TODO: Proper support for checking msi support from platform. Currently disable msi and sse and use pollingHere is the call graph for this function:

Here is the caller graph for this function:

30.258.2.26 sbi_mpxy_read_attrs()

```
int sbi_mpxy_read_attrs (
            u32 channel_id,
            u32 base_attr_id,
            u32 attr_count )
```

Read MPXY channel attributes Even if the message protocol driver does not provide read attribute callback, return bad range error instead of not supported to let client distinguish it from channel id not supported. Check the complate range supported for message protocol attributes. Actual supported attributes will be checked by the message protocol driver.

Function expected to return the SBI supported errors At this point both base attribute id and only the mpxy supported range been verified. Platform callback must check if the range requested is supported by message protocol driverHere is the call graph for this function:

Here is the caller graph for this function:

30.258.2.27 sbi_mpxy_register_channel()

```
int sbi_mpxy_register_channel (
            struct sbi_mpxy_channel * channel )
```

Register a channel with MPXY framework. Called by message protocol drivers Here is the call graph for this function:

Here is the caller graph for this function:

30.258.2.28 sbi_mpxy_send_message()

```
int sbi_mpxy_send_message (
            u32 channel_id,
            u8 msg_id,
            unsigned long msg_data_len,
            unsigned long * resp_data_len )
```

Send a message over a MPXY channel. In case if response is not expected, resp_data_len will be NULL. Here is the call graph for this function:

Here is the caller graph for this function:

30.258.2.29 sbi_mpxy_set_shmem()

```
int sbi_mpxy_set_shmem (
            unsigned long shmem_size,
            unsigned long shmem_phys_lo,
            unsigned long shmem_phys_hi,
            unsigned long flags )
```

Set Message proxy shared memory on the calling HART Disable shared memory if both hi and lo have all bit 1s
Check shared memory size and address aligned to 4K Page
Save the current shmem details in new shmem region

Setup the new shared memoryHere is the call graph for this function:

Here is the caller graph for this function:

30.258.2.30 sbi_mpxy_shmem_disable()

```
static void sbi_mpxy_shmem_disable (
            struct mpxy_state * ms )  [inline], [static]
```

Disable hart shared memory Here is the caller graph for this function:

30.258.2.31 sbi_mpxy_write_attrs()

```
int sbi_mpxy_write_attrs (
            u32 channel_id,
            u32 base_attr_id,
            u32 attr_count )
```

Write MPXY channel attributes Verify the attribute ids range and values

Message protocol specific attributes: If attributes belong to message protocol, they are simply passed to the message protocol driver callback after checking the valid range. Attributes contiguous range & permission & other checks are done by the mpxy and message protocol glue layer.

Even if the message protocol driver does not provide write attribute callback, return bad range error instead of not supported to let client distinguish it from channel id not supported.

Function expected to return the SBI supported errors At this point both base attribute id and only the mpxy supported range been verified. Platform callback must check if the range requested is supported by message protocol

driverHere is the call graph for this function:

Here is the caller graph for this function:

30.258.3 Variable Documentation

30.258.3.1 mpxy_state_offset

`unsigned long mpxy_state_offset [static]`

Offset of pointer to MPXY state in scratch space

30.259 lib/sbi/sbi_platform.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_platform.c:

Functions

- static char * sbi_platform_feature_id2string (unsigned long feature)
- void sbi_platform_get_features_str (const struct sbi_platform *plat, char *features_str, int nfstr)

30.259.1 Function Documentation

30.259.1.1 sbi_platform_feature_id2string()

```
static char * sbi_platform_feature_id2string (
            unsigned long feature )  [inline], [static]
```

Here is the caller graph for this function:

30.259.1.2 sbi_platform_get_features_str()

```
void sbi_platform_get_features_str (
            const struct sbi_platform * plat,
            char * features_str,
            int nfstr )
```

Get the platform features in string format

Parameters

plat	pointer to struct sbi_platform
features_str	pointer to a char array where the features string will be updated
nfstr	length of the features_str. The feature string will be truncated if nfstr is not long enough.

Here is the call graph for this function:

Here is the caller graph for this function:

30.260 lib/sbi/sbi_pmu.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_sse.h>
```

Include dependency graph for sbi_pmu.c:

Macros

- #define pmu_get_hart_state_ptr(__scratch) phs_ptr_offset ? sbi_scratch_read_type((__scratch), void *, phs_ptr_offset) : NULL
- #define pmu_thishart_state_ptr() pmu_get_hart_state_ptr(sbi_scratch_thishart_ptr())
- #define pmu_set_hart_state_ptr(__scratch, __phs) sbi_scratch_write_type((__scratch), void *, phs_ptr_offset, (__phs))
- #define get_cidx_type(x) (((x) & SBI_PMU_EVENT_IDX_TYPE_MASK) >> SBI_PMU_EVENT_IDX_TYPE_OFFSET)
- #define get_cidx_code(x) (x & SBI_PMU_EVENT_IDX_CODE_MASK)

Functions

- static bool pmu_event_range_overlap (struct sbi_pmu_hw_event *evtA, struct sbi_pmu_hw_event *evtB)
- static bool pmu_event_select_overlap (struct sbi_pmu_hw_event *evt, uint64_t select_val, uint64_t select↩ _mask)
- static int pmu_event_validate (struct sbi_pmu_hart_state *phs, unsigned long event_idx, uint64_t edata)
- static int pmu_ctr_validate (struct sbi_pmu_hart_state *phs, uint32_t cidx, uint32_t *event_idx_code)
- int sbi_pmu_ctr_fw_read (uint32_t cidx, uint64_t *cval)
- static int pmu_add_hw_event_map (u32 eidx_start, u32 eidx_end, u32 cmap, uint64_t select, uint64_↩ t select_mask)
- int sbi_pmu_add_hw_event_counter_map (u32 eidx_start, u32 eidx_end, u32 cmap)
- int sbi_pmu_add_raw_event_counter_map (uint64_t select, uint64_t select_mask, u32 cmap)
- void sbi_pmu_ovf_irq ()
- static int pmu_ctr_enable_irq_hw (int ctr_idx)
- static void pmu_ctr_write_hw (uint32_t cidx, uint64_t ival)
- static int pmu_ctr_start_hw (uint32_t cidx, uint64_t ival, bool ival_update)
- int sbi_pmu_irq_bit (void)
- static int pmu_ctr_start_fw (struct sbi_pmu_hart_state *phs, uint32_t cidx, uint32_t event_code, uint64_↩ t event_data, uint64_t ival, bool ival_update)
- int sbi_pmu_ctr_start (unsigned long cbase, unsigned long cmask, unsigned long flags, uint64_t ival)
- static int pmu_ctr_stop_hw (uint32_t cidx)
- static int pmu_ctr_stop_fw (struct sbi_pmu_hart_state *phs, uint32_t cidx, uint32_t event_code)
- static int pmu_reset_hw_mhpmevent (int ctr_idx)
- int sbi_pmu_ctr_stop (unsigned long cbase, unsigned long cmask, unsigned long flag)
- static void pmu_update_inhibit_flags (unsigned long flags, uint64_t *mhpmevent_val)
- static int pmu_update_hw_mhpmevent (struct sbi_pmu_hw_event *hw_evt, int ctr_idx, unsigned long flags, unsigned long eindex, uint64_t data)
- static int pmu_fixed_ctr_update_inhibit_bits (int fixed_ctr, unsigned long flags)
- static int pmu_ctr_find_fixed_hw (unsigned long evt_idx_code)
- static int pmu_ctr_find_hw (struct sbi_pmu_hart_state *phs, unsigned long cbase, unsigned long cmask, unsigned long flags, unsigned long event_idx, uint64_t data)
- static int pmu_ctr_find_fw (struct sbi_pmu_hart_state *phs, unsigned long cbase, unsigned long cmask, uint32_t event_code, uint64_t edata)
- int sbi_pmu_ctr_cfg_match (unsigned long cidx_base, unsigned long cidx_mask, unsigned long flags, unsigned long event_idx, uint64_t event_data)
- int sbi_pmu_ctr_incr_fw (enum sbi_pmu_fw_event_code_id fw_id)
- unsigned long sbi_pmu_num_ctr (void)
- int sbi_pmu_ctr_get_info (uint32_t cidx, unsigned long *ctr_info)
- int sbi_pmu_event_get_info (unsigned long shmem_phys_lo, unsigned long shmem_phys_hi, unsigned long num_events, unsigned long flags)
- static void pmu_reset_event_map (struct sbi_pmu_hart_state *phs)
- const struct sbi_pmu_device * sbi_pmu_get_device (void)
- void sbi_pmu_set_device (const struct sbi_pmu_device *dev)
- void sbi_pmu_exit (struct sbi_scratch *scratch)
- static void pmu_sse_enable (uint32_t event_id)
- static void pmu_sse_disable (uint32_t event_id)
- static void pmu_sse_complete (uint32_t event_id)
- int sbi_pmu_init (struct sbi_scratch *scratch, bool cold_boot)

Variables

- static unsigned long phs_ptr_offset
- static const struct sbi_pmu_device * pmu_dev = NULL
- static struct sbi_pmu_hw_event * hw_event_map
- static uint32_t num_hw_events
- static uint32_t num_hw_ctrs
- static uint32_t total_ctrs
- static const struct sbi_sse_cb_ops pmu_sse_cb_ops

30.260.1 Macro Definition Documentation

30.260.1.1 get_cidx_code

```
#define get_cidx_code(
            x ) (x & SBI_PMU_EVENT_IDX_CODE_MASK)
```

30.260.1.2 get_cidx_type

```
#define get_cidx_type(
            x )  (((x) & SBI_PMU_EVENT_IDX_TYPE_MASK) >> SBI_PMU_EVENT_IDX_TYPE_OFFSET)
```

30.260.1.3 pmu_get_hart_state_ptr

```
#define pmu_get_hart_state_ptr(
            __scratch )  phs_ptr_offset ?  sbi_scratch_read_type((__scratch), void *, phs_ptr_offset)
:  NULL
```

30.260.1.4 pmu_set_hart_state_ptr

```
#define pmu_set_hart_state_ptr(
            __scratch,
            __phs )  sbi_scratch_write_type((__scratch), void *, phs_ptr_offset, (__phs))
```

30.260.1.5 pmu_thishart_state_ptr

```
#define pmu_thishart_state_ptr( )  pmu_get_hart_state_ptr(sbi_scratch_thishart_ptr())
```

30.260.2 Function Documentation

30.260.2.1 pmu_add_hw_event_map()

```
static int pmu_add_hw_event_map (
            u32 eidx_start,
            u32 eidx_end,
            u32 cmap,
            uint64_t select,
            uint64_t select_mask )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.2 pmu_ctr_enable_irq_hw()

```
static int pmu_ctr_enable_irq_hw (
            int ctr_idx )  [static]
```

Clear out the OF bit so that next interrupt can be enabled. This should be done only when the corresponding overflow interrupt bit is cleared. That indicates that software has already handled the previous interrupts or the hardware yet to set an overflow interrupt. Otherwise, there will be race conditions where we may clear the bit the software is yet to handle the interrupt.Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.3 pmu_ctr_find_fixed_hw()

```
static int pmu_ctr_find_fixed_hw (
            unsigned long evt_idx_code )  [static]
```

Here is the caller graph for this function:

30.260.2.4 pmu_ctr_find_fw()

```
static int pmu_ctr_find_fw (
            struct sbi_pmu_hart_state * phs,
            unsigned long cbase,
            unsigned long cmask,
            uint32_t event_code,
            uint64_t edata )  [static]
```

Any firmware counter can map to any firmware event. Thus, select the first available fw counter after sanity check. Here is the caller graph for this function:

30.260.2.5 pmu_ctr_find_hw()

```
static int pmu_ctr_find_hw (
            struct sbi_pmu_hart_state * phs,
            unsigned long cbase,
            unsigned long cmask,
            unsigned long flags,
            unsigned long event_idx,
            uint64_t data )  [static]
```

If Sscof is present try to find the programmable counter for cycle/instret as well.
Some of the platform may not support mcountinhibit. Checking the active_events is enough for them
We can't find any programmable counters for cycle/instret. Return the fixed counter as they are mandatory anyways.Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.6 pmu_ctr_start_fw()

```
static int pmu_ctr_start_fw (
            struct sbi_pmu_hart_state * phs,
            uint32_t cidx,
            uint32_t event_code,
            uint64_t event_data,
            uint64_t ival,
            bool ival_update )  [static]
```

Here is the caller graph for this function:

30.260.2.7 pmu_ctr_start_hw()

```
static int pmu_ctr_start_hw (
            uint32_t cidx,
            uint64_t ival,
            bool ival_update )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.8 pmu_ctr_stop_fw()

```
static int pmu_ctr_stop_fw (
            struct sbi_pmu_hart_state * phs,
            uint32_t cidx,
            uint32_t event_code )  [static]
```

Here is the caller graph for this function:

30.260.2.9 pmu_ctr_stop_hw()

```
static int pmu_ctr_stop_hw (
            uint32_t cidx )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.10 pmu_ctr_validate()

```
static int pmu_ctr_validate (
            struct sbi_pmu_hart_state * phs,
            uint32_t cidx,
            uint32_t * event_idx_code )  [static]
```

Here is the caller graph for this function:

30.260.2.11 pmu_ctr_write_hw()

```
static void pmu_ctr_write_hw (
            uint32_t cidx,
            uint64_t ival )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.12 pmu_event_range_overlap()

```
static bool pmu_event_range_overlap (
            struct sbi_pmu_hw_event * evtA,
            struct sbi_pmu_hw_event * evtB )  [static]
```

Perform a sanity check on event & counter mappings with event range overlap check

Parameters

evtA	Pointer to the existing hw event structure
evtB	Pointer to the new hw event structure

Return false if the range doesn't overlap, true otherwise Here is the caller graph for this function:

30.260.2.13 pmu_event_select_overlap()

```
static bool pmu_event_select_overlap (
            struct sbi_pmu_hw_event * evt,
            uint64_t select_val,
            uint64_t select_mask )  [static]
```

Here is the caller graph for this function:

30.260.2.14 pmu_event_validate()

```
static int pmu_event_validate (
            struct sbi_pmu_hart_state * phs,
            unsigned long event_idx,
            uint64_t edata )  [static]
```

Here is the caller graph for this function:

30.260.2.15 pmu_fixed_ctr_update_inhibit_bits()

```
static int pmu_fixed_ctr_update_inhibit_bits (
            int fixed_ctr,
            unsigned long flags )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.16 pmu_reset_event_map()

```
static void pmu_reset_event_map (
            struct sbi_pmu_hart_state * phs )  [static]
```

Here is the caller graph for this function:

30.260.2.17 pmu_reset_hw_mhpmevent()

```
static int pmu_reset_hw_mhpmevent (
            int ctr_idx ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.18 pmu_sse_complete()

```
static void pmu_sse_complete (
            uint32_t event_id ) [static]
```

30.260.2.19 pmu_sse_disable()

```
static void pmu_sse_disable (
            uint32_t event_id ) [static]
```

Here is the call graph for this function:

30.260.2.20 pmu_sse_enable()

```
static void pmu_sse_enable (
            uint32_t event_id )  [static]
```

Here is the call graph for this function:

30.260.2.21 pmu_update_hw_mhpmevent()

```
static int pmu_update_hw_mhpmevent (
            struct sbi_pmu_hw_event * hw_evt,
            int ctr_idx,
            unsigned long flags,
            unsigned long eindex,
            uint64_t data )  [static]
```

Always set the OVF bit(disable interrupts) and inhibit counting of events in M-mode. The OVF bit should be enabled during the start call.Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.22 pmu_update_inhibit_flags()

```
static void pmu_update_inhibit_flags (
            unsigned long flags,
            uint64_t * mhpmevent_val )  [static]
```

Here is the caller graph for this function:

30.260.2.23 sbi_pmu_add_hw_event_counter_map()

```
int sbi_pmu_add_hw_event_counter_map (
            u32 eidx_start,
            u32 eidx_end,
            u32 cmap )
```

Logical counter ids are assigned to hardware counters are assigned consecutively. E.g. counter0 must count MCYCLE where counter2 must count minstret. Similarly, counterX will mhpmcounterX. Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.24 sbi_pmu_add_raw_event_counter_map()

```
int sbi_pmu_add_raw_event_counter_map (
            uint64_t select,
            uint64_t select_mask,
            u32 cmap )
```

Add the raw hardware event selector and supported counter information. This should be called from the platform code to update the mapping table.

Parameters

info	a pointer to the hardware event info

Returns

0 on success, error otherwise.

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.25 sbi_pmu_ctr_cfg_match()

```
int sbi_pmu_ctr_cfg_match (
            unsigned long cidx_base,
            unsigned long cidx_mask,
            unsigned long flags,
            unsigned long event_idx,
            uint64_t event_data )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.26 sbi_pmu_ctr_fw_read()

```
int sbi_pmu_ctr_fw_read (
            uint32_t cidx,
            uint64_t * cval )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.27 sbi_pmu_ctr_get_info()

```
int sbi_pmu_ctr_get_info (
            uint32_t cidx,
            unsigned long * ctr_info )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.28 sbi_pmu_ctr_incr_fw()

```
int sbi_pmu_ctr_incr_fw (
            enum sbi_pmu_fw_event_code_id fw_id )
```

Here is the caller graph for this function:

30.260.2.29 sbi_pmu_ctr_start()

```
int sbi_pmu_ctr_start (
            unsigned long cbase,
            unsigned long cmask,
            unsigned long flags,
            uint64_t ival )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.30 sbi_pmu_ctr_stop()

```
int sbi_pmu_ctr_stop (
            unsigned long cbase,
```

```
                unsigned long cmask,
                unsigned long flag )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.31 sbi_pmu_event_get_info()

```
int sbi_pmu_event_get_info (
                unsigned long shmem_phys_lo,
                unsigned long shmem_phys_hi,
                unsigned long num_events,
                unsigned long flags )
```

Check shared memory size and address aligned to 16 byteHere is the call graph for this function:

Here is the caller graph for this function:

30.260.2.32 sbi_pmu_exit()

```
void sbi_pmu_exit (
                struct sbi_scratch * scratch )
```

Reset PMU during hart exit Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.33 sbi_pmu_get_device()

```
const struct sbi_pmu_device * sbi_pmu_get_device (
            void  )
```

Get the PMU platform device Here is the caller graph for this function:

30.260.2.34 sbi_pmu_init()

```
int sbi_pmu_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Initialize PMU Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.35 sbi_pmu_irq_bit()

```
int sbi_pmu_irq_bit (
            void  )
```

Return the pmu irq bit depending on extension existence Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.36 sbi_pmu_num_ctr()

```
unsigned long sbi_pmu_num_ctr (
            void  )
```

Here is the caller graph for this function:

30.260.2.37 sbi_pmu_ovf_irq()

```
void sbi_pmu_ovf_irq ( )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.260.2.38 sbi_pmu_set_device()

```
void sbi_pmu_set_device (
            const struct sbi_pmu_device * dev )
```

Set the PMU platform device

30.260.3 Variable Documentation

30.260.3.1 hw_event_map

```
struct sbi_pmu_hw_event* hw_event_map  [static]
```

30.260.3.2 num_hw_ctrs

```
uint32_t num_hw_ctrs  [static]
```

30.260.3.3 num_hw_events

```
uint32_t num_hw_events [static]
```

30.260.3.4 phs_ptr_offset

```
unsigned long phs_ptr_offset [static]
```

Offset of pointer to PMU HART state in scratch space

30.260.3.5 pmu_dev

```
const struct sbi_pmu_device* pmu_dev = NULL [static]
```

30.260.3.6 pmu_sse_cb_ops

```
const struct sbi_sse_cb_ops pmu_sse_cb_ops [static]
```

Initial value:

```
= {
    .enable_cb = pmu_sse_enable,
    .disable_cb = pmu_sse_disable,
    .complete_cb = pmu_sse_complete,
}
```

30.260.3.7 total_ctrs

```
uint32_t total_ctrs [static]
```

30.261 lib/sbi/sbi_scratch.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_scratch.c:

Typedefs

- typedef struct sbi_scratch *(* hartid2scratch) (ulong hartid, ulong hartindex)

Functions

- u32 sbi_hartid_to_hartindex (u32 hartid)
- int sbi_scratch_init (struct sbi_scratch *scratch)
- unsigned long sbi_scratch_alloc_offset (unsigned long size)
- void sbi_scratch_free_offset (unsigned long offset)
- unsigned long sbi_scratch_used_space (void)

Variables

- u32 last_hartindex_having_scratch = 0
- u32 hartindex_to_hartid_table [SBI_HARTMASK_MAX_BITS+1] = { -1U }
- struct sbi_scratch * hartindex_to_scratch_table [SBI_HARTMASK_MAX_BITS+1] = { 0 }
- static spinlock_t extra_lock = SPIN_LOCK_INITIALIZER
- static unsigned long extra_offset = SBI_SCRATCH_EXTRA_SPACE_OFFSET

30.261.1 Typedef Documentation

30.261.1.1 hartid2scratch

```
typedef struct sbi_scratch *(* hartid2scratch) (ulong hartid, ulong hartindex)
```

30.261.2 Function Documentation

30.261.2.1 sbi_hartid_to_hartindex()

```
u32 sbi_hartid_to_hartindex (
            u32 hartid )
```

Get logical index for given HART id

Parameters

hartid	physical HART id

Returns

value between 0 to SBI_HARTMASK_MAX_BITS upon success and SBI_HARTMASK_MAX_BITS upon failure.

Here is the caller graph for this function:

30.261.2.2 sbi_scratch_alloc_offset()

```
unsigned long sbi_scratch_alloc_offset (
            unsigned long size )
```

Allocate from extra space in sbi_scratch

Returns

zero on failure and non-zero (>= SBI_SCRATCH_EXTRA_SPACE_OFFSET) on success

Here is the call graph for this function:

Here is the caller graph for this function:

30.261.2.3 sbi_scratch_free_offset()

```
void sbi_scratch_free_offset (
            unsigned long offset )
```

Free-up extra space in sbi_scratch Here is the caller graph for this function:

30.261.2.4 sbi_scratch_init()

```
int sbi_scratch_init (
            struct sbi_scratch * scratch )
```

Initialize scratch table and allocator Here is the caller graph for this function:

30.261.2.5 sbi_scratch_used_space()

```
unsigned long sbi_scratch_used_space (
            void )
```

Amount (in bytes) of used space in in sbi_scratch Here is the call graph for this function:

Here is the caller graph for this function:

30.261.3 Variable Documentation

30.261.3.1 extra_lock

```
spinlock_t extra_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.261.3.2 extra_offset

```
unsigned long extra_offset = SBI_SCRATCH_EXTRA_SPACE_OFFSET  [static]
```

30.261.3.3 hartindex_to_hartid_table

```
u32 hartindex_to_hartid_table[SBI_HARTMASK_MAX_BITS+1] = { -1U }
```

HART index to HART id table

30.261.3.4 hartindex_to_scratch_table

```
struct sbi_scratch* hartindex_to_scratch_table[SBI_HARTMASK_MAX_BITS+1] = { 0 }
```

HART index to scratch table

30.261.3.5 last_hartindex_having_scratch

```
u32 last_hartindex_having_scratch = 0
```

Last HART index having a sbi_scratch pointer

30.262 lib/sbi/sbi_sse.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_locks.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_fifo.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_list.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_sse.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for sbi_sse.c:

Macros

- #define sse_get_hart_state_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, shs_ptr_off)
- #define sse_thishart_state_ptr() sse_get_hart_state_ptr(sbi_scratch_thishart_ptr())
- #define sse_set_hart_state_ptr(__scratch, __sse_state) sbi_scratch_write_type((__scratch), void *, shs_ptr_off, (__sse_state))
- #define EVENT_IS_GLOBAL(__event_id) ((__event_id) & SBI_SSE_EVENT_GLOBAL_BIT)
- #define EVENT_COUNT array_size(supported_events)
- #define sse_event_invoke_cb(_event, _cb, ...)
- #define assert_field_offset(field, attr_offset)

Functions

- assert_field_offset (status, SBI_SSE_ATTR_STATUS)
- assert_field_offset (prio, SBI_SSE_ATTR_PRIO)
- assert_field_offset (config, SBI_SSE_ATTR_CONFIG)
- assert_field_offset (hartid, SBI_SSE_ATTR_PREFERRED_HART)
- assert_field_offset (entry.pc, SBI_SSE_ATTR_ENTRY_PC)
- assert_field_offset (entry.arg, SBI_SSE_ATTR_ENTRY_ARG)
- assert_field_offset (interrupted.sepc, SBI_SSE_ATTR_INTERRUPTED_SEPC)
- assert_field_offset (interrupted.flags, SBI_SSE_ATTR_INTERRUPTED_FLAGS)
- assert_field_offset (interrupted.a6, SBI_SSE_ATTR_INTERRUPTED_A6)
- assert_field_offset (interrupted.a7, SBI_SSE_ATTR_INTERRUPTED_A7)
- static int sse_ipi_inject_send (unsigned long hartid, uint32_t event_id)
- static unsigned long sse_event_state (struct sbi_sse_event *e)
- static unsigned long sse_event_pending (struct sbi_sse_event *e)
- static bool sse_event_is_global (struct sbi_sse_event *e)
- static bool sse_event_is_local (struct sbi_sse_event *e)
- static struct sse_hart_state * sse_get_hart_state (struct sbi_sse_event *e)
- static struct sse_global_event * sse_get_global_event (struct sbi_sse_event *e)
- static void sse_enabled_event_lock (struct sbi_sse_event *e)
- static void sse_enabled_event_unlock (struct sbi_sse_event *e)
- static void sse_event_set_state (struct sbi_sse_event *e, unsigned long new_state)
- static struct sbi_sse_event * sse_event_get (uint32_t event_id)
- static void sse_event_put (struct sbi_sse_event *e)
- static void sse_event_remove_from_list (struct sbi_sse_event *e)
- static void sse_event_add_to_list (struct sbi_sse_event *e)
- static int sse_event_disable (struct sbi_sse_event *e)
- static int sse_event_set_hart_id_check (struct sbi_sse_event *e, unsigned long new_hartid)
- static int sse_event_set_attr_check (struct sbi_sse_event *e, uint32_t attr_id, unsigned long val)
- static void sse_event_set_attr (struct sbi_sse_event *e, uint32_t attr_id, unsigned long val)
- static int sse_event_register (struct sbi_sse_event *e, unsigned long handler_entry_pc, unsigned long handler_entry_arg)
- static int sse_event_unregister (struct sbi_sse_event *e)
- static unsigned long sse_interrupted_flags (unsigned long mstatus)
- static void sse_event_inject (struct sbi_sse_event *e, struct sbi_trap_regs *regs)
- static void sse_event_resume (struct sbi_sse_event *e, struct sbi_trap_regs *regs)
- static bool sse_event_is_ready (struct sbi_sse_event *e)
- static bool sse_event_check_inject (struct sbi_sse_event *e, struct sbi_trap_regs *regs)
- void sbi_sse_process_pending_events (struct sbi_trap_regs *regs)
- static int sse_event_set_pending (struct sbi_sse_event *e)
- static void sse_ipi_inject_process (struct sbi_scratch *scratch)
- static int sse_inject_event (uint32_t event_id, unsigned long hartid)
- static int sse_event_enable (struct sbi_sse_event *e)
- static int sse_event_complete (struct sbi_sse_event *e, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- int sbi_sse_complete (struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- int sbi_sse_enable (uint32_t event_id)
- int sbi_sse_disable (uint32_t event_id)
- int sbi_sse_hart_mask (void)
- int sbi_sse_hart_unmask (void)
- int sbi_sse_inject_from_ecall (uint32_t event_id, unsigned long hartid, struct sbi_ecall_return *out)
- int sbi_sse_inject_event (uint32_t event_id)
- int sbi_sse_set_cb_ops (uint32_t event_id, const struct sbi_sse_cb_ops *cb_ops)
- int sbi_sse_attr_check (uint32_t base_attr_id, uint32_t attr_count, unsigned long phys_lo, unsigned long phys_hi, unsigned long access)

- static void copy_attrs (unsigned long *out, const unsigned long *in, unsigned int long_count)
- int sbi_sse_read_attrs (uint32_t event_id, uint32_t base_attr_id, uint32_t attr_count, unsigned long output↩ _phys_lo, unsigned long output_phys_hi)
- static int sse_write_attrs (struct sbi_sse_event *e, uint32_t base_attr_id, uint32_t attr_count, unsigned long input_phys)
- int sbi_sse_write_attrs (uint32_t event_id, uint32_t base_attr_id, uint32_t attr_count, unsigned long input_↩ phys_lo, unsigned long input_phys_hi)
- int sbi_sse_register (uint32_t event_id, unsigned long handler_entry_pc, unsigned long handler_entry_arg)
- int sbi_sse_unregister (uint32_t event_id)
- static void sse_event_init (struct sbi_sse_event *e, uint32_t event_id)
- static void sse_event_count_init ()
- static int sse_global_init ()
- static void sse_local_init (struct sse_hart_state *shs)
- int sbi_sse_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_sse_exit (struct sbi_scratch *scratch)

Variables

- static const uint32_t supported_events [ ]
- static unsigned int local_event_count
- static unsigned int global_event_count
- static struct sse_global_event * global_events
- static unsigned long sse_inject_fifo_off
- static unsigned long sse_inject_fifo_mem_off
- static unsigned long shs_ptr_off
- static u32 sse_ipi_inject_event = SBI_IPI_EVENT_MAX
- static struct sbi_ipi_event_ops sse_ipi_inject_ops

30.262.1 Macro Definition Documentation

30.262.1.1 assert_field_offset

```
#define assert_field_offset(
            field,
            attr_offset )
```

Value:

```
    _Static_assert(                                          \
        ((offsetof(struct sbi_sse_event_attrs, field)) / \
         sizeof(unsigned long)) == attr_offset,          \
        "field " #field                                  \
        " from struct sbi_sse_event_attrs invalid offset, expected " #attr_offset)
```

30.262.1.2 EVENT_COUNT

```
#define EVENT_COUNT array_size(supported_events)
```

30.262.1.3 EVENT_IS_GLOBAL

```
#define EVENT_IS_GLOBAL(
            __event_id ) ((__event_id) & SBI_SSE_EVENT_GLOBAL_BIT)
```

30.262.1.4 sse_event_invoke_cb

```
#define sse_event_invoke_cb(
            _event,
            _cb,
            ...  )
```

Value:

```
    {                                                                    \
        if (_event->cb_ops && _event->cb_ops->_cb)                   \
            _event->cb_ops->_cb(_event->event_id, ##__VA_ARGS__); \
    }
```

30.262.1.5 sse_get_hart_state_ptr

```
#define sse_get_hart_state_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, shs_ptr_off)
```

30.262.1.6 sse_set_hart_state_ptr

```
#define sse_set_hart_state_ptr(
            __scratch,
            __sse_state )  sbi_scratch_write_type((__scratch), void *, shs_ptr_off, (__sse_↩
state))
```

30.262.1.7 sse_thishart_state_ptr

```
#define sse_thishart_state_ptr( )  sse_get_hart_state_ptr(sbi_scratch_thishart_ptr())
```

30.262.2 Function Documentation

30.262.2.1 assert_field_offset() [1/10]

```
assert_field_offset (
            config ,
            SBI_SSE_ATTR_CONFIG  )
```

30.262.2.2 assert_field_offset() [2/10]

```
assert_field_offset (
            entry.  arg,
            SBI_SSE_ATTR_ENTRY_ARG  )
```

30.262.2.3 assert_field_offset() [3/10]

```
assert_field_offset (
            entry.  pc,
            SBI_SSE_ATTR_ENTRY_PC  )
```

30.262.2.4 assert_field_offset() [4/10]

```
assert_field_offset (
            hartid ,
            SBI_SSE_ATTR_PREFERRED_HART  )
```

30.262.2.5 assert_field_offset() [5/10]

```
assert_field_offset (
            interrupted.  a6,
            SBI_SSE_ATTR_INTERRUPTED_A6  )
```

30.262.2.6 assert_field_offset() [6/10]

```
assert_field_offset (
            interrupted.  a7,
            SBI_SSE_ATTR_INTERRUPTED_A7  )
```

30.262.2.7 assert_field_offset() [7/10]

```
assert_field_offset (
            interrupted.  flags,
            SBI_SSE_ATTR_INTERRUPTED_FLAGS  )
```

30.262.2.8 assert_field_offset() [8/10]

```
assert_field_offset (
            interrupted.  sepc,
            SBI_SSE_ATTR_INTERRUPTED_SEPC  )
```

30.262.2.9 assert_field_offset() [9/10]

```
assert_field_offset (
            prio ,
            SBI_SSE_ATTR_PRIO  )
```

30.262.2.10 assert_field_offset() [10/10]

```
assert_field_offset (
            status ,
            SBI_SSE_ATTR_STATUS  )
```

30.262.2.11 copy_attrs()

```
static void copy_attrs (
            unsigned long * out,
            const unsigned long * in,
            unsigned int long_count )  [static]
```

Here is the caller graph for this function:

30.262.2.12 sbi_sse_attr_check()

```
int sbi_sse_attr_check (
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long phys_lo,
            unsigned long phys_hi,
            unsigned long access )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.13 sbi_sse_complete()

```
int sbi_sse_complete (
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.14 sbi_sse_disable()

```
int sbi_sse_disable (
            uint32_t event_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.15 sbi_sse_enable()

```
int sbi_sse_enable (
            uint32_t event_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.16 sbi_sse_exit()

```
void sbi_sse_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.17 sbi_sse_hart_mask()

```
int sbi_sse_hart_mask (
            void )
```

Here is the caller graph for this function:

30.262.2.18 sbi_sse_hart_unmask()

```
int sbi_sse_hart_unmask (
            void  )
```

Here is the caller graph for this function:

30.262.2.19 sbi_sse_init()

```
int sbi_sse_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.20 sbi_sse_inject_event()

```
int sbi_sse_inject_event (
            uint32_t event_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.21 sbi_sse_inject_from_ecall()

```
int sbi_sse_inject_from_ecall (
            uint32_t event_id,
            unsigned long hartid,
            struct sbi_ecall_return * out )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.22 sbi_sse_process_pending_events()

```
void sbi_sse_process_pending_events (
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.23 sbi_sse_read_attrs()

```
int sbi_sse_read_attrs (
            uint32_t event_id,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long output_phys_lo,
            unsigned long output_phys_hi )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.24 sbi_sse_register()

```
int sbi_sse_register (
            uint32_t event_id,
            unsigned long handler_entry_pc,
            unsigned long handler_entry_arg )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.25 sbi_sse_set_cb_ops()

```
int sbi_sse_set_cb_ops (
            uint32_t event_id,
            const struct sbi_sse_cb_ops * cb_ops )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.26 sbi_sse_unregister()

```
int sbi_sse_unregister (
            uint32_t event_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.27 sbi_sse_write_attrs()

```
int sbi_sse_write_attrs (
            uint32_t event_id,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long input_phys_lo,
            unsigned long input_phys_hi )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.28 sse_enabled_event_lock()

```
static void sse_enabled_event_lock (
            struct sbi_sse_event * e )  [static]
```

If event is global, must be called under enabled event lock Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.29 sse_enabled_event_unlock()

```
static void sse_enabled_event_unlock (
            struct sbi_sse_event * e )  [static]
```

If event is global, must be called under enabled event lock Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.30 sse_event_add_to_list()

```
static void sse_event_add_to_list (
            struct sbi_sse_event * e )  [static]
```

Must be called under owner hart lock Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.31 sse_event_check_inject()

```
static bool sse_event_check_inject (
            struct sbi_sse_event * e,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.32 sse_event_complete()

```
static int sse_event_complete (
            struct sbi_sse_event * e,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.33 sse_event_count_init()

```
static void sse_event_count_init ( ) [static]
```

Here is the caller graph for this function:

30.262.2.34 sse_event_disable()

```
static int sse_event_disable (
            struct sbi_sse_event * e ) [static]
```

Must be called under owner hart lock Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.35 sse_event_enable()

```
static int sse_event_enable (
            struct sbi_sse_event * e ) [static]
```

Must be called under owner hart lock Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.36 sse_event_get()

```
static struct sbi_sse_event * sse_event_get (
            uint32_t event_id )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.37 sse_event_init()

```
static void sse_event_init (
            struct sbi_sse_event * e,
            uint32_t event_id )  [static]
```

Here is the caller graph for this function:

30.262.2.38 sse_event_inject()

```
static void sse_event_inject (
            struct sbi_sse_event * e,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.39 sse_event_is_global()

```
static bool sse_event_is_global (
            struct sbi_sse_event * e )  [static]
```

Here is the caller graph for this function:

30.262.2.40 sse_event_is_local()

```
static bool sse_event_is_local (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.41 sse_event_is_ready()

```
static bool sse_event_is_ready (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.42 sse_event_pending()

```
static unsigned long sse_event_pending (
            struct sbi_sse_event * e )  [static]
```

Here is the caller graph for this function:

30.262.2.43 sse_event_put()

```
static void sse_event_put (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.44 sse_event_register()

```
static int sse_event_register (
            struct sbi_sse_event * e,
            unsigned long handler_entry_pc,
            unsigned long handler_entry_arg )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.45 sse_event_remove_from_list()

```
static void sse_event_remove_from_list (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.46 sse_event_resume()

```
static void sse_event_resume (
            struct sbi_sse_event * e,
            struct sbi_trap_regs * regs )  [static]
```

Here is the caller graph for this function:

30.262.2.47 sse_event_set_attr()

```
static void sse_event_set_attr (
            struct sbi_sse_event * e,
            uint32_t attr_id,
            unsigned long val )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.48 sse_event_set_attr_check()

```
static int sse_event_set_attr_check (
            struct sbi_sse_event * e,
            uint32_t attr_id,
            unsigned long val )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.49 sse_event_set_hart_id_check()

```
static int sse_event_set_hart_id_check (
            struct sbi_sse_event * e,
            unsigned long new_hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.50 sse_event_set_pending()

```
static int sse_event_set_pending (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.51 sse_event_set_state()

```
static void sse_event_set_state (
            struct sbi_sse_event * e,
            unsigned long new_state )  [static]
```

Here is the caller graph for this function:

30.262.2.52 sse_event_state()

```
static unsigned long sse_event_state (
            struct sbi_sse_event * e )  [static]
```

Here is the caller graph for this function:

30.262.2.53 sse_event_unregister()

```
static int sse_event_unregister (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.54 sse_get_global_event()

```
static struct sse_global_event * sse_get_global_event (
            struct sbi_sse_event * e )  [static]
```

Here is the caller graph for this function:

30.262.2.55 sse_get_hart_state()

```
static struct sse_hart_state * sse_get_hart_state (
            struct sbi_sse_event * e )  [static]
```

If event is global, must be called under global event lock Here is the caller graph for this function:

30.262.2.56 sse_global_init()

```
static int sse_global_init ( ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.57 sse_inject_event()

```
static int sse_inject_event (
            uint32_t event_id,
            unsigned long hartid ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.58 sse_interrupted_flags()

```
static unsigned long sse_interrupted_flags (
            unsigned long mstatus )  [static]
```

Here is the caller graph for this function:

30.262.2.59 sse_ipi_inject_process()

```
static void sse_ipi_inject_process (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

30.262.2.60 sse_ipi_inject_send()

```
static int sse_ipi_inject_send (
            unsigned long hartid,
            uint32_t event_id )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.61 sse_local_init()

```
static void sse_local_init (
            struct sse_hart_state * shs )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.2.62 sse_write_attrs()

```
static int sse_write_attrs (
            struct sbi_sse_event * e,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long input_phys )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.262.3 Variable Documentation

30.262.3.1 global_event_count

```
unsigned int global_event_count  [static]
```

30.262.3.2 global_events

```
struct sse_global_event* global_events  [static]
```

30.262.3.3 local_event_count

```
unsigned int local_event_count  [static]
```

30.262.3.4 shs_ptr_off

```
unsigned long shs_ptr_off  [static]
```

30.262.3.5 sse_inject_fifo_mem_off

```
unsigned long sse_inject_fifo_mem_off  [static]
```

30.262.3.6 sse_inject_fifo_off

```
unsigned long sse_inject_fifo_off  [static]
```

30.262.3.7 sse_ipi_inject_event

```
u32 sse_ipi_inject_event = SBI_IPI_EVENT_MAX  [static]
```

30.262.3.8 sse_ipi_inject_ops

```
struct sbi_ipi_event_ops sse_ipi_inject_ops [static]
```

Initial value:

```
= {
    .name    = "IPI_SSE_INJECT",
    .process = sse_ipi_inject_process,
}
```

30.262.3.9 supported_events

```
const uint32_t supported_events[] [static]
```

Initial value:

```
= {
    SBI_SSE_EVENT_LOCAL_RAS,
    SBI_SSE_EVENT_LOCAL_DOUBLE_TRAP,
    SBI_SSE_EVENT_GLOBAL_RAS,
    SBI_SSE_EVENT_LOCAL_PMU,
    SBI_SSE_EVENT_LOCAL_SOFTWARE,
    SBI_SSE_EVENT_GLOBAL_SOFTWARE,
}
```

30.263 lib/sbi/sbi_string.c File Reference

```
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_string.c:

Functions

- int sbi_strcmp (const char *a, const char *b)
- int sbi_strncmp (const char *a, const char *b, size_t count)
- size_t sbi_strlen (const char *str)
- size_t sbi_strnlen (const char *str, size_t count)
- char * sbi_strcpy (char *dest, const char *src)
- char * sbi_strncpy (char *dest, const char *src, size_t count)
- char * sbi_strchr (const char *s, int c)
- char * sbi_strrchr (const char *s, int c)
- void * sbi_memset (void *s, int c, size_t count)
- void * sbi_memcpy (void *dest, const void *src, size_t count)
- void * sbi_memmove (void *dest, const void *src, size_t count)
- int sbi_memcmp (const void *s1, const void *s2, size_t count)
- void * sbi_memchr (const void *s, int c, size_t count)

30.263.1 Function Documentation

30.263.1.1 sbi_memchr()

```
void * sbi_memchr (
            const void * s,
            int c,
            size_t count )
```

30.263.1.2 sbi_memcmp()

```
int sbi_memcmp (
            const void * s1,
            const void * s2,
            size_t count )
```

Here is the caller graph for this function:

30.263.1.3 sbi_memcpy()

```
void * sbi_memcpy (
            void * dest,
            const void * src,
            size_t count )
```

Here is the caller graph for this function:

30.263.1.4 sbi_memmove()

```
void * sbi_memmove (
            void * dest,
            const void * src,
            size_t count )
```

30.263.1.5 sbi_memset()

```
void * sbi_memset (
            void * s,
            int c,
            size_t count )
```

Here is the caller graph for this function:

30.263.1.6 sbi_strchr()

```
char * sbi_strchr (
            const char * s,
            int c )
```

30.263.1.7 sbi_strcmp()

```
int sbi_strcmp (
```

```
            const char * a,
            const char * b )
```

30.263.1.8 sbi_strcpy()

```
char * sbi_strcpy (
            char * dest,
            const char * src )
```

30.263.1.9 sbi_strlen()

```
size_t sbi_strlen (
            const char * str )
```

Here is the caller graph for this function:

30.263.1.10 sbi_strncmp()

```
int sbi_strncmp (
            const char * a,
            const char * b,
            size_t count )
```

30.263.1.11 sbi_strncpy()

```
char * sbi_strncpy (
            char * dest,
            const char * src,
            size_t count )
```

Here is the caller graph for this function:

30.263.1.12 sbi_strnlen()

```
size_t sbi_strnlen (
            const char * str,
            size_t count )
```

30.263.1.13 sbi_strrchr()

```
char * sbi_strrchr (
            const char * s,
            int c )
```

Here is the call graph for this function:

30.264 lib/sbi/sbi_system.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_system.h>
```

```
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_init.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_system.c:

Functions

- static SBI_LIST_HEAD (reset_devices_list)
- const struct sbi_system_reset_device ∗ sbi_system_reset_get_device (u32 reset_type, u32 reset_reason)
- void sbi_system_reset_add_device (struct sbi_system_reset_device ∗dev)
- bool sbi_system_reset_supported (u32 reset_type, u32 reset_reason)
- void __noreturn sbi_system_reset (u32 reset_type, u32 reset_reason)
- const struct sbi_system_suspend_device ∗ sbi_system_suspend_get_device (void)
- void sbi_system_suspend_set_device (struct sbi_system_suspend_device ∗dev)
- static int sbi_system_suspend_test_check (u32 sleep_type)
- static int sbi_system_suspend_test_suspend (u32 sleep_type, unsigned long mmode_resume_addr)
- void sbi_system_suspend_test_enable (void)
- bool sbi_system_suspend_supported (u32 sleep_type)
- int sbi_system_suspend (u32 sleep_type, ulong resume_addr, ulong opaque)

Variables

- static const struct sbi_system_suspend_device ∗ suspend_dev = NULL
- static struct sbi_system_suspend_device sbi_system_suspend_test

30.264.1 Function Documentation

30.264.1.1 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            reset_devices_list  )  [static]
```

30.264.1.2 sbi_system_reset()

```
void __noreturn sbi_system_reset (
            u32 reset_type,
            u32 reset_reason )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.264.1.3 sbi_system_reset_add_device()

```
void sbi_system_reset_add_device (
            struct sbi_system_reset_device * dev )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.264.1.4 sbi_system_reset_get_device()

```
const struct sbi_system_reset_device * sbi_system_reset_get_device (
            u32 reset_type,
            u32 reset_reason )
```

lowest priority - any non zero is our candidate
reset_type not supportedHere is the call graph for this function:

Here is the caller graph for this function:

30.264.1.5 sbi_system_reset_supported()

```
bool sbi_system_reset_supported (
```

```
            u32 reset_type,
            u32 reset_reason )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.264.1.6 sbi_system_suspend()

```
int sbi_system_suspend (
            u32 sleep_type,
            ulong resume_addr,
            ulong opaque )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.264.1.7 sbi_system_suspend_get_device()

```
const struct sbi_system_suspend_device * sbi_system_suspend_get_device (
            void  )
```

Here is the caller graph for this function:

30.264.1.8 sbi_system_suspend_set_device()

```
void sbi_system_suspend_set_device (
            struct sbi_system_suspend_device * dev )
```

Here is the caller graph for this function:

30.264.1.9 sbi_system_suspend_supported()

```
bool sbi_system_suspend_supported (
            u32 sleep_type )
```

Here is the caller graph for this function:

30.264.1.10 sbi_system_suspend_test_check()

```
static int sbi_system_suspend_test_check (
            u32 sleep_type )  [static]
```

30.264.1.11 sbi_system_suspend_test_enable()

```
void sbi_system_suspend_test_enable (
            void  )
```

Here is the call graph for this function:

30.264.1.12 sbi_system_suspend_test_suspend()

```
static int sbi_system_suspend_test_suspend (
            u32 sleep_type,
            unsigned long mmode_resume_addr )  [static]
```

Here is the call graph for this function:

30.264.2 Variable Documentation

30.264.2.1 sbi_system_suspend_test

```
struct sbi_system_suspend_device sbi_system_suspend_test  [static]
```

Initial value:

```
= {
    .name = "system-suspend-test",
    .system_suspend_check = sbi_system_suspend_test_check,
    .system_suspend = sbi_system_suspend_test_suspend,
}
```

30.264.2.2 suspend_dev

```
const struct sbi_system_suspend_device* suspend_dev = NULL  [static]
```

30.265 lib/sbi/sbi_timer.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_timer.c:

Functions

- static u64 get_ticks (void)
- static void nop_delay_fn (void ∗opaque)
- void sbi_timer_delay_loop (ulong units, u64 unit_freq, void(∗delay_fn)(void ∗), void ∗opaque)
- bool sbi_timer_waitms_until (bool(∗predicate)(void ∗), void ∗arg, uint64_t timeout_ms)
- u64 sbi_timer_value (void)
- u64 sbi_timer_virt_value (void)
- u64 sbi_timer_get_delta (void)
- void sbi_timer_set_delta (ulong delta)
- void sbi_timer_event_start (u64 next_event)
- void sbi_timer_process (void)
- const struct sbi_timer_device ∗ sbi_timer_get_device (void)
- void sbi_timer_set_device (const struct sbi_timer_device ∗dev)
- int sbi_timer_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_timer_exit (struct sbi_scratch ∗scratch)

Variables

- static unsigned long time_delta_off
- static u64(∗ get_time_val)(void)
- static const struct sbi_timer_device ∗ timer_dev = NULL

30.265.1 Function Documentation

30.265.1.1 get_ticks()

```
static u64 get_ticks (
            void ) [static]
```

Here is the caller graph for this function:

30.265.1.2 nop_delay_fn()

```
static void nop_delay_fn (
            void * opaque ) [static]
```

Here is the caller graph for this function:

30.265.1.3 sbi_timer_delay_loop()

```
void sbi_timer_delay_loop (
            ulong units,
            u64 unit_freq,
            void(*)(void *) delay_fn,
            void * opaque )
```

Generic delay loop of desired granularity Here is the call graph for this function:

Here is the caller graph for this function:

30.265.1.4 sbi_timer_event_start()

```
void sbi_timer_event_start (
            u64 next_event )
```

Start timer event for current HART Update the stimecmp directly if available. This allows the older software to leverage sstc extension on newer hardware.Here is the call graph for this function:

Here is the caller graph for this function:

30.265.1.5 sbi_timer_exit()

```
void sbi_timer_exit (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.265.1.6 sbi_timer_get_delta()

```
u64 sbi_timer_get_delta (
            void  )
```

Get timer delta value for current HART Here is the caller graph for this function:

30.265.1.7 sbi_timer_get_device()

```
const struct sbi_timer_device * sbi_timer_get_device (
            void  )
```

Get current timer device Here is the caller graph for this function:

30.265.1.8 sbi_timer_init()

```
int sbi_timer_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.265.1.9 sbi_timer_process()

```
void sbi_timer_process (
            void  )
```

Process timer event for current HART Here is the call graph for this function:

Here is the caller graph for this function:

30.265.1.10 sbi_timer_set_delta()

```
void sbi_timer_set_delta (
            ulong delta )
```

Set timer delta value for current HART Here is the caller graph for this function:

30.265.1.11 sbi_timer_set_device()

```
void sbi_timer_set_device (
            const struct sbi_timer_device * dev )
```

Register timer device Here is the caller graph for this function:

30.265.1.12 sbi_timer_value()

```
u64 sbi_timer_value (
            void )
```

Get timer value for current HART Here is the caller graph for this function:

30.265.1.13 sbi_timer_virt_value()

```
u64 sbi_timer_virt_value (
            void )
```

Get virtualized timer value for current HART Here is the call graph for this function:

Here is the caller graph for this function:

30.265.1.14 sbi_timer_waitms_until()

```
bool sbi_timer_waitms_until (
            bool(*)(void *) predicate,
            void * arg,
            uint64_t timeout_ms )
```

A blocking function that will wait until `predicate` returns true or `timeout_ms` milliseconds elapsed. `arg` will be passed as argument to `predicate` function.

Parameters

predicate	Pointer to a function that returns true if certain condition is met. It shouldn't block the code execution.
arg	Argument to pass to `predicate`.
timeout_ms	Timeout value in milliseconds. The function will return false if `timeout_ms` time period elapsed but still `predicate` doesn't return true.

Returns

true if `predicate` returns true within `timeout_ms`, false otherwise.

Here is the call graph for this function:

30.265.2 Variable Documentation

30.265.2.1 get_time_val

```
u64(* get_time_val) (void) (
            void  )  [static]
```

30.265.2.2 time_delta_off

```
unsigned long time_delta_off  [static]
```

30.265.2.3 timer_dev

```
const struct sbi_timer_device* timer_dev = NULL  [static]
```

30.266 lib/sbi/sbi_tlb.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_barrier.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_fifo.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_tlb.h>
#include <sbi/sbi_hfence.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
```

Include dependency graph for sbi_tlb.c:

Functions

- static void tlb_flush_all (void)
- static void sbi_tlb_local_hfence_vvma (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_local_hfence_gvma (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_local_sfence_vma (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_local_hfence_vvma_asid (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_local_hfence_gvma_vmid (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_local_sfence_vma_asid (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_local_fence_i (struct sbi_tlb_info *tinfo)
- static void tlb_entry_local_process (struct sbi_tlb_info *data)
- static void tlb_entry_process (struct sbi_tlb_info *tinfo)

- static bool tlb_process_once (struct sbi_scratch ∗scratch)
- static void tlb_process (struct sbi_scratch ∗scratch)
- static void tlb_sync (struct sbi_scratch ∗scratch)
- static int tlb_range_check (struct sbi_tlb_info ∗curr, struct sbi_tlb_info ∗next)
- static int tlb_update_cb (void ∗in, void ∗data)
- static int tlb_update (struct sbi_scratch ∗scratch, struct sbi_scratch ∗remote_scratch, u32 remote_hartindex, void ∗data)
- int sbi_tlb_request (ulong hmask, ulong hbase, struct sbi_tlb_info ∗tinfo)
- int sbi_tlb_init (struct sbi_scratch ∗scratch, bool cold_boot)

Variables

- static unsigned long tlb_sync_off
- static unsigned long tlb_fifo_off
- static unsigned long tlb_fifo_mem_off
- static unsigned long tlb_range_flush_limit
- static struct sbi_ipi_event_ops tlb_ops
- static u32 tlb_event = SBI_IPI_EVENT_MAX
- static const u32 tlb_type_to_pmu_fw_event [SBI_TLB_TYPE_MAX]

30.266.1 Function Documentation

30.266.1.1 sbi_tlb_init()

```
int sbi_tlb_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.2 sbi_tlb_local_fence_i()

```
static void sbi_tlb_local_fence_i (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.3 sbi_tlb_local_hfence_gvma()

```
static void sbi_tlb_local_hfence_gvma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.4 sbi_tlb_local_hfence_gvma_vmid()

```
static void sbi_tlb_local_hfence_gvma_vmid (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.5 sbi_tlb_local_hfence_vvma()

```
static void sbi_tlb_local_hfence_vvma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.6 sbi_tlb_local_hfence_vvma_asid()

```
static void sbi_tlb_local_hfence_vvma_asid (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.7 sbi_tlb_local_sfence_vma()

```
static void sbi_tlb_local_sfence_vma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.8 sbi_tlb_local_sfence_vma_asid()

```
static void sbi_tlb_local_sfence_vma_asid (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.9 sbi_tlb_request()

```
int sbi_tlb_request (
            ulong hmask,
```

```
            ulong hbase,
            struct sbi_tlb_info * tinfo )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.10 tlb_entry_local_process()

```
static void tlb_entry_local_process (
            struct sbi_tlb_info * data )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.11 tlb_entry_process()

```
static void tlb_entry_process (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.12 tlb_flush_all()

```
static void tlb_flush_all (
            void ) [static]
```

Here is the caller graph for this function:

30.266.1.13 tlb_process()

```
static void tlb_process (
            struct sbi_scratch * scratch ) [static]
```

Here is the call graph for this function:

30.266.1.14 tlb_process_once()

```
static bool tlb_process_once (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.15 tlb_range_check()

```
static int tlb_range_check (
```

```
                struct sbi_tlb_info * curr,
                struct sbi_tlb_info * next )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.16 tlb_sync()

```
static void tlb_sync (
                struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.266.1.17 tlb_update()

```
static int tlb_update (
            struct sbi_scratch * scratch,
            struct sbi_scratch * remote_scratch,
            u32 remote_hartindex,
            void * data )  [static]
```

For now, Busy loop until there is space in the fifo. There may be case where target hart is also enqueue in source hart's fifo. Both hart may busy loop leading to a deadlock. TODO: Introduce a wait/wakeup event mechanism to handle this properly.Here is the call graph for this function:

30.266.1.18 tlb_update_cb()

```
static int tlb_update_cb (
            void * in,
            void * data )  [static]
```

Call back to decide if an inplace fifo update is required or next entry can can be skipped. Here are the different cases that are being handled.
Case1: if next flush request range lies within one of the existing entry, skip the next entry. Case2: if flush request range in current fifo entry lies within next flush request, update the current entry.
Note: We can not issue a fifo reset anymore if a complete vma flush is requested. This is because we are queueing FENCE.I requests as well now. To ease up the pressure in enqueue/fifo sync path, try to dequeue 1 element before continuing the while loop. This method is preferred over wfi/ipi because of MMIO cost involved in later method. Here is the call graph for this function:

Here is the caller graph for this function:

30.266.2 Variable Documentation

30.266.2.1 tlb_event

```
u32 tlb_event = SBI_IPI_EVENT_MAX  [static]
```

30.266.2.2 tlb_fifo_mem_off

```
unsigned long tlb_fifo_mem_off  [static]
```

30.266.2.3 tlb_fifo_off

```
unsigned long tlb_fifo_off  [static]
```

30.266.2.4 tlb_ops

```
struct sbi_ipi_event_ops tlb_ops  [static]
```

Initial value:

```
= {
    .name = "IPI_TLB",
    .update = tlb_update,
    .sync = tlb_sync,
    .process = tlb_process,
}
```

30.266.2.5 tlb_range_flush_limit

```
unsigned long tlb_range_flush_limit  [static]
```

30.266.2.6 tlb_sync_off

```
unsigned long tlb_sync_off  [static]
```

30.266.2.7 tlb_type_to_pmu_fw_event

```
const u32 tlb_type_to_pmu_fw_event[SBI_TLB_TYPE_MAX]  [static]
```

Initial value:

```
= {
    [SBI_TLB_FENCE_I] = SBI_PMU_FW_FENCE_I_SENT,
    [SBI_TLB_SFENCE_VMA] = SBI_PMU_FW_SFENCE_VMA_SENT,
    [SBI_TLB_SFENCE_VMA_ASID] = SBI_PMU_FW_SFENCE_VMA_ASID_SENT,
    [SBI_TLB_HFENCE_GVMA_VMID] = SBI_PMU_FW_HFENCE_GVMA_VMID_SENT,
    [SBI_TLB_HFENCE_GVMA] = SBI_PMU_FW_HFENCE_GVMA_SENT,
    [SBI_TLB_HFENCE_VVMA_ASID] = SBI_PMU_FW_HFENCE_VVMA_ASID_SENT,
    [SBI_TLB_HFENCE_VVMA] = SBI_PMU_FW_HFENCE_VVMA_SENT,
}
```

30.267 lib/sbi/sbi_trap.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_illegal_insn.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_irqchip.h>
#include <sbi/sbi_trap_ldst.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_sse.h>
#include <sbi/sbi_timer.h>
```

```
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_trap.c:

Functions

- static void sbi_trap_error_one (const struct sbi_trap_context ∗tcntx, const char ∗prefix, u32 hartid, u32 depth)
- static void __noreturn sbi_trap_error (const char ∗msg, int rc, const struct sbi_trap_context ∗tcntx)
- int sbi_trap_redirect (struct sbi_trap_regs ∗regs, const struct sbi_trap_info ∗trap)
- static int sbi_trap_nonaia_irq (unsigned long irq)
- static int sbi_trap_aia_irq (void)
- struct sbi_trap_context ∗ sbi_trap_handler (struct sbi_trap_context ∗tcntx)

30.267.1 Function Documentation

30.267.1.1 sbi_trap_aia_irq()

```
static int sbi_trap_aia_irq (
            void ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.267.1.2 sbi_trap_error()

```
static void __noreturn sbi_trap_error (
            const char * msg,
            int rc,
            const struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.267.1.3 sbi_trap_error_one()

```
static void sbi_trap_error_one (
            const struct sbi_trap_context * tcntx,
            const char * prefix,
            u32 hartid,
            u32 depth )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.267.1.4 sbi_trap_handler()

```
struct sbi_trap_context * sbi_trap_handler (
            struct sbi_trap_context * tcntx )
```

Handle trap/interrupt

This function is called by firmware linked to OpenSBI library for handling trap/interrupt. It expects the following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. The 'mcause' CSR is having exception/interrupt cause
3. The 'mtval' CSR is having additional trap information
4. The 'mtval2' CSR is having additional trap information
5. The 'mtinst' CSR is having decoded trap instruction
6. Stack pointer (SP) is setup for current HART
7. Interrupts are disabled in MSTATUS CSR

Parameters

tcntx	pointer to trap context

Here is the call graph for this function:

30.267.1.5 sbi_trap_nonaia_irq()

```
static int sbi_trap_nonaia_irq (
            unsigned long irq )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.267.1.6 sbi_trap_redirect()

```
int sbi_trap_redirect (
            struct sbi_trap_regs * regs,
            const struct sbi_trap_info * trap )
```

Redirect trap to lower privledge mode (S-mode or U-mode)

Parameters

regs	pointer to register state
trap	pointer to trap details

Returns

0 on success and negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.268 lib/sbi/sbi_trap_ldst.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_fp.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap_ldst.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
#include <sbi/sbi_platform.h>
```

Include dependency graph for sbi_trap_ldst.c:

Typedefs

- typedef int(∗ sbi_trap_ld_emulator) (int rlen, union sbi_ldst_data ∗out_val, struct sbi_trap_context ∗tcntx)
- typedef int(∗ sbi_trap_st_emulator) (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context ∗tcntx)

Functions

- ulong sbi_misaligned_tinst_fixup (ulong orig_tinst, ulong new_tinst, ulong addr_offset)
- static int sbi_trap_emulate_load (struct sbi_trap_context ∗tcntx, sbi_trap_ld_emulator emu)
- static int sbi_trap_emulate_store (struct sbi_trap_context ∗tcntx, sbi_trap_st_emulator emu)
- static int sbi_misaligned_ld_emulator (int rlen, union sbi_ldst_data ∗out_val, struct sbi_trap_context ∗tcntx)
- int sbi_misaligned_load_handler (struct sbi_trap_context ∗tcntx)
- static int sbi_misaligned_st_emulator (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context ∗tcntx)
- int sbi_misaligned_store_handler (struct sbi_trap_context ∗tcntx)
- static int sbi_ld_access_emulator (int rlen, union sbi_ldst_data ∗out_val, struct sbi_trap_context ∗tcntx)
- int sbi_load_access_handler (struct sbi_trap_context ∗tcntx)
- static int sbi_st_access_emulator (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context ∗tcntx)
- int sbi_store_access_handler (struct sbi_trap_context ∗tcntx)

30.268.1 Typedef Documentation

30.268.1.1 sbi_trap_ld_emulator

```
typedef int(* sbi_trap_ld_emulator) (int rlen, union sbi_ldst_data *out_val, struct sbi_trap_context
*tcntx)
```

Load emulator callback:

Returns

rlen=success, 0=success w/o regs modification, or negative error

30.268.1.2 sbi_trap_st_emulator

```
typedef int(* sbi_trap_st_emulator) (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context
*tcntx)
```

Store emulator callback:

Returns

wlen=success, 0=success w/o regs modification, or negative error

30.268.2 Function Documentation

30.268.2.1 sbi_ld_access_emulator()

```
static int sbi_ld_access_emulator (
            int rlen,
            union sbi_ldst_data * out_val,
            struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.2 sbi_load_access_handler()

```
int sbi_load_access_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.3 sbi_misaligned_ld_emulator()

```
static int sbi_misaligned_ld_emulator (
            int rlen,
            union sbi_ldst_data * out_val,
            struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.4 sbi_misaligned_load_handler()

```
int sbi_misaligned_load_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.5 sbi_misaligned_st_emulator()

```
static int sbi_misaligned_st_emulator (
            int wlen,
            union sbi_ldst_data in_val,
            struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.6 sbi_misaligned_store_handler()

```
int sbi_misaligned_store_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.7 sbi_misaligned_tinst_fixup()

```
ulong sbi_misaligned_tinst_fixup (
            ulong orig_tinst,
            ulong new_tinst,
            ulong addr_offset )
```

Here is the caller graph for this function:

30.268.2.8 sbi_st_access_emulator()

```
static int sbi_st_access_emulator (
            int wlen,
            union sbi_ldst_data in_val,
            struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.9 sbi_store_access_handler()

```
int sbi_store_access_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.10 sbi_trap_emulate_load()

```
static int sbi_trap_emulate_load (
            struct sbi_trap_context * tcntx,
            sbi_trap_ld_emulator emu )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.268.2.11 sbi_trap_emulate_store()

```
static int sbi_trap_emulate_store (
            struct sbi_trap_context * tcntx,
            sbi_trap_st_emulator emu )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.269 lib/sbi/sbi_trap_v_ldst.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap_ldst.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
```

Include dependency graph for sbi_trap_v_ldst.c:

Functions

- int sbi_misaligned_v_ld_emulator (int rlen, union sbi_ldst_data *out_val, struct sbi_trap_context *tcntx)
- int sbi_misaligned_v_st_emulator (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context *tcntx)

30.269.1 Function Documentation

30.269.1.1 sbi_misaligned_v_ld_emulator()

```
int sbi_misaligned_v_ld_emulator (
            int rlen,
            union sbi_ldst_data * out_val,
            struct sbi_trap_context * tcntx )
```

Here is the caller graph for this function:

30.269.1.2 sbi_misaligned_v_st_emulator()

```
int sbi_misaligned_v_st_emulator (
            int wlen,
            union sbi_ldst_data in_val,
            struct sbi_trap_context * tcntx )
```

Here is the caller graph for this function:

30.270 lib/sbi/sbi_unpriv.c File Reference

```
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
```

Include dependency graph for sbi_unpriv.c:

Macros

- #define DEFINE_UNPRIVILEGED_LOAD_FUNCTION(type, insn)
- #define DEFINE_UNPRIVILEGED_STORE_FUNCTION(type, insn)

Functions

- ulong sbi_get_insn (ulong mepc, struct sbi_trap_info *trap)

30.270.1 Macro Definition Documentation

30.270.1.1 DEFINE_UNPRIVILEGED_LOAD_FUNCTION

```
#define DEFINE_UNPRIVILEGED_LOAD_FUNCTION(
            type,
```

```
                insn )
```

Value:

```
type sbi_load_##type(const type *addr,                                  \
                     struct sbi_trap_info *trap)                        \
{                                                                       \
    register ulong tinfo asm("a3");                                     \
    register ulong mstatus = 0;                                         \
    register ulong mtvec = sbi_hart_expected_trap_addr();               \
    type ret = 0;                                                       \
    trap->cause = 0;                                                    \
    asm volatile(                                                       \
        "add %[tinfo], %[taddr], zero\n"                                \
        "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"                \
        "csrrs %[mstatus], " STR(CSR_MSTATUS) ", %[mprv]\n"             \
        ".option push\n"                                                \
        ".option norvc\n"                                               \
        #insn " %[ret], %[addr]\n"                                      \
        ".option pop\n"                                                 \
        "csrw " STR(CSR_MSTATUS) ", %[mstatus]\n"                       \
        "csrw " STR(CSR_MTVEC) ", %[mtvec]"                             \
        : [mstatus] "+&r"(mstatus), [mtvec] "+&r"(mtvec),               \
          [tinfo] "+&r"(tinfo), [ret] "=&r"(ret)                        \
        : [addr] "m"(*addr), [mprv] "r"(MSTATUS_MPRV),                  \
          [taddr] "r"((ulong)trap)                                      \
        : "a4", "memory");                                              \
    return ret;                                                         \
}
```

a3 must a pointer to the sbi_trap_info and a4 is used as a temporary register in the trap handler. Make sure that compiler doesn't use a3 & a4.

30.270.1.2 DEFINE_UNPRIVILEGED_STORE_FUNCTION

```
#define DEFINE_UNPRIVILEGED_STORE_FUNCTION(
                type,
                insn )
```

Value:

```
void sbi_store_##type(type *addr, type val,                             \
                      struct sbi_trap_info *trap)                       \
{                                                                       \
    register ulong tinfo asm("a3") = (ulong)trap;                       \
    register ulong mstatus = 0;                                         \
    register ulong mtvec = sbi_hart_expected_trap_addr();               \
    trap->cause = 0;                                                    \
    asm volatile(                                                       \
        "add %[tinfo], %[taddr], zero\n"                                \
        "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"                \
        "csrrs %[mstatus], " STR(CSR_MSTATUS) ", %[mprv]\n"             \
        ".option push\n"                                                \
        ".option norvc\n"                                               \
        #insn " %[val], %[addr]\n"                                      \
        ".option pop\n"                                                 \
        "csrw " STR(CSR_MSTATUS) ", %[mstatus]\n"                       \
        "csrw " STR(CSR_MTVEC) ", %[mtvec]"                             \
        : [mstatus] "+&r"(mstatus), [mtvec] "+&r"(mtvec),               \
          [tinfo] "+&r"(tinfo)                                          \
        : [addr] "m"(*addr), [mprv] "r"(MSTATUS_MPRV),                  \
          [val] "r"(val), [taddr] "r"((ulong)trap)                      \
        : "a4", "memory");                                              \
}
```

30.270.2 Function Documentation

30.270.2.1 sbi_get_insn()

```
ulong sbi_get_insn (
            ulong mepc,
            struct sbi_trap_info * trap )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.271 lib/sbi/tests/riscv_atomic_test.c File Reference

```
#include <sbi/sbi_unit_test.h>
#include <sbi/riscv_atomic.h>
#include <sbi/sbi_bitops.h>
```

Include dependency graph for riscv_atomic_test.c:

Macros

- #define ATOMIC_TEST_VAL1 239l
- #define ATOMIC_TEST_VAL2 30l
- #define ATOMIC_TEST_VAL3 2024l

- #define ATOMIC_TEST_BIT_NUM 3
- #define ATOMIC_TEST_RAW_BIT_CELL 1
- #define ATOMIC_TEST_RAW_BIT_NUM 15

Functions

- static void atomic_test_suite_init (void)
- static void atomic_rw_test (struct sbiunit_test_case *test)
- static void add_return_test (struct sbiunit_test_case *test)
- static void sub_return_test (struct sbiunit_test_case *test)
- static void cmpxchg_test (struct sbiunit_test_case *test)
- static void atomic_xchg_test (struct sbiunit_test_case *test)
- static void atomic_raw_set_bit_test (struct sbiunit_test_case *test)
- static void atomic_raw_clear_bit_test (struct sbiunit_test_case *test)
- static void atomic_set_bit_test (struct sbiunit_test_case *test)
- static void atomic_clear_bit_test (struct sbiunit_test_case *test)

Variables

- static atomic_t test_atomic
- static struct sbiunit_test_case atomic_test_cases [ ]
- const struct sbiunit_test_suite atomic_test_suite

30.271.1 Macro Definition Documentation

30.271.1.1 ATOMIC_TEST_BIT_NUM

```
#define ATOMIC_TEST_BIT_NUM 3
```

30.271.1.2 ATOMIC_TEST_RAW_BIT_CELL

```
#define ATOMIC_TEST_RAW_BIT_CELL 1
```

30.271.1.3 ATOMIC_TEST_RAW_BIT_NUM

```
#define ATOMIC_TEST_RAW_BIT_NUM 15
```

30.271.1.4 ATOMIC_TEST_VAL1

```
#define ATOMIC_TEST_VAL1 2391
```

30.271.1.5 ATOMIC_TEST_VAL2

```
#define ATOMIC_TEST_VAL2 301
```

30.271.1.6 ATOMIC_TEST_VAL3

```
#define ATOMIC_TEST_VAL3 20241
```

30.271.2 Function Documentation

30.271.2.1 add_return_test()

```
static void add_return_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.2.2 atomic_clear_bit_test()

```
static void atomic_clear_bit_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.2.3 atomic_raw_clear_bit_test()

```
static void atomic_raw_clear_bit_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.2.4 atomic_raw_set_bit_test()

```
static void atomic_raw_set_bit_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.2.5 atomic_rw_test()

```
static void atomic_rw_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.2.6 atomic_set_bit_test()

```
static void atomic_set_bit_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.2.7 atomic_test_suite_init()

```
static void atomic_test_suite_init (
            void  )  [static]
```

30.271.2.8 atomic_xchg_test()

```
static void atomic_xchg_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.2.9 cmpxchg_test()

```
static void cmpxchg_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.2.10 sub_return_test()

```
static void sub_return_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.271.3 Variable Documentation

30.271.3.1 atomic_test_cases

```
struct sbiunit_test_case atomic_test_cases[] [static]
```

Initial value:

```
= {
    SBIUNIT_TEST_CASE(atomic_rw_test),
    SBIUNIT_TEST_CASE(add_return_test),
    SBIUNIT_TEST_CASE(sub_return_test),
    SBIUNIT_TEST_CASE(cmpxchg_test),
    SBIUNIT_TEST_CASE(atomic_xchg_test),
    SBIUNIT_TEST_CASE(atomic_raw_set_bit_test),
    SBIUNIT_TEST_CASE(atomic_raw_clear_bit_test),
    SBIUNIT_TEST_CASE(atomic_set_bit_test),
    SBIUNIT_TEST_CASE(atomic_clear_bit_test),
    SBIUNIT_END_CASE,
}
```

30.271.3.2 atomic_test_suite

```
const struct sbiunit_test_suite atomic_test_suite
```

Initial value:

```
= {
    .name = "atomic_test_suite",
    .cases = atomic_test_cases,
    .init = atomic_test_suite_init
}
```

30.271.3.3 test_atomic

```
atomic_t test_atomic [static]
```

30.272 lib/sbi/tests/riscv_locks_test.c File Reference

```
#include <sbi/sbi_unit_test.h>
#include <sbi/riscv_locks.h>
```

Include dependency graph for riscv_locks_test.c:

Functions

- static void spin_lock_test (struct sbiunit_test_case *test)
- static void spin_trylock_fail (struct sbiunit_test_case *test)
- static void spin_trylock_success (struct sbiunit_test_case *test)
- SBIUNIT_TEST_SUITE (locks_test_suite, locks_test_cases)

Variables

- static spinlock_t test_lock = SPIN_LOCK_INITIALIZER
- static struct sbiunit_test_case locks_test_cases [ ]

30.272.1 Function Documentation

30.272.1.1 SBIUNIT_TEST_SUITE()

```
SBIUNIT_TEST_SUITE (
            locks_test_suite ,
            locks_test_cases  )
```

30.272.1.2 spin_lock_test()

```
static void spin_lock_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.272.1.3 spin_trylock_fail()

```
static void spin_trylock_fail (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.272.1.4 spin_trylock_success()

```
static void spin_trylock_success (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.272.2 Variable Documentation

30.272.2.1 locks_test_cases

```
struct sbiunit_test_case locks_test_cases[]  [static]
```

Initial value:

```
= {
    SBIUNIT_TEST_CASE(spin_lock_test),
    SBIUNIT_TEST_CASE(spin_trylock_fail),
    SBIUNIT_TEST_CASE(spin_trylock_success),
    SBIUNIT_END_CASE,
}
```

30.272.2.2 test_lock

```
spinlock_t test_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.273 lib/sbi/tests/sbi_bitmap_test.c File Reference

```
#include <sbi/sbi_bitmap.h>
#include <sbi/sbi_unit_test.h>
```

Include dependency graph for sbi_bitmap_test.c:

Macros

- #define DATA_SIZE sizeof(data_zero)
- #define DATA_BIT_SIZE (DATA_SIZE ∗ 8)

Functions

- static void bitmap_and_test (struct sbiunit_test_case ∗test)
- static void bitmap_or_test (struct sbiunit_test_case ∗test)
- static void bitmap_xor_test (struct sbiunit_test_case ∗test)
- SBIUNIT_TEST_SUITE (bitmap_test_suite, bitmap_test_cases)

Variables

- static unsigned long data_a [ ] = { 0xDEADBEEF, 0x00BAB10C, 0x1BADB002, 0xABADBABE }
- static unsigned long data_b [ ] = { 0xC00010FF, 0x00BAB10C, 0xBAAAAAAD, 0xBADDCAFE }
- static unsigned long data_zero [ ] = { 0, 0, 0, 0 }
- static struct sbiunit_test_case bitmap_test_cases [ ]

30.273.1 Macro Definition Documentation

30.273.1.1 DATA_BIT_SIZE

```
#define DATA_BIT_SIZE (DATA_SIZE * 8)
```

30.273.1.2 DATA_SIZE

```
#define DATA_SIZE sizeof(data_zero)
```

30.273.2 Function Documentation

30.273.2.1 bitmap_and_test()

```
static void bitmap_and_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.273.2.2 bitmap_or_test()

```
static void bitmap_or_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.273.2.3 bitmap_xor_test()

```
static void bitmap_xor_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.273.2.4 SBIUNIT_TEST_SUITE()

```
SBIUNIT_TEST_SUITE (
            bitmap_test_suite ,
            bitmap_test_cases  )
```

30.273.3 Variable Documentation

30.273.3.1 bitmap_test_cases

```
struct sbiunit_test_case bitmap_test_cases[]  [static]
```

Initial value:

```
= {
    SBIUNIT_TEST_CASE(bitmap_and_test),
    SBIUNIT_TEST_CASE(bitmap_or_test),
    SBIUNIT_TEST_CASE(bitmap_xor_test),
    SBIUNIT_END_CASE,
}
```

30.273.3.2 data_a

```
unsigned long data_a[] = { 0xDEADBEEF, 0x00BAB10C, 0x1BADB002, 0xABADBABE }  [static]
```

30.273.3.3 data_b

```
unsigned long data_b[] = { 0xC00010FF, 0x00BAB10C, 0xBAAAAAAD, 0xBADDCAFE }  [static]
```

30.273.3.4 data_zero

```
unsigned long data_zero[] = { 0, 0, 0, 0 }  [static]
```

30.274 lib/sbi/tests/sbi_console_test.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_unit_test.h>
```

Include dependency graph for sbi_console_test.c:

Macros

- #define TEST_CONSOLE_BUF_LEN 1024
- #define PUTS_TEST(test, expected, str)
- #define PRINTF_TEST(test, expected, format, ...)

Functions

- static void test_console_putc (char c)
- static void clear_test_console_buf (void)
- static void test_console_begin (const struct sbi_console_device *device)
- static void test_console_end (void)
- static void putc_test (struct sbiunit_test_case *test)
- static void puts_test (struct sbiunit_test_case *test)
- static void printf_test (struct sbiunit_test_case *test)
- SBIUNIT_TEST_SUITE (console_test_suite, console_test_cases)

Variables

- static const struct sbi_console_device * old_dev
- static char test_console_buf [TEST_CONSOLE_BUF_LEN]
- static u32 test_console_buf_pos
- static spinlock_t test_console_lock = SPIN_LOCK_INITIALIZER
- static const struct sbi_console_device test_console_dev
- static struct sbiunit_test_case console_test_cases [ ]

30.274.1 Macro Definition Documentation

30.274.1.1 PRINTF_TEST

```
#define PRINTF_TEST(
            test,
            expected,
            format,
            ...  )
```

Value:

```
do {        \
spin_lock(&test_console_lock);          \
clear_test_console_buf();           \
test_console_begin(&test_console_dev);          \
```

```
    size_t __res = sbi_printf(format, ##__VA_ARGS__);    \
    test_console_end();                                  \
    SBIUNIT_ASSERT_EQ(test, __res, sbi_strlen(expected)); \
    SBIUNIT_ASSERT_STREQ(test, test_console_buf, expected,  \
                sbi_strlen(expected));        \
    spin_unlock(&test_console_lock);            \
} while (0)
```

30.274.1.2 PUTS_TEST

```
#define PUTS_TEST(
                test,
                expected,
                str )
```

Value:

```
    do {            \
    spin_lock(&test_console_lock);              \
    clear_test_console_buf();                \
    test_console_begin(&test_console_dev);          \
    sbi_puts(str);                          \
    test_console_end();                     \
    SBIUNIT_ASSERT_STREQ(test, test_console_buf, expected,  \
                sbi_strlen(expected));        \
    spin_unlock(&test_console_lock);            \
} while (0)
```

30.274.1.3 TEST_CONSOLE_BUF_LEN

```
#define TEST_CONSOLE_BUF_LEN 1024
```

30.274.2 Function Documentation

30.274.2.1 clear_test_console_buf()

```
static void clear_test_console_buf (
              void  )  [static]
```

Here is the caller graph for this function:

30.274.2.2 printf_test()

```
static void printf_test (
              struct sbiunit_test_case * test )  [static]
```

30.274.2.3 putc_test()

```
static void putc_test (
              struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.274.2.4 puts_test()

```
static void puts_test (
            struct sbiunit_test_case * test )  [static]
```

30.274.2.5 SBIUNIT_TEST_SUITE()

```
SBIUNIT_TEST_SUITE (
            console_test_suite ,
            console_test_cases  )
```

30.274.2.6 test_console_begin()

```
static void test_console_begin (
            const struct sbi_console_device * device )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.274.2.7 test_console_end()

```
static void test_console_end (
            void  )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.274.2.8 test_console_putc()

```
static void test_console_putc (
            char c )  [static]
```

30.274.3 Variable Documentation

30.274.3.1 console_test_cases

```
struct sbiunit_test_case console_test_cases[]  [static]
```

Initial value:

```
= {
    SBIUNIT_TEST_CASE(putc_test),
    SBIUNIT_TEST_CASE(puts_test),
    SBIUNIT_TEST_CASE(printf_test),
    SBIUNIT_END_CASE,
}
```

30.274.3.2 old_dev

```
const struct sbi_console_device* old_dev  [static]
```

30.274.3.3 test_console_buf

```
char test_console_buf[TEST_CONSOLE_BUF_LEN]  [static]
```

30.274.3.4 test_console_buf_pos

```
u32 test_console_buf_pos  [static]
```

30.274.3.5 test_console_dev

```
const struct sbi_console_device test_console_dev  [static]
```

Initial value:

```
= {
    .name = "Test console device",
    .console_putc = test_console_putc,
}
```

30.274.3.6 test_console_lock

```
spinlock_t test_console_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.275 lib/sbi/tests/sbi_math_test.c File Reference

```
#include <sbi/sbi_math.h>
#include <sbi/sbi_unit_test.h>
```

Include dependency graph for sbi_math_test.c:

Functions

- static void log2roundup_test (struct sbiunit_test_case *test)
- SBIUNIT_TEST_SUITE (math_test_suite, math_test_cases)

Variables

- static struct sbiunit_test_case math_test_cases [ ]

30.275.1 Function Documentation

30.275.1.1 log2roundup_test()

```
static void log2roundup_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:

30.275.1.2 SBIUNIT_TEST_SUITE()

```
SBIUNIT_TEST_SUITE (
            math_test_suite ,
            math_test_cases  )
```

30.275.2 Variable Documentation

30.275.2.1 math_test_cases

```
struct sbiunit_test_case math_test_cases[] [static]
```

Initial value:

```
= {
    SBIUNIT_TEST_CASE(log2roundup_test),
    SBIUNIT_END_CASE,
}
```

30.276 lib/sbi/tests/sbi_unit_test.c File Reference

```
#include <sbi/sbi_unit_test.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for sbi_unit_test.c:

Macros

- #define ANSI_COLOR_GREEN "\x1b[32m"
- #define ANSI_COLOR_RED "\x1b[31m"
- #define ANSI_COLOR_RESET "\x1b[0m"

Functions

- static void run_test_suite (struct sbiunit_test_suite *suite)
- void run_all_tests (void)

Variables

- struct sbiunit_test_suite *const sbi_unit_tests [ ]

30.276.1 Macro Definition Documentation

30.276.1.1 ANSI_COLOR_GREEN

```
#define ANSI_COLOR_GREEN "\x1b[32m"
```

30.276.1.2 ANSI_COLOR_RED

```
#define ANSI_COLOR_RED "\x1b[31m"
```

30.276.1.3 ANSI_COLOR_RESET

```
#define ANSI_COLOR_RESET "\x1b[0m"
```

30.276.2 Function Documentation

30.276.2.1 run_all_tests()

```
void run_all_tests (
            void  )
```

Here is the call graph for this function:

30.276.2.2 run_test_suite()

```
static void run_test_suite (
            struct sbiunit_test_suite * suite )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.276.3 Variable Documentation

30.276.3.1 sbi_unit_tests

```
struct sbiunit_test_suite* const sbi_unit_tests[]  [extern]
```

30.277 lib/utils/cppc/fdt_cppc.c File Reference

`#include <sbi_utils/cppc/fdt_cppc.h>`

Include dependency graph for fdt_cppc.c:

Functions

- void fdt_cppc_init (const void *fdt)

Variables

- const struct fdt_driver *const fdt_cppc_drivers [ ]

30.277.1 Function Documentation

30.277.1.1 fdt_cppc_init()

```
void fdt_cppc_init (
            const void * fdt )
```

Here is the call graph for this function:

30.277.2 Variable Documentation

30.277.2.1 fdt_cppc_drivers

```
const struct fdt_driver* const fdt_cppc_drivers[] [extern]
```

30.278 lib/utils/cppc/fdt_cppc_rpmi.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_cppc.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/cppc/fdt_cppc.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/mailbox/fdt_mailbox.h>
#include <sbi_utils/mailbox/rpmi_mailbox.h>
```

Include dependency graph for fdt_cppc_rpmi.c:

Macros

- #define RPMI_CPPC_HART_FASTCHAN_SIZE 0x8

Functions

- static struct rpmi_cppc ∗ rpmi_cppc_get_pointer (u32 hartid)
- static void rpmi_cppc_fc_db_trigger (struct rpmi_cppc ∗cppc)
- static int rpmi_cppc_read (unsigned long reg, u64 ∗val)
- static int rpmi_cppc_write (unsigned long reg, u64 val)
- static int rpmi_cppc_probe (unsigned long reg)
- static int rpmi_cppc_update_hart_scratch (struct mbox_chan ∗chan)
- static int rpmi_cppc_cold_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static unsigned long rpmi_cppc_offset
- static struct sbi_cppc_device sbi_rpmi_cppc
- static const struct fdt_match rpmi_cppc_match [ ]
- struct fdt_driver fdt_cppc_rpmi

30.278.1 Macro Definition Documentation

30.278.1.1 RPMI_CPPC_HART_FASTCHAN_SIZE

```
#define RPMI_CPPC_HART_FASTCHAN_SIZE 0x8
```

Per hart RPMI CPPC fast channel size (bytes) PASSIVE MODE: 0x0: DESIRED_PERFORMANCE (4-byte) 0x4↩ : __RESERVED (4-byte) ACTIVE MODE: (not supported yet) 0x0: MINIMUM PERFORMANCE (4-byte) 0x4↩ : MAXIMUM PERFORMANCE (4-byte)

30.278.2 Function Documentation

30.278.2.1 rpmi_cppc_cold_init()

```
static int rpmi_cppc_cold_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match ) [static]
```

Here is the call graph for this function:

30.278.2.2 rpmi_cppc_fc_db_trigger()

```
static void rpmi_cppc_fc_db_trigger (
            struct rpmi_cppc * cppc ) [static]
```

Here is the caller graph for this function:

30.278.2.3 rpmi_cppc_get_pointer()

```
static struct rpmi_cppc * rpmi_cppc_get_pointer (
            u32 hartid ) [static]
```

Here is the caller graph for this function:

30.278.2.4 rpmi_cppc_probe()

```
static int rpmi_cppc_probe (
            unsigned long reg )  [static]
```

Here is the call graph for this function:

30.278.2.5 rpmi_cppc_read()

```
static int rpmi_cppc_read (
            unsigned long reg,
            u64 * val )  [static]
```

Here is the call graph for this function:

30.278.2.6 rpmi_cppc_update_hart_scratch()

```
static int rpmi_cppc_update_hart_scratch (
            struct mbox_chan * chan )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.278.2.7 rpmi_cppc_write()

```
static int rpmi_cppc_write (
            unsigned long reg,
            u64 val )  [static]
```

Here is the call graph for this function:

30.278.3 Variable Documentation

30.278.3.1 fdt_cppc_rpmi

```
struct fdt_driver fdt_cppc_rpmi
```

Initial value:

```
= {
    .match_table = rpmi_cppc_match,
    .init = rpmi_cppc_cold_init,
    .experimental = true,
}
```

30.278.3.2 rpmi_cppc_match

```
const struct fdt_match rpmi_cppc_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,rpmi-cppc" },
    {},
}
```

30.278.3.3 rpmi_cppc_offset

```
unsigned long rpmi_cppc_offset  [static]
```

30.278.3.4 sbi_rpmi_cppc

```
struct sbi_cppc_device sbi_rpmi_cppc  [static]
```

Initial value:

```
= {
    .name       = "rpmi-cppc",
    .cppc_read  = rpmi_cppc_read,
    .cppc_write = rpmi_cppc_write,
    .cppc_probe = rpmi_cppc_probe,
}
```

30.279 lib/utils/fdt/fdt_domain.c File Reference

```
#include <libfdt.h>
#include <libfdt_env.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_domain.h>
#include <sbi_utils/fdt/fdt_helper.h>
```

Include dependency graph for fdt_domain.c:

Macros

- #define DISABLE_DEVICES_MASK
- #define FDT_DOMAIN_REGION_MAX_COUNT 16

Functions

- int fdt_iterate_each_domain (void ∗fdt, void ∗opaque, int(∗fn)(void ∗fdt, int domain_offset, void ∗opaque))
- static int fdt_iterate_each_domain_ro (const void ∗fdt, void ∗opaque, int(∗fn)(const void ∗fdt, int domain_↩ offset, void ∗opaque))
- int fdt_iterate_each_memregion (void ∗fdt, int domain_offset, void ∗opaque, int(∗fn)(void ∗fdt, int domain_↩ offset, int region_offset, u32 region_access, void ∗opaque))
- static int fdt_iterate_each_memregion_ro (const void ∗fdt, int domain_offset, void ∗opaque, int(∗fn)(const void ∗fdt, int domain_offset, int region_offset, u32 region_access, void ∗opaque))
- static int __fixup_find_domain_offset (void ∗fdt, int doff, void ∗p)
- static int __fixup_count_disable_devices (void ∗fdt, int doff, int roff, u32 perm, void ∗p)
- static int __fixup_disable_devices (void ∗fdt, int doff, int roff, u32 raccess, void ∗p)
- void fdt_domain_fixup (void ∗fdt)

- static int __fdt_parse_region (const void ∗fdt, int domain_offset, int region_offset, u32 region_access, void ∗opaque)
- static int __fdt_parse_domain (const void ∗fdt, int domain_offset, void ∗opaque)
- int fdt_domains_populate (const void ∗fdt)

30.279.1 Macro Definition Documentation

30.279.1.1 DISABLE_DEVICES_MASK

```
#define DISABLE_DEVICES_MASK
```

Value:

```
(SBI_DOMAIN_MEMREGION_READABLE | \
SBI_DOMAIN_MEMREGION_WRITEABLE | \
SBI_DOMAIN_MEMREGION_EXECUTABLE)
```

30.279.1.2 FDT_DOMAIN_REGION_MAX_COUNT

```
#define FDT_DOMAIN_REGION_MAX_COUNT 16
```

30.279.2 Function Documentation

30.279.2.1 __fdt_parse_domain()

```
static int __fdt_parse_domain (
            const void * fdt,
            int domain_offset,
            void * opaque )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.279.2.2 __fdt_parse_region()

```
static int __fdt_parse_region (
            const void * fdt,
            int domain_offset,
            int region_offset,
            u32 region_access,
            void * opaque )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.279.2.3 __fixup_count_disable_devices()

```
static int __fixup_count_disable_devices (
            void * fdt,
            int doff,
            int roff,
            u32 perm,
            void * p )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.279.2.4 __fixup_disable_devices()

```
static int __fixup_disable_devices (
            void * fdt,
            int doff,
            int roff,
            u32 raccess,
            void * p )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.279.2.5 __fixup_find_domain_offset()

```
static int __fixup_find_domain_offset (
            void * fdt,
            int doff,
            void * p )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.279.2.6 fdt_domain_fixup()

```
void fdt_domain_fixup (
            void * fdt )
```

Here is the call graph for this function:

30.279.2.7 fdt_domains_populate()

```
int fdt_domains_populate (
            const void * fdt )
```

Here is the call graph for this function:

30.279.2.8 fdt_iterate_each_domain()

```
int fdt_iterate_each_domain (
            void * fdt,
            void * opaque,
            int(*)(void *fdt, int domain_offset, void *opaque) fn )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.279.2.9 fdt_iterate_each_domain_ro()

```
static int fdt_iterate_each_domain_ro (
            const void * fdt,
            void * opaque,
            int(*)(const void *fdt, int domain_offset, void *opaque) fn )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.279.2.10 fdt_iterate_each_memregion()

```
int fdt_iterate_each_memregion (
            void * fdt,
            int domain_offset,
            void * opaque,
            int(*)(void *fdt, int domain_offset, int region_offset, u32 region_access, void
*opaque) fn )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.279.2.11 fdt_iterate_each_memregion_ro()

```
static int fdt_iterate_each_memregion_ro (
            const void * fdt,
            int domain_offset,
            void * opaque,
            int(*)(const void *fdt, int domain_offset, int region_offset, u32 region_access,
void *opaque) fn )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.280 lib/utils/fdt/fdt_driver.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_driver.h>
#include <sbi_utils/fdt/fdt_helper.h>
```

Include dependency graph for fdt_driver.c:

Functions

- int fdt_driver_init_by_offset (const void ∗fdt, int nodeoff, const struct fdt_driver ∗const ∗drivers)
- static int fdt_driver_init_scan (const void ∗fdt, const struct fdt_driver ∗const ∗drivers, bool one)
- int fdt_driver_init_all (const void ∗fdt, const struct fdt_driver ∗const ∗drivers)
- int fdt_driver_init_one (const void ∗fdt, const struct fdt_driver ∗const ∗drivers)

30.280.1 Function Documentation

30.280.1.1 fdt_driver_init_all()

```
int fdt_driver_init_all (
            const void * fdt,
            const struct fdt_driver *const * drivers )
```

Initialize a driver instance for each DT node that matches any of the provided drivers

Parameters

fdt	devicetree blob
drivers	NULL-terminated array of drivers to match against each node

Returns

0 if drivers for all matches (if any) were successfully initialized or a negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.280.1.2 fdt_driver_init_by_offset()

```
int fdt_driver_init_by_offset (
            const void * fdt,
```

```
int nodeoff,
const struct fdt_driver *const * drivers )
```

Initialize a driver instance for a specific DT node

Parameters

fdt	devicetree blob
nodeoff	offset of a node in the devicetree blob
drivers	NULL-terminated array of drivers to match against this node

Returns

0 if a driver was matched and successfully initialized or a negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.280.1.3 fdt_driver_init_one()

```
int fdt_driver_init_one (
```

```
            const void * fdt,
            const struct fdt_driver *const * drivers )
```

Initialize a driver instance for the first DT node that matches any of the provided drivers

Parameters

fdt	devicetree blob
drivers	NULL-terminated array of drivers to match against each node

Returns

0 if a driver was matched and successfully initialized or a negative error code on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.280.1.4 fdt_driver_init_scan()

```
static int fdt_driver_init_scan (
            const void * fdt,
            const struct fdt_driver *const * drivers,
            bool one )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.281 lib/utils/fdt/fdt_fixup.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_math.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_fixup.h>
#include <sbi_utils/fdt/fdt_pmu.h>
#include <sbi_utils/fdt/fdt_helper.h>
```

Include dependency graph for fdt_fixup.c:

Functions

- int fdt_add_cpu_idle_states (void ∗fdt, const struct sbi_cpu_idle_state ∗state)
- void fdt_cpu_fixup (void ∗fdt)

- static void fdt_domain_based_fixup_one (void ∗fdt, int nodeoff)
- static void fdt_fixup_node (void ∗fdt, const char ∗compatible)
- void fdt_aplic_fixup (void ∗fdt)
- void fdt_imsic_fixup (void ∗fdt)
- void fdt_plic_fixup (void ∗fdt)
- static int fdt_resv_memory_update_node (void ∗fdt, unsigned long addr, unsigned long size, int index, int parent)
- int fdt_reserved_memory_fixup (void ∗fdt)
- void fdt_config_fixup (void ∗fdt)
- static SBI_LIST_HEAD (fixup_list)
- int fdt_register_general_fixup (struct fdt_general_fixup ∗fixup)
- void fdt_unregister_general_fixup (struct fdt_general_fixup ∗fixup)
- void fdt_fixups (void ∗fdt)

30.281.1 Function Documentation

30.281.1.1 fdt_add_cpu_idle_states()

```
int fdt_add_cpu_idle_states (
            void * fdt,
            const struct sbi_cpu_idle_state * state )
```

Add CPU idle states to cpu nodes in the DT

Add information about CPU idle states to the devicetree. This function assumes that CPU idle states are not already present in the devicetree, and that all CPU states are equally applicable to all CPUs.

Parameters

fdt	device tree blob
states	array of idle state descriptions, ending with empty element

Returns

zero on success and -ve on failure

Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.2 fdt_aplic_fixup()

```
void fdt_aplic_fixup (
            void * fdt )
```

Fix up the APLIC nodes in the device tree

This routine disables APLIC nodes which are not accessible to the next booting stage based on currently assigned domain.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.3 fdt_config_fixup()

```
void fdt_config_fixup (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.4 fdt_cpu_fixup()

```
void fdt_cpu_fixup (
            void * fdt )
```

Fix up the CPU node in the device tree

This routine updates the "status" property of a CPU node in the device tree to "disabled" if that hart is in disabled state in OpenSBI.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

30.281.1.5 fdt_domain_based_fixup_one()

```
static void fdt_domain_based_fixup_one (
            void * fdt,
            int nodeoff )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.6 fdt_fixup_node()

```
static void fdt_fixup_node (
            void * fdt,
            const char * compatible )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.7 fdt_fixups()

```
void fdt_fixups (
            void * fdt )
```

General device tree fix-up

This routine do all required device tree fix-ups for a typical platform. It fixes up the PLIC node, IMSIC nodes, APLIC nodes, and the reserved memory node in the device tree by calling the corresponding helper routines to accomplish the task.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

30.281.1.8 fdt_imsic_fixup()

```
void fdt_imsic_fixup (
            void * fdt )
```

Fix up the IMSIC nodes in the device tree

This routine disables IMSIC nodes which are not accessible to the next booting stage based on currently assigned domain.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.9 fdt_plic_fixup()

```
void fdt_plic_fixup (
            void * fdt )
```

Fix up the PLIC node in the device tree

This routine updates the "interrupt-extended" property of the PLIC node in the device tree to hide the M-mode external interrupt from CPUs.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.10 fdt_register_general_fixup()

```
int fdt_register_general_fixup (
            struct fdt_general_fixup * fixup )
```

Register a general fixup Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.11 fdt_reserved_memory_fixup()

```
int fdt_reserved_memory_fixup (
            void * fdt )
```

We use PMP to protect OpenSBI firmware to safe-guard it from buggy S-mode software, see pmp_init() in lib/sbi/sbi_hart.c. The protected memory region information needs to be conveyed to S-mode software (e.g.↩ : operating system) via some well-known method.

With device tree, this can be done by inserting a child node of the reserved memory node which is used to specify one or more regions of reserved memory.

For the reserved memory node bindings, see Linux kernel documentation at Documentation/devicetree/bindings/reserved-memory/reserved-memory.txt

Some additional memory spaces may be protected by platform codes via PMP as well, and corresponding child nodes will be inserted. Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.12 fdt_resv_memory_update_node()

```
static int fdt_resv_memory_update_node (
            void * fdt,
            unsigned long addr,
            unsigned long size,
            int index,
            int parent )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.281.1.13 fdt_unregister_general_fixup()

```
void fdt_unregister_general_fixup (
            struct fdt_general_fixup * fixup )
```

UnRegister a general fixup Here is the call graph for this function:

30.281.1.14 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            fixup_list  )  [static]
```

30.282 lib/utils/fdt/fdt_helper.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_hart.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/aplic.h>
#include <sbi_utils/irqchip/imsic.h>
#include <sbi_utils/irqchip/plic.h>
```

Include dependency graph for fdt_helper.c:

Macros

- #define DEFAULT_UART_FREQ 0
- #define DEFAULT_UART_BAUD 115200
- #define DEFAULT_UART_REG_SHIFT 0
- #define DEFAULT_UART_REG_IO_WIDTH 1
- #define DEFAULT_UART_REG_OFFSET 0
- #define DEFAULT_RENESAS_SCIF_FREQ 100000000
- #define DEFAULT_RENESAS_SCIF_BAUD 115200
- #define DEFAULT_SIFIVE_UART_FREQ 0
- #define DEFAULT_SIFIVE_UART_BAUD 115200
- #define DEFAULT_SHAKTI_UART_FREQ 50000000
- #define DEFAULT_SHAKTI_UART_BAUD 115200

- #define RISCV_ISA_EXT_NAME_LEN_MAX 32
- #define set_multi_letter_ext(name, bit)

Functions

- const struct fdt_match * fdt_match_node (const void *fdt, int nodeoff, const struct fdt_match *match_table)
- int fdt_find_match (const void *fdt, int startoff, const struct fdt_match *match_table, const struct fdt_match **out_match)
- int fdt_parse_phandle_with_args (const void *fdt, int nodeoff, const char *prop, const char *cells_prop, int index, struct fdt_phandle_args *out_args)
- static int fdt_translate_address (const void *fdt, uint64_t reg, int parent, uint64_t *addr)
- int fdt_get_node_addr_size (const void *fdt, int node, int index, uint64_t *addr, uint64_t *size)
- int fdt_get_node_addr_size_by_name (const void *fdt, int node, const char *name, uint64_t *addr, uint64_t *size)
- bool fdt_node_is_enabled (const void *fdt, int nodeoff)
- int fdt_parse_hart_id (const void *fdt, int cpu_offset, u32 *hartid)
- int fdt_parse_max_enabled_hart_id (const void *fdt, u32 *max_hartid)
- int fdt_parse_timebase_frequency (const void *fdt, unsigned long *freq)
- static int fdt_parse_isa_one_hart (const char *isa, unsigned long *extensions)
- static void fdt_parse_isa_extensions_one_hart (const char *isa, unsigned long *extensions, int len)
- static int fdt_parse_isa_all_harts (const void *fdt)
- int fdt_parse_isa_extensions (const void *fdt, unsigned int hartid, unsigned long *extensions)
- static int fdt_parse_uart_node_common (const void *fdt, int nodeoffset, struct platform_uart_data *uart, unsigned long default_freq, unsigned long default_baud)
- int fdt_parse_gaisler_uart_node (const void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_renesas_scif_node (const void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_shakti_uart_node (const void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_sifive_uart_node (const void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_uart_node (const void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_uart8250 (const void *fdt, struct platform_uart_data *uart, const char *compatible)
- int fdt_parse_xlnx_uartlite_node (const void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_aplic_node (const void *fdt, int nodeoff, struct aplic_data *aplic)
- bool fdt_check_imsic_mlevel (const void *fdt)
- int fdt_parse_imsic_node (const void *fdt, int nodeoff, struct imsic_data *imsic)
- int fdt_parse_plic_node (const void *fdt, int nodeoffset, struct plic_data *plic)
- int fdt_parse_plic (const void *fdt, struct plic_data *plic, const char *compat)
- static int fdt_get_aclint_addr_size_by_name (const void *fdt, int nodeoffset, unsigned long *out_addr1, unsigned long *out_size1, unsigned long *out_addr2, unsigned long *out_size2)
- static int fdt_get_aclint_addr_size (const void *fdt, int nodeoffset, unsigned long *out_addr1, unsigned long *out_size1, unsigned long *out_addr2, unsigned long *out_size2)
- int fdt_parse_aclint_node (const void *fdt, int nodeoffset, bool for_timer, bool allow_regname, unsigned long *out_addr1, unsigned long *out_size1, unsigned long *out_addr2, unsigned long *out_size2, u32 *out_↩ first_hartid, u32 *out_hart_count)
- int fdt_parse_plmt_node (const void *fdt, int nodeoffset, unsigned long *plmt_base, unsigned long *plmt_↩ size, u32 *hart_count)
- int fdt_parse_plicsw_node (const void *fdt, int nodeoffset, unsigned long *plicsw_base, unsigned long *size, u32 *hart_count)
- int fdt_parse_compat_addr (const void *fdt, uint64_t *addr, const char *compatible)

Variables

- static unsigned long fdt_isa_bitmap_offset

30.282.1 Macro Definition Documentation

30.282.1.1 DEFAULT_RENESAS_SCIF_BAUD

```
#define DEFAULT_RENESAS_SCIF_BAUD 115200
```

30.282.1.2 DEFAULT_RENESAS_SCIF_FREQ

```
#define DEFAULT_RENESAS_SCIF_FREQ 100000000
```

30.282.1.3 DEFAULT_SHAKTI_UART_BAUD

```
#define DEFAULT_SHAKTI_UART_BAUD 115200
```

30.282.1.4 DEFAULT_SHAKTI_UART_FREQ

```
#define DEFAULT_SHAKTI_UART_FREQ 50000000
```

30.282.1.5 DEFAULT_SIFIVE_UART_BAUD

```
#define DEFAULT_SIFIVE_UART_BAUD 115200
```

30.282.1.6 DEFAULT_SIFIVE_UART_FREQ

```
#define DEFAULT_SIFIVE_UART_FREQ 0
```

30.282.1.7 DEFAULT_UART_BAUD

```
#define DEFAULT_UART_BAUD 115200
```

30.282.1.8 DEFAULT_UART_FREQ

```
#define DEFAULT_UART_FREQ 0
```

30.282.1.9 DEFAULT_UART_REG_IO_WIDTH

```
#define DEFAULT_UART_REG_IO_WIDTH 1
```

30.282.1.10 DEFAULT_UART_REG_OFFSET

```
#define DEFAULT_UART_REG_OFFSET 0
```

30.282.1.11 DEFAULT_UART_REG_SHIFT

```
#define DEFAULT_UART_REG_SHIFT 0
```

30.282.1.12 RISCV_ISA_EXT_NAME_LEN_MAX

```
#define RISCV_ISA_EXT_NAME_LEN_MAX 32
```

30.282.1.13 set_multi_letter_ext

```
#define set_multi_letter_ext(
            name,
            bit )
```

Value:

```
        if (!strcmp(mstr, name)) {        \
            __set_bit(bit, extensions); \
            continue;           \
        }
```

30.282.2 Function Documentation

30.282.2.1 fdt_check_imsic_mlevel()

```
bool fdt_check_imsic_mlevel (
            const void * fdt )
```

Here is the call graph for this function:

30.282.2.2 fdt_find_match()

```
int fdt_find_match (
            const void * fdt,
            int startoff,
            const struct fdt_match * match_table,
            const struct fdt_match ** out_match )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.3 fdt_get_aclint_addr_size()

```
static int fdt_get_aclint_addr_size (
            const void * fdt,
            int nodeoffset,
            unsigned long * out_addr1,
            unsigned long * out_size1,
            unsigned long * out_addr2,
            unsigned long * out_size2 )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.4 fdt_get_aclint_addr_size_by_name()

```
static int fdt_get_aclint_addr_size_by_name (
            const void * fdt,
            int nodeoffset,
            unsigned long * out_addr1,
            unsigned long * out_size1,
            unsigned long * out_addr2,
            unsigned long * out_size2 )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.5 fdt_get_node_addr_size()

```
int fdt_get_node_addr_size (
            const void * fdt,
            int node,
            int index,
            uint64_t * addr,
            uint64_t * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.6 fdt_get_node_addr_size_by_name()

```
int fdt_get_node_addr_size_by_name (
            const void * fdt,
            int node,
            const char * name,
            uint64_t * addr,
            uint64_t * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.7 fdt_match_node()

```
const struct fdt_match * fdt_match_node (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match_table )
```

Here is the call graph for this function:

30.282.2.8 fdt_node_is_enabled()

```
bool fdt_node_is_enabled (
            const void * fdt,
            int nodeoff )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.9 fdt_parse_aclint_node()

```
int fdt_parse_aclint_node (
            const void * fdt,
            int nodeoffset,
            bool for_timer,
            bool allow_regname,
            unsigned long * out_addr1,
            unsigned long * out_size1,
            unsigned long * out_addr2,
            unsigned long * out_size2,
            u32 * out_first_hartid,
            u32 * out_hart_count )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.10 fdt_parse_aplic_node()

```
int fdt_parse_aplic_node (
            const void * fdt,
            int nodeoff,
            struct aplic_data * aplic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.11 fdt_parse_compat_addr()

```
int fdt_parse_compat_addr (
            const void * fdt,
            uint64_t * addr,
            const char * compatible )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.12 fdt_parse_gaisler_uart_node()

```
int fdt_parse_gaisler_uart_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.13 fdt_parse_hart_id()

```
int fdt_parse_hart_id (
            const void * fdt,
            int cpu_offset,
            u32 * hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.14 fdt_parse_imsic_node()

```
int fdt_parse_imsic_node (
            const void * fdt,
            int nodeoff,
            struct imsic_data * imsic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.15 fdt_parse_isa_all_harts()

```
static int fdt_parse_isa_all_harts (
            const void * fdt )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.16 fdt_parse_isa_extensions()

```
int fdt_parse_isa_extensions (
            const void * fdt,
```

```
            unsigned int hartid,
            unsigned long * extensions )
```

Here is the call graph for this function:

30.282.2.17 fdt_parse_isa_extensions_one_hart()

```
static void fdt_parse_isa_extensions_one_hart (
            const char * isa,
            unsigned long * extensions,
            int len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.18 fdt_parse_isa_one_hart()

```
static int fdt_parse_isa_one_hart (
            const char * isa,
            unsigned long * extensions )  [static]
```

Here is the caller graph for this function:

30.282.2.19 fdt_parse_max_enabled_hart_id()

```
int fdt_parse_max_enabled_hart_id (
            const void * fdt,
            u32 * max_hartid )
```

Here is the call graph for this function:

30.282.2.20 fdt_parse_phandle_with_args()

```
int fdt_parse_phandle_with_args (
            const void * fdt,
            int nodeoff,
            const char * prop,
            const char * cells_prop,
            int index,
            struct fdt_phandle_args * out_args )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.21 fdt_parse_plic()

```
int fdt_parse_plic (
```

```
                const void * fdt,
                struct plic_data * plic,
                const char * compat )
```

Here is the call graph for this function:

30.282.2.22 fdt_parse_plic_node()

```
int fdt_parse_plic_node (
                const void * fdt,
                int nodeoffset,
                struct plic_data * plic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.23 fdt_parse_plicsw_node()

```
int fdt_parse_plicsw_node (
                const void * fdt,
```

```
int nodeoffset,
unsigned long * plicsw_base,
unsigned long * size,
u32 * hart_count )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.24 fdt_parse_plmt_node()

```
int fdt_parse_plmt_node (
            const void * fdt,
            int nodeoffset,
            unsigned long * plmt_base,
            unsigned long * plmt_size,
            u32 * hart_count )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.25 fdt_parse_renesas_scif_node()

```
int fdt_parse_renesas_scif_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.26 fdt_parse_shakti_uart_node()

```
int fdt_parse_shakti_uart_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.27 fdt_parse_sifive_uart_node()

```
int fdt_parse_sifive_uart_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.28 fdt_parse_timebase_frequency()

```
int fdt_parse_timebase_frequency (
            const void * fdt,
            unsigned long * freq )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.29 fdt_parse_uart8250()

```
int fdt_parse_uart8250 (
            const void * fdt,
            struct platform_uart_data * uart,
            const char * compatible )
```

Here is the call graph for this function:

30.282.2.30 fdt_parse_uart_node()

```
int fdt_parse_uart_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.31 fdt_parse_uart_node_common()

```
static int fdt_parse_uart_node_common (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart,
            unsigned long default_freq,
            unsigned long default_baud )  [static]
```

UART address is mandatory. clock-frequency and current-speed may not be present. Don't return error.Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.32 fdt_parse_xlnx_uartlite_node()

```
int fdt_parse_xlnx_uartlite_node (
            const void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.2.33 fdt_translate_address()

```
static int fdt_translate_address (
            const void * fdt,
```

```
uint64_t reg,
int parent,
uint64_t * addr )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.282.3 Variable Documentation

30.282.3.1 fdt_isa_bitmap_offset

```
unsigned long fdt_isa_bitmap_offset  [static]
```

30.283 lib/utils/fdt/fdt_pmu.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_error.h>
```

```
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/fdt/fdt_pmu.h>
```

Include dependency graph for fdt_pmu.c:

Macros

- #define FDT_PMU_HW_EVENT_MAX (SBI_PMU_HW_EVENT_MAX ∗ 2)

Functions

- uint64_t fdt_pmu_get_select_value (uint32_t event_idx)
- int fdt_pmu_fixup (void ∗fdt)
- int fdt_pmu_setup (const void ∗fdt)

Variables

- struct fdt_pmu_hw_event_select_map fdt_pmu_evt_select [FDT_PMU_HW_EVENT_MAX] = {0}
- uint32_t hw_event_count

30.283.1 Macro Definition Documentation

30.283.1.1 FDT_PMU_HW_EVENT_MAX

```
#define FDT_PMU_HW_EVENT_MAX (SBI_PMU_HW_EVENT_MAX * 2)
```

30.283.2 Function Documentation

30.283.2.1 fdt_pmu_fixup()

```
int fdt_pmu_fixup (
            void * fdt )
```

Here is the call graph for this function:

30.283.2.2 fdt_pmu_get_select_value()

```
uint64_t fdt_pmu_get_select_value (
            uint32_t event_idx )
```

30.283.2.3 fdt_pmu_setup()

```
int fdt_pmu_setup (
            const void * fdt )
```

Here is the call graph for this function:

30.283.3 Variable Documentation

30.283.3.1 fdt_pmu_evt_select

```
struct fdt_pmu_hw_event_select_map fdt_pmu_evt_select[FDT_PMU_HW_EVENT_MAX] = {0}
```

30.283.3.2 hw_event_count

```
uint32_t hw_event_count
```

30.284 lib/utils/gpio/fdt_gpio.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
```

Include dependency graph for fdt_gpio.c:

Functions

- static int fdt_gpio_init (const void ∗fdt, int nodeoff)
- static int fdt_gpio_chip_find (const void ∗fdt, int nodeoff, struct gpio_chip ∗∗out_chip)
- int fdt_gpio_pin_get (const void ∗fdt, int nodeoff, int index, struct gpio_pin ∗out_pin)
- int fdt_gpio_simple_xlate (struct gpio_chip ∗chip, const struct fdt_phandle_args ∗pargs, struct gpio_pin ∗out_pin)

Variables

- const struct fdt_driver *const fdt_gpio_drivers [ ]

30.284.1 Function Documentation

30.284.1.1 fdt_gpio_chip_find()

```
static int fdt_gpio_chip_find (
            const void * fdt,
            int nodeoff,
            struct gpio_chip ** out_chip )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.284.1.2 fdt_gpio_init()

```
static int fdt_gpio_init (
            const void * fdt,
            int nodeoff )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.284.1.3 fdt_gpio_pin_get()

```
int fdt_gpio_pin_get (
            const void * fdt,
            int nodeoff,
            int index,
            struct gpio_pin * out_pin )
```

Get a GPIO pin using "gpios" DT property of client DT node Here is the call graph for this function:

Here is the caller graph for this function:

30.284.1.4 fdt_gpio_simple_xlate()

```
int fdt_gpio_simple_xlate (
            struct gpio_chip * chip,
            const struct fdt_phandle_args * pargs,
            struct gpio_pin * out_pin )
```

Simple xlate function to convert two GPIO FDT cells into GPIO pin

30.284.2 Variable Documentation

30.284.2.1 fdt_gpio_drivers

```
const struct fdt_driver* const fdt_gpio_drivers[] [extern]
```

30.285 lib/utils/gpio/fdt_gpio_designware.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
```

Include dependency graph for fdt_gpio_designware.c:

Macros

- #define DW_GPIO_PINS_MAX 32
- #define DW_GPIO_DDR 0x4
- #define DW_GPIO_DR 0x0
- #define DW_GPIO_BIT(_b) (1UL << (_b))
- #define pin_to_chip(__p) container_of((__p)->chip, struct dw_gpio_chip, chip);

Functions

- static int dw_gpio_direction_output (struct gpio_pin *gp, int value)
- static void dw_gpio_set (struct gpio_pin *gp, int value)
- static int dw_gpio_init_bank (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- const struct fdt_gpio fdt_gpio_designware
- static const struct fdt_match dw_gpio_match [ ]

30.285.1 Macro Definition Documentation

30.285.1.1 DW_GPIO_BIT

```
#define DW_GPIO_BIT(
            _b ) (1UL << (_b))
```

30.285.1.2 DW_GPIO_DDR

```
#define DW_GPIO_DDR 0x4
```

30.285.1.3 DW_GPIO_DR

```
#define DW_GPIO_DR 0x0
```

30.285.1.4 DW_GPIO_PINS_MAX

```
#define DW_GPIO_PINS_MAX 32
```

30.285.1.5 pin_to_chip

```
#define pin_to_chip(
            __p ) container_of((__p)->chip, struct dw_gpio_chip, chip);
```

30.285.2 Function Documentation

30.285.2.1 dw_gpio_direction_output()

```
static int dw_gpio_direction_output (
            struct gpio_pin * gp,
            int value )  [static]
```

Here is the caller graph for this function:

30.285.2.2 dw_gpio_init_bank()

```
static int dw_gpio_init_bank (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.285.2.3 dw_gpio_set()

```
static void dw_gpio_set (
            struct gpio_pin * gp,
            int value )  [static]
```

Here is the caller graph for this function:

30.285.3 Variable Documentation

30.285.3.1 dw_gpio_match

```
const struct fdt_match dw_gpio_match[] [static]
```

Initial value:

```
= {
    { .compatible = "snps,dw-apb-gpio-port" },
    { },
}
```

30.285.3.2 fdt_gpio_designware

```
const struct fdt_gpio fdt_gpio_designware
```

Initial value:

```
= {
    .driver = {
        .match_table = dw_gpio_match,
        .init = dw_gpio_init_bank,
    },
    .xlate = fdt_gpio_simple_xlate,
}
```

30.286 lib/utils/gpio/fdt_gpio_sifive.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
```

Include dependency graph for fdt_gpio_sifive.c:

Macros

- #define SIFIVE_GPIO_PINS_MIN 1
- #define SIFIVE_GPIO_PINS_MAX 32
- #define SIFIVE_GPIO_PINS_DEF 16
- #define SIFIVE_GPIO_OUTEN 0x8
- #define SIFIVE_GPIO_OUTVAL 0xc
- #define SIFIVE_GPIO_BIT(b) (1U << (b))

Functions

- static int sifive_gpio_direction_output (struct gpio_pin *gp, int value)
- static void sifive_gpio_set (struct gpio_pin *gp, int value)
- static int sifive_gpio_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- const struct fdt_gpio fdt_gpio_sifive
- static const struct fdt_match sifive_gpio_match [ ]

30.286.1 Macro Definition Documentation

30.286.1.1 SIFIVE_GPIO_BIT

```
#define SIFIVE_GPIO_BIT(
            b ) (1U << (b))
```

30.286.1.2 SIFIVE_GPIO_OUTEN

```
#define SIFIVE_GPIO_OUTEN 0x8
```

30.286.1.3 SIFIVE_GPIO_OUTVAL

```
#define SIFIVE_GPIO_OUTVAL 0xc
```

30.286.1.4 SIFIVE_GPIO_PINS_DEF

```
#define SIFIVE_GPIO_PINS_DEF 16
```

30.286.1.5 SIFIVE_GPIO_PINS_MAX

```
#define SIFIVE_GPIO_PINS_MAX 32
```

30.286.1.6 SIFIVE_GPIO_PINS_MIN

```
#define SIFIVE_GPIO_PINS_MIN 1
```

30.286.2 Function Documentation

30.286.2.1 sifive_gpio_direction_output()

```
static int sifive_gpio_direction_output (
            struct gpio_pin * gp,
            int value )  [static]
```

Here is the caller graph for this function:

30.286.2.2 sifive_gpio_init()

```
static int sifive_gpio_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.286.2.3 sifive_gpio_set()

```
static void sifive_gpio_set (
            struct gpio_pin * gp,
            int value )  [static]
```

Here is the caller graph for this function:

30.286.3 Variable Documentation

30.286.3.1 fdt_gpio_sifive

```
const struct fdt_gpio fdt_gpio_sifive
```

Initial value:

```
= {
    .driver = {
        .match_table = sifive_gpio_match,
        .init = sifive_gpio_init,
    },
    .xlate = fdt_gpio_simple_xlate,
}
```

30.286.3.2 sifive_gpio_match

```
const struct fdt_match sifive_gpio_match[] [static]
```

Initial value:

```
= {
    { .compatible = "sifive,gpio0" },
    { },
}
```

30.287 lib/utils/gpio/fdt_gpio_starfive.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
```

Include dependency graph for fdt_gpio_starfive.c:

Macros

- #define STARFIVE_GPIO_PINS_DEF 64
- #define STARFIVE_GPIO_OUTVAL 0x40
- #define STARFIVE_GPIO_MASK 0xff
- #define STARFIVE_GPIO_REG_SHIFT_MASK 0x3
- #define STARFIVE_GPIO_SHIFT_BITS 0x3

Functions

- static int starfive_gpio_direction_output (struct gpio_pin *gp, int value)
- static void starfive_gpio_set (struct gpio_pin *gp, int value)
- static int starfive_gpio_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- const struct fdt_gpio fdt_gpio_starfive
- static const struct fdt_match starfive_gpio_match [ ]

30.287.1 Macro Definition Documentation

30.287.1.1 STARFIVE_GPIO_MASK

```
#define STARFIVE_GPIO_MASK 0xff
```

30.287.1.2 STARFIVE_GPIO_OUTVAL

```
#define STARFIVE_GPIO_OUTVAL 0x40
```

30.287.1.3 STARFIVE_GPIO_PINS_DEF

```
#define STARFIVE_GPIO_PINS_DEF 64
```

30.287.1.4 STARFIVE_GPIO_REG_SHIFT_MASK

```
#define STARFIVE_GPIO_REG_SHIFT_MASK 0x3
```

30.287.1.5 STARFIVE_GPIO_SHIFT_BITS

```
#define STARFIVE_GPIO_SHIFT_BITS 0x3
```

30.287.2 Function Documentation

30.287.2.1 starfive_gpio_direction_output()

```
static int starfive_gpio_direction_output (
            struct gpio_pin * gp,
            int value )  [static]
```

Here is the caller graph for this function:

30.287.2.2 starfive_gpio_init()

```
static int starfive_gpio_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.287.2.3 starfive_gpio_set()

```
static void starfive_gpio_set (
            struct gpio_pin * gp,
            int value )  [static]
```

Here is the caller graph for this function:

30.287.3 Variable Documentation

30.287.3.1 fdt_gpio_starfive

```
const struct fdt_gpio fdt_gpio_starfive
```

Initial value:

```
= {
    .driver = {
        .match_table = starfive_gpio_match,
        .init = starfive_gpio_init,
    },
    .xlate = fdt_gpio_simple_xlate,
}
```

30.287.3.2 starfive_gpio_match

```
const struct fdt_match starfive_gpio_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "starfive,jh7110-sys-pinctrl" },
    { .compatible = "starfive,iomux-pinctrl" },
    { },
}
```

30.288 lib/utils/gpio/gpio.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/gpio/gpio.h>
```

Include dependency graph for gpio.c:

Functions

- static SBI_LIST_HEAD (gpio_chip_list)
- struct gpio_chip ∗ gpio_chip_find (unsigned int id)
- int gpio_chip_add (struct gpio_chip ∗gc)
- void gpio_chip_remove (struct gpio_chip ∗gc)
- int gpio_get_direction (struct gpio_pin ∗gp)
- int gpio_direction_input (struct gpio_pin ∗gp)
- int gpio_direction_output (struct gpio_pin ∗gp, int value)
- int gpio_get (struct gpio_pin ∗gp)
- int gpio_set (struct gpio_pin ∗gp, int value)

30.288.1 Function Documentation

30.288.1.1 gpio_chip_add()

```
int gpio_chip_add (
            struct gpio_chip * gc )
```

Register GPIO chip Here is the call graph for this function:

Here is the caller graph for this function:

30.288.1.2 gpio_chip_find()

```
struct gpio_chip * gpio_chip_find (
            unsigned int id )
```

Find a registered GPIO chip Here is the call graph for this function:

Here is the caller graph for this function:

30.288.1.3 gpio_chip_remove()

```
void gpio_chip_remove (
            struct gpio_chip * gc )
```

Un-register GPIO chip Here is the call graph for this function:

30.288.1.4 gpio_direction_input()

```
int gpio_direction_input (
            struct gpio_pin * gp )
```

Set input direction of GPIO pin Here is the caller graph for this function:

30.288.1.5 gpio_direction_output()

```
int gpio_direction_output (
            struct gpio_pin * gp,
            int value )
```

Set output direction of GPIO pin Here is the caller graph for this function:

30.288.1.6 gpio_get()

```
int gpio_get (
            struct gpio_pin * gp )
```

Get current value of GPIO pin

30.288.1.7 gpio_get_direction()

```
int gpio_get_direction (
            struct gpio_pin * gp )
```

Get current direction of GPIO pin

30.288.1.8 gpio_set()

```
int gpio_set (
            struct gpio_pin * gp,
            int value )
```

Set output value of GPIO pin Here is the caller graph for this function:

30.288.1.9 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            gpio_chip_list ) [static]
```

30.289 lib/utils/hsm/fdt_hsm.c File Reference

`#include <sbi_utils/hsm/fdt_hsm.h>`

Include dependency graph for fdt_hsm.c:

Functions

- void fdt_hsm_init (const void *fdt)

Variables

- const struct fdt_driver *const fdt_hsm_drivers [ ]

30.289.1 Function Documentation

30.289.1.1 fdt_hsm_init()

```
void fdt_hsm_init (
            const void * fdt )
```

Here is the call graph for this function:

30.289.2 Variable Documentation

30.289.2.1 fdt_hsm_drivers

```
const struct fdt_driver* const fdt_hsm_drivers[] [extern]
```

30.290 lib/utils/hsm/fdt_hsm_rpmi.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/fdt/fdt_fixup.h>
#include <sbi_utils/hsm/fdt_hsm.h>
#include <sbi_utils/mailbox/fdt_mailbox.h>
#include <sbi_utils/mailbox/mailbox.h>
#include <sbi_utils/mailbox/rpmi_mailbox.h>
```

Include dependency graph for fdt_hsm_rpmi.c:

Macros

- #define MAX_HSM_SUPSEND_STATE_NAMELEN 16

Functions

- static struct rpmi_hsm * rpmi_hsm_get_pointer (u32 hartid)
- static int rpmi_hsm_start (u32 hartid, ulong resume_addr)
- static int rpmi_hsm_stop (void)
- static bool is_rpmi_hsm_susp_supported (struct rpmi_hsm_suspend *susp, u32 type)
- static int rpmi_hsm_suspend (u32 type, ulong resume_addr)
- static void rpmi_hsm_do_fixup (struct fdt_general_fixup *f, void *fdt)
- static int rpmi_hsm_get_num_suspend_states (struct mbox_chan *chan, struct rpmi_hsm_suspend *susp)
- static int rpmi_hsm_get_suspend_states (struct mbox_chan *chan, struct rpmi_hsm_suspend *susp)
- static int rpmi_hsm_update_hart_scratch (struct mbox_chan *chan, struct rpmi_hsm_suspend *susp)
- static int rpmi_hsm_cold_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static unsigned long rpmi_hsm_offset
- static struct sbi_hsm_device sbi_hsm_rpmi
- static struct fdt_general_fixup rpmi_hsm_fixup
- static const struct fdt_match rpmi_hsm_match [ ]
- struct fdt_driver fdt_hsm_rpmi

30.290.1 Macro Definition Documentation

30.290.1.1 MAX_HSM_SUPSEND_STATE_NAMELEN

```
#define MAX_HSM_SUPSEND_STATE_NAMELEN 16
```

30.290.2 Function Documentation

30.290.2.1 is_rpmi_hsm_susp_supported()

```
static bool is_rpmi_hsm_susp_supported (
            struct rpmi_hsm_suspend * susp,
            u32 type )  [static]
```

Here is the caller graph for this function:

30.290.2.2 rpmi_hsm_cold_init()

```
static int rpmi_hsm_cold_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.290.2.3 rpmi_hsm_do_fixup()

```
static void rpmi_hsm_do_fixup (
            struct fdt_general_fixup * f,
            void * fdt )  [static]
```

Here is the call graph for this function:

30.290.2.4 rpmi_hsm_get_num_suspend_states()

```
static int rpmi_hsm_get_num_suspend_states (
            struct mbox_chan * chan,
            struct rpmi_hsm_suspend * susp )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.290.2.5 rpmi_hsm_get_pointer()

```
static struct rpmi_hsm * rpmi_hsm_get_pointer (
            u32 hartid )  [static]
```

Here is the caller graph for this function:

30.290.2.6 rpmi_hsm_get_suspend_states()

```
static int rpmi_hsm_get_suspend_states (
            struct mbox_chan * chan,
            struct rpmi_hsm_suspend * susp )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.290.2.7 rpmi_hsm_start()

```
static int rpmi_hsm_start (
            u32 hartid,
            ulong resume_addr )  [static]
```

Here is the call graph for this function:

30.290.2.8 rpmi_hsm_stop()

```
static int rpmi_hsm_stop (
            void  )  [static]
```

Here is the call graph for this function:

30.290.2.9 rpmi_hsm_suspend()

```
static int rpmi_hsm_suspend (
            u32 type,
            ulong resume_addr )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.290.2.10 rpmi_hsm_update_hart_scratch()

```
static int rpmi_hsm_update_hart_scratch (
            struct mbox_chan * chan,
            struct rpmi_hsm_suspend * susp )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.290.3 Variable Documentation

30.290.3.1 fdt_hsm_rpmi

```
struct fdt_driver fdt_hsm_rpmi
```

Initial value:

```
= {
    .match_table = rpmi_hsm_match,
    .init = rpmi_hsm_cold_init,
    .experimental = true,
}
```

30.290.3.2 rpmi_hsm_fixup

```
struct fdt_general_fixup rpmi_hsm_fixup  [static]
```

Initial value:

```
= {
    .name = "rpmi-hsm-fixup",
    .do_fixup = rpmi_hsm_do_fixup,
}
```

30.290.3.3 rpmi_hsm_match

```
const struct fdt_match rpmi_hsm_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,rpmi-hsm" },
    {},
}
```

30.290.3.4 rpmi_hsm_offset

```
unsigned long rpmi_hsm_offset  [static]
```

30.290.3.5 sbi_hsm_rpmi

```
struct sbi_hsm_device sbi_hsm_rpmi  [static]
```

Initial value:

```
= {
    .name       = "rpmi-hsm",
    .hart_start = rpmi_hsm_start,
    .hart_stop  = rpmi_hsm_stop,
    .hart_suspend   = rpmi_hsm_suspend,
}
```

30.291 lib/utils/i2c/dw_i2c.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_bitops.h>
#include <sbi_utils/i2c/dw_i2c.h>
```

Include dependency graph for dw_i2c.c:

Macros

- #define DW_IC_CON 0x00
- #define DW_IC_TAR 0x04
- #define DW_IC_SAR 0x08
- #define DW_IC_DATA_CMD 0x10
- #define DW_IC_SS_SCL_HCNT 0x14
- #define DW_IC_SS_SCL_LCNT 0x18
- #define DW_IC_FS_SCL_HCNT 0x1c
- #define DW_IC_FS_SCL_LCNT 0x20
- #define DW_IC_HS_SCL_HCNT 0x24
- #define DW_IC_HS_SCL_LCNT 0x28
- #define DW_IC_INTR_STAT 0x2c
- #define DW_IC_INTR_MASK 0x30
- #define DW_IC_RAW_INTR_STAT 0x34
- #define DW_IC_RX_TL 0x38
- #define DW_IC_TX_TL 0x3c
- #define DW_IC_CLR_INTR 0x40
- #define DW_IC_CLR_RX_UNDER 0x44
- #define DW_IC_CLR_RX_OVER 0x48
- #define DW_IC_CLR_TX_OVER 0x4c
- #define DW_IC_CLR_RD_REQ 0x50
- #define DW_IC_CLR_TX_ABRT 0x54
- #define DW_IC_CLR_RX_DONE 0x58
- #define DW_IC_CLR_ACTIVITY 0x5c
- #define DW_IC_CLR_STOP_DET 0x60
- #define DW_IC_CLR_START_DET 0x64
- #define DW_IC_CLR_GEN_CALL 0x68
- #define DW_IC_ENABLE 0x6c
- #define DW_IC_STATUS 0x70
- #define DW_IC_TXFLR 0x74
- #define DW_IC_RXFLR 0x78
- #define DW_IC_SDA_HOLD 0x7c
- #define DW_IC_TX_ABRT_SOURCE 0x80
- #define DW_IC_ENABLE_STATUS 0x9c
- #define DW_IC_CLR_RESTART_DET 0xa8
- #define DW_IC_COMP_PARAM_1 0xf4
- #define DW_IC_COMP_VERSION 0xf8
- #define DW_I2C_STATUS_TXFIFO_EMPTY BIT(2)
- #define DW_I2C_STATUS_RXFIFO_NOT_EMPTY BIT(3)

- #define IC_DATA_CMD_READ BIT(8)
- #define IC_DATA_CMD_STOP BIT(9)
- #define IC_DATA_CMD_RESTART BIT(10)
- #define IC_INT_STATUS_STOPDET BIT(9)
- #define dw_i2c_adapter_poll_rxrdy(adap) dw_i2c_adapter_poll(adap, DW_I2C_STATUS_RXFIFO_NOT_EMPTY, DW_IC_STATUS, 0)
- #define dw_i2c_adapter_poll_txfifo_ready(adap) dw_i2c_adapter_poll(adap, DW_I2C_STATUS_TXFIFO_EMPTY, DW_IC_STATUS, 0)

Functions

- static void dw_i2c_setreg (struct dw_i2c_adapter *adap, u8 reg, u32 value)
- static u32 dw_i2c_getreg (struct dw_i2c_adapter *adap, u32 reg)
- static int dw_i2c_adapter_poll (struct dw_i2c_adapter *adap, u32 mask, u32 addr, bool inverted)
- static int dw_i2c_write_addr (struct dw_i2c_adapter *adap, u8 addr)
- static int dw_i2c_adapter_read (struct i2c_adapter *ia, u8 addr, u8 reg, u8 *buffer, int len)
- static int dw_i2c_adapter_write (struct i2c_adapter *ia, u8 addr, u8 reg, u8 *buffer, int len)
- int dw_i2c_init (struct i2c_adapter *adapter, int nodeoff)

30.291.1 Macro Definition Documentation

30.291.1.1 dw_i2c_adapter_poll_rxrdy

```
#define dw_i2c_adapter_poll_rxrdy(
            adap )  dw_i2c_adapter_poll(adap, DW_I2C_STATUS_RXFIFO_NOT_EMPTY, DW_IC_STATUS,
0)
```

30.291.1.2 dw_i2c_adapter_poll_txfifo_ready

```
#define dw_i2c_adapter_poll_txfifo_ready(
            adap )  dw_i2c_adapter_poll(adap, DW_I2C_STATUS_TXFIFO_EMPTY, DW_IC_STATUS, 0)
```

30.291.1.3 DW_I2C_STATUS_RXFIFO_NOT_EMPTY

```
#define DW_I2C_STATUS_RXFIFO_NOT_EMPTY BIT(3)
```

30.291.1.4 DW_I2C_STATUS_TXFIFO_EMPTY

```
#define DW_I2C_STATUS_TXFIFO_EMPTY BIT(2)
```

30.291.1.5 DW_IC_CLR_ACTIVITY

```
#define DW_IC_CLR_ACTIVITY 0x5c
```

30.291.1.6 DW_IC_CLR_GEN_CALL

```
#define DW_IC_CLR_GEN_CALL 0x68
```

30.291.1.7 DW_IC_CLR_INTR

```
#define DW_IC_CLR_INTR 0x40
```

30.291.1.8 DW_IC_CLR_RD_REQ

```
#define DW_IC_CLR_RD_REQ 0x50
```

30.291.1.9 DW_IC_CLR_RESTART_DET

```
#define DW_IC_CLR_RESTART_DET 0xa8
```

30.291.1.10 DW_IC_CLR_RX_DONE

```
#define DW_IC_CLR_RX_DONE 0x58
```

30.291.1.11 DW_IC_CLR_RX_OVER

```
#define DW_IC_CLR_RX_OVER 0x48
```

30.291.1.12 DW_IC_CLR_RX_UNDER

```
#define DW_IC_CLR_RX_UNDER 0x44
```

30.291.1.13 DW_IC_CLR_START_DET

```
#define DW_IC_CLR_START_DET 0x64
```

30.291.1.14 DW_IC_CLR_STOP_DET

```
#define DW_IC_CLR_STOP_DET 0x60
```

30.291.1.15 DW_IC_CLR_TX_ABRT

```
#define DW_IC_CLR_TX_ABRT 0x54
```

30.291.1.16 DW_IC_CLR_TX_OVER

```
#define DW_IC_CLR_TX_OVER 0x4c
```

30.291.1.17 DW_IC_COMP_PARAM_1

```
#define DW_IC_COMP_PARAM_1 0xf4
```

30.291.1.18 DW_IC_COMP_VERSION

```
#define DW_IC_COMP_VERSION 0xf8
```

30.291.1.19 DW_IC_CON

```
#define DW_IC_CON 0x00
```

30.291.1.20 DW_IC_DATA_CMD

```
#define DW_IC_DATA_CMD 0x10
```

30.291.1.21 DW_IC_ENABLE

```
#define DW_IC_ENABLE 0x6c
```

30.291.1.22 DW_IC_ENABLE_STATUS

```
#define DW_IC_ENABLE_STATUS 0x9c
```

30.291.1.23 DW_IC_FS_SCL_HCNT

```
#define DW_IC_FS_SCL_HCNT 0x1c
```

30.291.1.24 DW_IC_FS_SCL_LCNT

```
#define DW_IC_FS_SCL_LCNT 0x20
```

30.291.1.25 DW_IC_HS_SCL_HCNT

```
#define DW_IC_HS_SCL_HCNT 0x24
```

30.291.1.26 DW_IC_HS_SCL_LCNT

```
#define DW_IC_HS_SCL_LCNT 0x28
```

30.291.1.27 DW_IC_INTR_MASK

```
#define DW_IC_INTR_MASK 0x30
```

30.291.1.28 DW_IC_INTR_STAT

```
#define DW_IC_INTR_STAT 0x2c
```

30.291.1.29 DW_IC_RAW_INTR_STAT

```
#define DW_IC_RAW_INTR_STAT 0x34
```

30.291.1.30 DW_IC_RX_TL

```
#define DW_IC_RX_TL 0x38
```

30.291.1.31 DW_IC_RXFLR

```
#define DW_IC_RXFLR 0x78
```

30.291.1.32 DW_IC_SAR

```
#define DW_IC_SAR 0x08
```

30.291.1.33 DW_IC_SDA_HOLD

```
#define DW_IC_SDA_HOLD 0x7c
```

30.291.1.34 DW_IC_SS_SCL_HCNT

```
#define DW_IC_SS_SCL_HCNT 0x14
```

30.291.1.35 DW_IC_SS_SCL_LCNT

```
#define DW_IC_SS_SCL_LCNT 0x18
```

30.291.1.36 DW_IC_STATUS

```
#define DW_IC_STATUS 0x70
```

30.291.1.37 DW_IC_TAR

```
#define DW_IC_TAR 0x04
```

30.291.1.38 DW_IC_TX_ABRT_SOURCE

```
#define DW_IC_TX_ABRT_SOURCE 0x80
```

30.291.1.39 DW_IC_TX_TL

```
#define DW_IC_TX_TL 0x3c
```

30.291.1.40 DW_IC_TXFLR

```
#define DW_IC_TXFLR 0x74
```

30.291.1.41 IC_DATA_CMD_READ

```
#define IC_DATA_CMD_READ BIT(8)
```

30.291.1.42 IC_DATA_CMD_RESTART

```
#define IC_DATA_CMD_RESTART BIT(10)
```

30.291.1.43 IC_DATA_CMD_STOP

```
#define IC_DATA_CMD_STOP BIT(9)
```

30.291.1.44 IC_INT_STATUS_STOPDET

```
#define IC_INT_STATUS_STOPDET BIT(9)
```

30.291.2 Function Documentation

30.291.2.1 dw_i2c_adapter_poll()

```
static int dw_i2c_adapter_poll (
            struct dw_i2c_adapter * adap,
            u32 mask,
            u32 addr,
            bool inverted )  [static]
```

Here is the call graph for this function:

30.291.2.2 dw_i2c_adapter_read()

```
static int dw_i2c_adapter_read (
            struct i2c_adapter * ia,
            u8 addr,
            u8 reg,
            u8 * buffer,
            int len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.291.2.3 dw_i2c_adapter_write()

```
static int dw_i2c_adapter_write (
            struct i2c_adapter * ia,
            u8 addr,
            u8 reg,
            u8 * buffer,
            int len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.291.2.4 dw_i2c_getreg()

```
static u32 dw_i2c_getreg (
            struct dw_i2c_adapter * adap,
            u32 reg )  [inline], [static]
```

Here is the caller graph for this function:

30.291.2.5 dw_i2c_init()

```
int dw_i2c_init (
            struct i2c_adapter * adapter,
            int nodeoff )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.291.2.6 dw_i2c_setreg()

```
static void dw_i2c_setreg (
            struct dw_i2c_adapter * adap,
```

```
            u8 reg,
            u32 value )  [inline], [static]
```

Here is the caller graph for this function:

30.291.2.7 dw_i2c_write_addr()

```
static int dw_i2c_write_addr (
            struct dw_i2c_adapter * adap,
            u8 addr )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.292 lib/utils/i2c/fdt_i2c.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/i2c/fdt_i2c.h>
```

Include dependency graph for fdt_i2c.c:

Functions

- static int fdt_i2c_adapter_find (const void ∗fdt, int nodeoff, struct i2c_adapter ∗∗out_adapter)
- int fdt_i2c_adapter_get (const void ∗fdt, int nodeoff, struct i2c_adapter ∗∗out_adapter)

Variables

- const struct fdt_driver ∗const fdt_i2c_adapter_drivers [ ]

30.292.1 Function Documentation

30.292.1.1 fdt_i2c_adapter_find()

```
static int fdt_i2c_adapter_find (
            const void * fdt,
            int nodeoff,
            struct i2c_adapter ** out_adapter )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.292.1.2 fdt_i2c_adapter_get()

```
int fdt_i2c_adapter_get (
            const void * fdt,
            int nodeoff,
            struct i2c_adapter ** out_adapter )
```

Get I2C adapter identified by nodeoff Here is the call graph for this function:

Here is the caller graph for this function:

30.292.2 Variable Documentation

30.292.2.1 fdt_i2c_adapter_drivers

```
const struct fdt_driver* const fdt_i2c_adapter_drivers[] [extern]
```

30.293 lib/utils/i2c/fdt_i2c_dw.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_string.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/i2c/dw_i2c.h>
#include <sbi_utils/i2c/fdt_i2c.h>
```

Include dependency graph for fdt_i2c_dw.c:

Functions

- static int fdt_dw_i2c_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match fdt_dw_i2c_match [ ]
- const struct fdt_driver fdt_i2c_adapter_dw

30.293.1 Function Documentation

30.293.1.1 fdt_dw_i2c_init()

```
static int fdt_dw_i2c_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.293.2 Variable Documentation

30.293.2.1 fdt_dw_i2c_match

```
const struct fdt_match fdt_dw_i2c_match[] [static]
```

Initial value:

```
= {
    { .compatible = "snps,designware-i2c" },
    { .compatible = "starfive,jh7110-i2c" },
    { },
}
```

30.293.2.2 fdt_i2c_adapter_dw

```
const struct fdt_driver fdt_i2c_adapter_dw
```

Initial value:

```
= {
    .match_table = fdt_dw_i2c_match,
    .init = fdt_dw_i2c_init,
}
```

30.294 lib/utils/i2c/fdt_i2c_sifive.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/i2c/fdt_i2c.h>
```

Include dependency graph for fdt_i2c_sifive.c:

Macros

- #define SIFIVE_I2C_PRELO 0x00
- #define SIFIVE_I2C_PREHI 0x04
- #define SIFIVE_I2C_CTR 0x08
- #define SIFIVE_I2C_TXR 0x00c
- #define SIFIVE_I2C_RXR SIFIVE_I2C_TXR
- #define SIFIVE_I2C_CR 0x010
- #define SIFIVE_I2C_SR SIFIVE_I2C_CR
- #define SIFIVE_I2C_CTR_IEN (1 << 6)
- #define SIFIVE_I2C_CTR_EN (1 << 7)
- #define SIFIVE_I2C_CMD_IACK (1 << 0)
- #define SIFIVE_I2C_CMD_ACK (1 << 3)
- #define SIFIVE_I2C_CMD_WR (1 << 4)

- #define SIFIVE_I2C_CMD_RD (1 << 5)
- #define SIFIVE_I2C_CMD_STO (1 << 6)
- #define SIFIVE_I2C_CMD_STA (1 << 7)
- #define SIFIVE_I2C_STATUS_IF (1 << 0)
- #define SIFIVE_I2C_STATUS_TIP (1 << 1)
- #define SIFIVE_I2C_STATUS_AL (1 << 5)
- #define SIFIVE_I2C_STATUS_BUSY (1 << 6)
- #define SIFIVE_I2C_STATUS_RXACK (1 << 7)
- #define SIFIVE_I2C_WRITE_BIT (0 << 0)
- #define SIFIVE_I2C_READ_BIT (1 << 0)
- #define sifive_i2c_adapter_poll_tip(adap) sifive_i2c_adapter_poll(adap, SIFIVE_I2C_STATUS_TIP)
- #define sifive_i2c_adapter_poll_busy(adap) sifive_i2c_adapter_poll(adap, SIFIVE_I2C_STATUS_BUSY)

Functions

- static void sifive_i2c_setreg (struct sifive_i2c_adapter *adap, uint8_t reg, uint8_t value)
- static uint8_t sifive_i2c_getreg (struct sifive_i2c_adapter *adap, uint8_t reg)
- static int sifive_i2c_adapter_rxack (struct sifive_i2c_adapter *adap)
- static int sifive_i2c_adapter_poll (struct sifive_i2c_adapter *adap, uint32_t mask)
- static int sifive_i2c_adapter_start (struct sifive_i2c_adapter *adap, uint8_t addr, uint8_t bit)
- static int sifive_i2c_adapter_write (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- static int sifive_i2c_adapter_read (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- static int sifive_i2c_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match sifive_i2c_match [ ]
- const struct fdt_driver fdt_i2c_adapter_sifive

30.294.1 Macro Definition Documentation

30.294.1.1 sifive_i2c_adapter_poll_busy

```
#define sifive_i2c_adapter_poll_busy(
            adap ) sifive_i2c_adapter_poll(adap, SIFIVE_I2C_STATUS_BUSY)
```

30.294.1.2 sifive_i2c_adapter_poll_tip

```
#define sifive_i2c_adapter_poll_tip(
            adap ) sifive_i2c_adapter_poll(adap, SIFIVE_I2C_STATUS_TIP)
```

30.294.1.3 SIFIVE_I2C_CMD_ACK

```
#define SIFIVE_I2C_CMD_ACK (1 << 3)
```

30.294.1.4 SIFIVE_I2C_CMD_IACK

```
#define SIFIVE_I2C_CMD_IACK (1 << 0)
```

30.294.1.5 SIFIVE_I2C_CMD_RD

```
#define SIFIVE_I2C_CMD_RD (1 << 5)
```

30.294.1.6 SIFIVE_I2C_CMD_STA

```
#define SIFIVE_I2C_CMD_STA (1 << 7)
```

30.294.1.7 SIFIVE_I2C_CMD_STO

```
#define SIFIVE_I2C_CMD_STO (1 << 6)
```

30.294.1.8 SIFIVE_I2C_CMD_WR

```
#define SIFIVE_I2C_CMD_WR (1 << 4)
```

30.294.1.9 SIFIVE_I2C_CR

```
#define SIFIVE_I2C_CR 0x010
```

30.294.1.10 SIFIVE_I2C_CTR

```
#define SIFIVE_I2C_CTR 0x08
```

30.294.1.11 SIFIVE_I2C_CTR_EN

```
#define SIFIVE_I2C_CTR_EN (1 << 7)
```

30.294.1.12 SIFIVE_I2C_CTR_IEN

```
#define SIFIVE_I2C_CTR_IEN (1 << 6)
```

30.294.1.13 SIFIVE_I2C_PREHI

```
#define SIFIVE_I2C_PREHI 0x04
```

30.294.1.14 SIFIVE_I2C_PRELO

```
#define SIFIVE_I2C_PRELO 0x00
```

30.294.1.15 SIFIVE_I2C_READ_BIT

```
#define SIFIVE_I2C_READ_BIT (1 << 0)
```

30.294.1.16 SIFIVE_I2C_RXR

```
#define SIFIVE_I2C_RXR SIFIVE_I2C_TXR
```

30.294.1.17 SIFIVE_I2C_SR

```
#define SIFIVE_I2C_SR SIFIVE_I2C_CR
```

30.294.1.18 SIFIVE_I2C_STATUS_AL

```
#define SIFIVE_I2C_STATUS_AL (1 << 5)
```

30.294.1.19 SIFIVE_I2C_STATUS_BUSY

```
#define SIFIVE_I2C_STATUS_BUSY (1 << 6)
```

30.294.1.20 SIFIVE_I2C_STATUS_IF

```
#define SIFIVE_I2C_STATUS_IF (1 << 0)
```

30.294.1.21 SIFIVE_I2C_STATUS_RXACK

```
#define SIFIVE_I2C_STATUS_RXACK (1 << 7)
```

30.294.1.22 SIFIVE_I2C_STATUS_TIP

```
#define SIFIVE_I2C_STATUS_TIP (1 << 1)
```

30.294.1.23 SIFIVE_I2C_TXR

```
#define SIFIVE_I2C_TXR 0x00c
```

30.294.1.24 SIFIVE_I2C_WRITE_BIT

```
#define SIFIVE_I2C_WRITE_BIT (0 << 0)
```

30.294.2 Function Documentation

30.294.2.1 sifive_i2c_adapter_poll()

```
static int sifive_i2c_adapter_poll (
            struct sifive_i2c_adapter * adap,
            uint32_t mask )  [static]
```

Here is the call graph for this function:

30.294.2.2 sifive_i2c_adapter_read()

```
static int sifive_i2c_adapter_read (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.294.2.3 sifive_i2c_adapter_rxack()

```
static int sifive_i2c_adapter_rxack (
            struct sifive_i2c_adapter * adap )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.294.2.4 sifive_i2c_adapter_start()

```
static int sifive_i2c_adapter_start (
            struct sifive_i2c_adapter * adap,
            uint8_t addr,
            uint8_t bit )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.294.2.5 sifive_i2c_adapter_write()

```
static int sifive_i2c_adapter_write (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.294.2.6 sifive_i2c_getreg()

```
static uint8_t sifive_i2c_getreg (
            struct sifive_i2c_adapter * adap,
            uint8_t reg )  [inline], [static]
```

Here is the caller graph for this function:

30.294.2.7 sifive_i2c_init()

```
static int sifive_i2c_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.294.2.8 sifive_i2c_setreg()

```
static void sifive_i2c_setreg (
            struct sifive_i2c_adapter * adap,
            uint8_t reg,
            uint8_t value )  [inline], [static]
```

Here is the caller graph for this function:

30.294.3 Variable Documentation

30.294.3.1 fdt_i2c_adapter_sifive

```
const struct fdt_driver fdt_i2c_adapter_sifive
```

Initial value:

```
= {
    .match_table = sifive_i2c_match,
    .init = sifive_i2c_init,
}
```

30.294.3.2 sifive_i2c_match

```
const struct fdt_match sifive_i2c_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "sifive,i2c0" },
    { },
}
```

30.295 lib/utils/i2c/i2c.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/i2c/i2c.h>
```

Include dependency graph for i2c.c:

Functions

- static SBI_LIST_HEAD (i2c_adapters_list)
- struct i2c_adapter ∗ i2c_adapter_find (int id)
- int i2c_adapter_add (struct i2c_adapter ∗ia)
- void i2c_adapter_remove (struct i2c_adapter ∗ia)
- int i2c_adapter_write (struct i2c_adapter ∗ia, uint8_t addr, uint8_t reg, uint8_t ∗buffer, int len)
- int i2c_adapter_read (struct i2c_adapter ∗ia, uint8_t addr, uint8_t reg, uint8_t ∗buffer, int len)

30.295.1 Function Documentation

30.295.1.1 i2c_adapter_add()

```
int i2c_adapter_add (
            struct i2c_adapter * ia )
```

Register I2C adapter Here is the call graph for this function:

Here is the caller graph for this function:

30.295.1.2 i2c_adapter_find()

```
struct i2c_adapter * i2c_adapter_find (
            int id )
```

Find a registered I2C adapter Here is the call graph for this function:

Here is the caller graph for this function:

30.295.1.3 i2c_adapter_read()

```
int i2c_adapter_read (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )
```

Read from device on I2C adapter bus Here is the caller graph for this function:

30.295.1.4 i2c_adapter_remove()

```
void i2c_adapter_remove (
            struct i2c_adapter * ia )
```

Un-register I2C adapter Here is the call graph for this function:

30.295.1.5 i2c_adapter_write()

```
int i2c_adapter_write (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )
```

Send to device on I2C adapter bus Here is the caller graph for this function:

30.295.1.6 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            i2c_adapters_list ) [static]
```

30.296 lib/utils/ipi/aclint_mswi.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/ipi/aclint_mswi.h>
```

Include dependency graph for aclint_mswi.c:

Macros

- #define mswi_get_hart_data_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, mswi_ptr_offset)
- #define mswi_set_hart_data_ptr(__scratch, __mswi) sbi_scratch_write_type((__scratch), void *, mswi_ptr_offset, (__mswi))

Functions

- static void mswi_ipi_send (u32 hart_index)
- static void mswi_ipi_clear (void)
- int aclint_mswi_cold_init (struct aclint_mswi_data *mswi)

Variables

- static unsigned long mswi_ptr_offset
- static struct sbi_ipi_device aclint_mswi

30.296.1 Macro Definition Documentation

30.296.1.1 mswi_get_hart_data_ptr

```
#define mswi_get_hart_data_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, mswi_ptr_offset)
```

30.296.1.2 mswi_set_hart_data_ptr

```
#define mswi_set_hart_data_ptr(
            __scratch,
            __mswi )  sbi_scratch_write_type((__scratch), void *, mswi_ptr_offset, (__mswi))
```

30.296.2 Function Documentation

30.296.2.1 aclint_mswi_cold_init()

```
int aclint_mswi_cold_init (
            struct aclint_mswi_data * mswi )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.296.2.2 mswi_ipi_clear()

```
static void mswi_ipi_clear (
            void  )  [static]
```

30.296.2.3 mswi_ipi_send()

```
static void mswi_ipi_send (
            u32 hart_index )  [static]
```

30.296.3 Variable Documentation

30.296.3.1 aclint_mswi

```
struct sbi_ipi_device aclint_mswi  [static]
```

Initial value:

```
= {
    .name = "aclint-mswi",
    .ipi_send = mswi_ipi_send,
    .ipi_clear = mswi_ipi_clear
}
```

30.296.3.2 mswi_ptr_offset

```
unsigned long mswi_ptr_offset  [static]
```

30.297 lib/utils/ipi/andes_plicsw.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ipi.h>
```

```
#include <sbi_utils/ipi/andes_plicsw.h>
```

Include dependency graph for andes_plicsw.c:

Functions

- static void plicsw_ipi_send (u32 hart_index)
- static void plicsw_ipi_clear (void)
- int plicsw_cold_ipi_init (struct plicsw_data *plicsw)

Variables

- struct plicsw_data plicsw
- static struct sbi_ipi_device plicsw_ipi

30.297.1 Function Documentation

30.297.1.1 plicsw_cold_ipi_init()

```
int plicsw_cold_ipi_init (
            struct plicsw_data * plicsw )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.297.1.2 plicsw_ipi_clear()

```
static void plicsw_ipi_clear (
            void ) [static]
```

30.297.1.3 plicsw_ipi_send()

```
static void plicsw_ipi_send (
            u32 hart_index ) [static]
```

30.297.2 Variable Documentation

30.297.2.1 plicsw

```
struct plicsw_data plicsw
```

30.297.2.2 plicsw_ipi

```
struct sbi_ipi_device plicsw_ipi [static]
```

Initial value:

```
= {
    .name      = "andes_plicsw",
    .ipi_send  = plicsw_ipi_send,
    .ipi_clear = plicsw_ipi_clear
}
```

30.298 lib/utils/ipi/fdt_ipi.c File Reference

```
#include <sbi_utils/ipi/fdt_ipi.h>
```

Include dependency graph for fdt_ipi.c:

Functions

- int fdt_ipi_init (void)

Variables

- const struct fdt_driver *const fdt_ipi_drivers [ ]

30.298.1 Function Documentation

30.298.1.1 fdt_ipi_init()

```
int fdt_ipi_init (
            void )
```

Here is the call graph for this function:

30.298.2 Variable Documentation

30.298.2.1 fdt_ipi_drivers

```
const struct fdt_driver* const fdt_ipi_drivers[] [extern]
```

30.299 lib/utils/ipi/fdt_ipi_mswi.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/ipi/fdt_ipi.h>
#include <sbi_utils/ipi/aclint_mswi.h>
```

Include dependency graph for fdt_ipi_mswi.c:

Functions

- static int ipi_mswi_cold_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const unsigned long clint_offset = CLINT_MSWI_OFFSET
- static const struct fdt_match ipi_mswi_match [ ]
- const struct fdt_driver fdt_ipi_mswi

30.299.1 Function Documentation

30.299.1.1 ipi_mswi_cold_init()

```
static int ipi_mswi_cold_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.299.2 Variable Documentation

30.299.2.1 clint_offset

```
const unsigned long clint_offset = CLINT_MSWI_OFFSET  [static]
```

30.299.2.2 fdt_ipi_mswi

```
const struct fdt_driver fdt_ipi_mswi
```

Initial value:

```
= {
    .match_table = ipi_mswi_match,
    .init = ipi_mswi_cold_init,
}
```

30.299.2.3 ipi_mswi_match

```
const struct fdt_match ipi_mswi_match[ ]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,clint0", .data = &clint_offset },
    { .compatible = "sifive,clint0", .data = &clint_offset },
    { .compatible = "thead,c900-clint", .data = &clint_offset },
    { .compatible = "thead,c900-aclint-mswi" },
    { .compatible = "riscv,aclint-mswi" },
    { },
}
```

30.300 lib/utils/ipi/fdt_ipi_plicsw.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/ipi/fdt_ipi.h>
#include <sbi_utils/ipi/andes_plicsw.h>
```

Include dependency graph for fdt_ipi_plicsw.c:

Functions

- int fdt_plicsw_cold_ipi_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- struct plicsw_data plicsw
- static const struct fdt_match ipi_plicsw_match [ ]
- const struct fdt_driver fdt_ipi_plicsw

30.300.1 Function Documentation

30.300.1.1 fdt_plicsw_cold_ipi_init()

```
int fdt_plicsw_cold_ipi_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )
```

Here is the call graph for this function:

30.300.2 Variable Documentation

30.300.2.1 fdt_ipi_plicsw

```
const struct fdt_driver fdt_ipi_plicsw
```

Initial value:

```
= {
    .match_table = ipi_plicsw_match,
    .init = fdt_plicsw_cold_ipi_init,
}
```

30.300.2.2 ipi_plicsw_match

```
const struct fdt_match ipi_plicsw_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "andestech,plicsw" },
    {},
}
```

30.300.2.3 plicsw

```
struct plicsw_data plicsw  [extern]
```

30.301 lib/utils/irqchip/aplic.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/irqchip/aplic.h>
```

Include dependency graph for aplic.c:

Macros

- #define APLIC_MAX_IDC (1UL << 14)
- #define APLIC_MAX_SOURCE 1024
- #define APLIC_DOMAINCFG 0x0000
- #define APLIC_DOMAINCFG_IE (1 << 8)
- #define APLIC_DOMAINCFG_DM (1 << 2)
- #define APLIC_DOMAINCFG_BE (1 << 0)
- #define APLIC_SOURCECFG_BASE 0x0004
- #define APLIC_SOURCECFG_D (1 << 10)
- #define APLIC_SOURCECFG_CHILDIDX_MASK 0x000003ff
- #define APLIC_SOURCECFG_SM_MASK 0x00000007
- #define APLIC_SOURCECFG_SM_INACTIVE 0x0
- #define APLIC_SOURCECFG_SM_DETACH 0x1
- #define APLIC_SOURCECFG_SM_EDGE_RISE 0x4
- #define APLIC_SOURCECFG_SM_EDGE_FALL 0x5
- #define APLIC_SOURCECFG_SM_LEVEL_HIGH 0x6
- #define APLIC_SOURCECFG_SM_LEVEL_LOW 0x7
- #define APLIC_MMSICFGADDR 0x1bc0
- #define APLIC_MMSICFGADDRH 0x1bc4
- #define APLIC_SMSICFGADDR 0x1bc8
- #define APLIC_SMSICFGADDRH 0x1bcc
- #define APLIC_xMSICFGADDRH_L (1UL << 31)
- #define APLIC_xMSICFGADDRH_HHXS_MASK 0x1f
- #define APLIC_xMSICFGADDRH_HHXS_SHIFT 24
- #define APLIC_xMSICFGADDRH_LHXS_MASK 0x7
- #define APLIC_xMSICFGADDRH_LHXS_SHIFT 20
- #define APLIC_xMSICFGADDRH_HHXW_MASK 0x7
- #define APLIC_xMSICFGADDRH_HHXW_SHIFT 16
- #define APLIC_xMSICFGADDRH_LHXW_MASK 0xf
- #define APLIC_xMSICFGADDRH_LHXW_SHIFT 12
- #define APLIC_xMSICFGADDRH_BAPPN_MASK 0xfff
- #define APLIC_xMSICFGADDR_PPN_SHIFT 12
- #define APLIC_xMSICFGADDR_PPN_HART(__lhxs) ((1UL << (__lhxs)) - 1)
- #define APLIC_xMSICFGADDR_PPN_LHX_MASK(__lhxw) ((1UL << (__lhxw)) - 1)
- #define APLIC_xMSICFGADDR_PPN_LHX_SHIFT(__lhxs) ((__lhxs))
- #define APLIC_xMSICFGADDR_PPN_LHX(__lhxw, __lhxs)
- #define APLIC_xMSICFGADDR_PPN_HHX_MASK(__hhxw) ((1UL << (__hhxw)) - 1)
- #define APLIC_xMSICFGADDR_PPN_HHX_SHIFT(__hhxs) ((__hhxs) + APLIC_xMSICFGADDR_PPN_SHIFT)
- #define APLIC_xMSICFGADDR_PPN_HHX(__hhxw, __hhxs)
- #define APLIC_SETIP_BASE 0x1c00
- #define APLIC_SETIPNUM 0x1cdc
- #define APLIC_CLRIP_BASE 0x1d00
- #define APLIC_CLRIPNUM 0x1ddc
- #define APLIC_SETIE_BASE 0x1e00
- #define APLIC_SETIENUM 0x1edc
- #define APLIC_CLRIE_BASE 0x1f00
- #define APLIC_CLRIENUM 0x1fdc
- #define APLIC_SETIPNUM_LE 0x2000
- #define APLIC_SETIPNUM_BE 0x2004
- #define APLIC_TARGET_BASE 0x3004
- #define APLIC_TARGET_HART_IDX_SHIFT 18
- #define APLIC_TARGET_HART_IDX_MASK 0x3fff
- #define APLIC_TARGET_GUEST_IDX_SHIFT 12
- #define APLIC_TARGET_GUEST_IDX_MASK 0x3f

- #define APLIC_TARGET_IPRIO_MASK 0xff
- #define APLIC_TARGET_EIID_MASK 0x7ff
- #define APLIC_IDC_BASE 0x4000
- #define APLIC_IDC_SIZE 32
- #define APLIC_IDC_IDELIVERY 0x00
- #define APLIC_IDC_IFORCE 0x04
- #define APLIC_IDC_ITHRESHOLD 0x08
- #define APLIC_IDC_TOPI 0x18
- #define APLIC_IDC_TOPI_ID_SHIFT 16
- #define APLIC_IDC_TOPI_ID_MASK 0x3ff
- #define APLIC_IDC_TOPI_PRIO_MASK 0xff
- #define APLIC_IDC_CLAIMI 0x1c
- #define APLIC_DEFAULT_PRIORITY 1
- #define APLIC_DISABLE_IDELIVERY 0
- #define APLIC_ENABLE_IDELIVERY 1
- #define APLIC_DISABLE_ITHRESHOLD 1
- #define APLIC_ENABLE_ITHRESHOLD 0

Functions

- static void aplic_writel_msicfg (struct aplic_msicfg_data *msicfg, void *msicfgaddr, void *msicfgaddrH)
- static int aplic_check_msicfg (struct aplic_msicfg_data *msicfg)
- int aplic_cold_irqchip_init (struct aplic_data *aplic)

30.301.1 Macro Definition Documentation

30.301.1.1 APLIC_CLRIE_BASE

```
#define APLIC_CLRIE_BASE 0x1f00
```

30.301.1.2 APLIC_CLRIENUM

```
#define APLIC_CLRIENUM 0x1fdc
```

30.301.1.3 APLIC_CLRIP_BASE

```
#define APLIC_CLRIP_BASE 0x1d00
```

30.301.1.4 APLIC_CLRIPNUM

```
#define APLIC_CLRIPNUM 0x1ddc
```

30.301.1.5 APLIC_DEFAULT_PRIORITY

```
#define APLIC_DEFAULT_PRIORITY 1
```

30.301.1.6 APLIC_DISABLE_IDELIVERY

```
#define APLIC_DISABLE_IDELIVERY 0
```

30.301.1.7 APLIC_DISABLE_ITHRESHOLD

```
#define APLIC_DISABLE_ITHRESHOLD 1
```

30.301.1.8 APLIC_DOMAINCFG

```
#define APLIC_DOMAINCFG 0x0000
```

30.301.1.9 APLIC_DOMAINCFG_BE

```
#define APLIC_DOMAINCFG_BE (1 << 0)
```

30.301.1.10 APLIC_DOMAINCFG_DM

```
#define APLIC_DOMAINCFG_DM (1 << 2)
```

30.301.1.11 APLIC_DOMAINCFG_IE

```
#define APLIC_DOMAINCFG_IE (1 << 8)
```

30.301.1.12 APLIC_ENABLE_IDELIVERY

```
#define APLIC_ENABLE_IDELIVERY 1
```

30.301.1.13 APLIC_ENABLE_ITHRESHOLD

```
#define APLIC_ENABLE_ITHRESHOLD 0
```

30.301.1.14 APLIC_IDC_BASE

```
#define APLIC_IDC_BASE 0x4000
```

30.301.1.15 APLIC_IDC_CLAIMI

```
#define APLIC_IDC_CLAIMI 0x1c
```

30.301.1.16 APLIC_IDC_IDELIVERY

```
#define APLIC_IDC_IDELIVERY 0x00
```

30.301.1.17 APLIC_IDC_IFORCE

```
#define APLIC_IDC_IFORCE 0x04
```

30.301.1.18 APLIC_IDC_ITHRESHOLD

```
#define APLIC_IDC_ITHRESHOLD 0x08
```

30.301.1.19 APLIC_IDC_SIZE

```
#define APLIC_IDC_SIZE 32
```

30.301.1.20 APLIC_IDC_TOPI

```
#define APLIC_IDC_TOPI 0x18
```

30.301.1.21 APLIC_IDC_TOPI_ID_MASK

```
#define APLIC_IDC_TOPI_ID_MASK 0x3ff
```

30.301.1.22 APLIC_IDC_TOPI_ID_SHIFT

```
#define APLIC_IDC_TOPI_ID_SHIFT 16
```

30.301.1.23 APLIC_IDC_TOPI_PRIO_MASK

```
#define APLIC_IDC_TOPI_PRIO_MASK 0xff
```

30.301.1.24 APLIC_MAX_IDC

```
#define APLIC_MAX_IDC (1UL << 14)
```

30.301.1.25 APLIC_MAX_SOURCE

```
#define APLIC_MAX_SOURCE 1024
```

30.301.1.26 APLIC_MMSICFGADDR

```
#define APLIC_MMSICFGADDR 0x1bc0
```

30.301.1.27 APLIC_MMSICFGADDRH

```
#define APLIC_MMSICFGADDRH 0x1bc4
```

30.301.1.28 APLIC_SETIE_BASE

```
#define APLIC_SETIE_BASE 0x1e00
```

30.301.1.29 APLIC_SETIENUM

```
#define APLIC_SETIENUM 0x1edc
```

30.301.1.30 APLIC_SETIP_BASE

```
#define APLIC_SETIP_BASE 0x1c00
```

30.301.1.31 APLIC_SETIPNUM

```
#define APLIC_SETIPNUM 0x1cdc
```

30.301.1.32 APLIC_SETIPNUM_BE

```
#define APLIC_SETIPNUM_BE 0x2004
```

30.301.1.33 APLIC_SETIPNUM_LE

```
#define APLIC_SETIPNUM_LE 0x2000
```

30.301.1.34 APLIC_SMSICFGADDR

```
#define APLIC_SMSICFGADDR 0x1bc8
```

30.301.1.35 APLIC_SMSICFGADDRH

```
#define APLIC_SMSICFGADDRH 0x1bcc
```

30.301.1.36 APLIC_SOURCECFG_BASE

```
#define APLIC_SOURCECFG_BASE 0x0004
```

30.301.1.37 APLIC_SOURCECFG_CHILDIDX_MASK

```
#define APLIC_SOURCECFG_CHILDIDX_MASK 0x000003ff
```

30.301.1.38 APLIC_SOURCECFG_D

```
#define APLIC_SOURCECFG_D (1 << 10)
```

30.301.1.39 APLIC_SOURCECFG_SM_DETACH

```
#define APLIC_SOURCECFG_SM_DETACH 0x1
```

30.301.1.40 APLIC_SOURCECFG_SM_EDGE_FALL

```
#define APLIC_SOURCECFG_SM_EDGE_FALL 0x5
```

30.301.1.41 APLIC_SOURCECFG_SM_EDGE_RISE

```
#define APLIC_SOURCECFG_SM_EDGE_RISE 0x4
```

30.301.1.42 APLIC_SOURCECFG_SM_INACTIVE

```
#define APLIC_SOURCECFG_SM_INACTIVE 0x0
```

30.301.1.43 APLIC_SOURCECFG_SM_LEVEL_HIGH

```
#define APLIC_SOURCECFG_SM_LEVEL_HIGH 0x6
```

30.301.1.44 APLIC_SOURCECFG_SM_LEVEL_LOW

```
#define APLIC_SOURCECFG_SM_LEVEL_LOW 0x7
```

30.301.1.45 APLIC_SOURCECFG_SM_MASK

```
#define APLIC_SOURCECFG_SM_MASK 0x00000007
```

30.301.1.46 APLIC_TARGET_BASE

```
#define APLIC_TARGET_BASE 0x3004
```

30.301.1.47 APLIC_TARGET_EIID_MASK

```
#define APLIC_TARGET_EIID_MASK 0x7ff
```

30.301.1.48 APLIC_TARGET_GUEST_IDX_MASK

```
#define APLIC_TARGET_GUEST_IDX_MASK 0x3f
```

30.301.1.49 APLIC_TARGET_GUEST_IDX_SHIFT

```
#define APLIC_TARGET_GUEST_IDX_SHIFT 12
```

30.301.1.50 APLIC_TARGET_HART_IDX_MASK

```
#define APLIC_TARGET_HART_IDX_MASK 0x3fff
```

30.301.1.51 APLIC_TARGET_HART_IDX_SHIFT

```
#define APLIC_TARGET_HART_IDX_SHIFT 18
```

30.301.1.52 APLIC_TARGET_IPRIO_MASK

```
#define APLIC_TARGET_IPRIO_MASK 0xff
```

30.301.1.53 APLIC_xMSICFGADDR_PPN_HART

```
#define APLIC_xMSICFGADDR_PPN_HART(
            __lhxs )  ((1UL << (__lhxs)) - 1)
```

30.301.1.54 APLIC_xMSICFGADDR_PPN_HHX

```
#define APLIC_xMSICFGADDR_PPN_HHX(
            __hhxw,
            __hhxs )
```

Value:

```
(APLIC_xMSICFGADDR_PPN_HHX_MASK(__hhxw) « \
 APLIC_xMSICFGADDR_PPN_HHX_SHIFT(__hhxs))
```

30.301.1.55 APLIC_xMSICFGADDR_PPN_HHX_MASK

```
#define APLIC_xMSICFGADDR_PPN_HHX_MASK(
            __hhxw )  ((1UL << (__hhxw)) - 1)
```

30.301.1.56 APLIC_xMSICFGADDR_PPN_HHX_SHIFT

```
#define APLIC_xMSICFGADDR_PPN_HHX_SHIFT(
            __hhxs )  ((__hhxs) + APLIC_xMSICFGADDR_PPN_SHIFT)
```

30.301.1.57 APLIC_xMSICFGADDR_PPN_LHX

```
#define APLIC_xMSICFGADDR_PPN_LHX(
            __lhxw,
            __lhxs )
```

Value:

```
(APLIC_xMSICFGADDR_PPN_LHX_MASK(__lhxw) « \
 APLIC_xMSICFGADDR_PPN_LHX_SHIFT(__lhxs))
```

30.301.1.58 APLIC_xMSICFGADDR_PPN_LHX_MASK

```
#define APLIC_xMSICFGADDR_PPN_LHX_MASK(
            __lhxw )  ((1UL << (__lhxw)) - 1)
```

30.301.1.59 APLIC_xMSICFGADDR_PPN_LHX_SHIFT

```
#define APLIC_xMSICFGADDR_PPN_LHX_SHIFT(
            __lhxs )  ((__lhxs))
```

30.301.1.60 APLIC_xMSICFGADDR_PPN_SHIFT

```
#define APLIC_xMSICFGADDR_PPN_SHIFT 12
```

30.301.1.61 APLIC_xMSICFGADDRH_BAPPN_MASK

```
#define APLIC_xMSICFGADDRH_BAPPN_MASK 0xfff
```

30.301.1.62 APLIC_xMSICFGADDRH_HHXS_MASK

```
#define APLIC_xMSICFGADDRH_HHXS_MASK 0x1f
```

30.301.1.63 APLIC_xMSICFGADDRH_HHXS_SHIFT

```
#define APLIC_xMSICFGADDRH_HHXS_SHIFT 24
```

30.301.1.64 APLIC_xMSICFGADDRH_HHXW_MASK

```
#define APLIC_xMSICFGADDRH_HHXW_MASK 0x7
```

30.301.1.65 APLIC_xMSICFGADDRH_HHXW_SHIFT

```
#define APLIC_xMSICFGADDRH_HHXW_SHIFT 16
```

30.301.1.66 APLIC_xMSICFGADDRH_L

```
#define APLIC_xMSICFGADDRH_L (1UL << 31)
```

30.301.1.67 APLIC_xMSICFGADDRH_LHXS_MASK

```
#define APLIC_xMSICFGADDRH_LHXS_MASK 0x7
```

30.301.1.68 APLIC_xMSICFGADDRH_LHXS_SHIFT

```
#define APLIC_xMSICFGADDRH_LHXS_SHIFT 20
```

30.301.1.69 APLIC_xMSICFGADDRH_LHXW_MASK

```
#define APLIC_xMSICFGADDRH_LHXW_MASK 0xf
```

30.301.1.70 APLIC_xMSICFGADDRH_LHXW_SHIFT

```
#define APLIC_xMSICFGADDRH_LHXW_SHIFT 12
```

30.301.2 Function Documentation

30.301.2.1 aplic_check_msicfg()

```
static int aplic_check_msicfg (
            struct aplic_msicfg_data * msicfg )  [static]
```

Here is the caller graph for this function:

30.301.2.2 aplic_cold_irqchip_init()

```
int aplic_cold_irqchip_init (
            struct aplic_data * aplic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.301.2.3 aplic_writel_msicfg()

```
static void aplic_writel_msicfg (
            struct aplic_msicfg_data * msicfg,
            void * msicfgaddr,
            void * msicfgaddrH )  [static]
```

Here is the caller graph for this function:

30.302 lib/utils/irqchip/fdt_irqchip.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/fdt_irqchip.h>
```

Include dependency graph for fdt_irqchip.c:

Functions

- int fdt_irqchip_init (void)

Variables

- struct fdt_irqchip ∗const fdt_irqchip_drivers [ ]

30.302.1 Function Documentation

30.302.1.1 fdt_irqchip_init()

```
int fdt_irqchip_init (
            void  )
```

Here is the call graph for this function:

30.302.2 Variable Documentation

30.302.2.1 fdt_irqchip_drivers

```
struct fdt_irqchip* const fdt_irqchip_drivers[] [extern]
```

30.303 lib/utils/irqchip/fdt_irqchip_aplic.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/fdt_irqchip.h>
#include <sbi_utils/irqchip/aplic.h>
```

Include dependency graph for fdt_irqchip_aplic.c:

Functions

- static int irqchip_aplic_cold_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match irqchip_aplic_match [ ]
- struct fdt_irqchip fdt_irqchip_aplic

30.303.1 Function Documentation

30.303.1.1 irqchip_aplic_cold_init()

```
static int irqchip_aplic_cold_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.303.2 Variable Documentation

30.303.2.1 fdt_irqchip_aplic

```
struct fdt_irqchip fdt_irqchip_aplic
```

Initial value:

```
= {
    .match_table = irqchip_aplic_match,
    .cold_init = irqchip_aplic_cold_init,
}
```

30.303.2.2 irqchip_aplic_match

```
const struct fdt_match irqchip_aplic_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,aplic" },
    { },
}
```

30.304 lib/utils/irqchip/fdt_irqchip_imsic.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/fdt_irqchip.h>
#include <sbi_utils/irqchip/imsic.h>
```

Include dependency graph for fdt_irqchip_imsic.c:

Functions

- static int irqchip_imsic_update_hartid_table (const void ∗fdt, int nodeoff, struct imsic_data ∗id)
- static int irqchip_imsic_cold_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match irqchip_imsic_match [ ]
- struct fdt_irqchip fdt_irqchip_imsic

30.304.1 Function Documentation

30.304.1.1 irqchip_imsic_cold_init()

```
static int irqchip_imsic_cold_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.304.1.2 irqchip_imsic_update_hartid_table()

```
static int irqchip_imsic_update_hartid_table (
            const void * fdt,
```

```
int nodeoff,
struct imsic_data * id )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.304.2 Variable Documentation

30.304.2.1 fdt_irqchip_imsic

```
struct fdt_irqchip fdt_irqchip_imsic
```

Initial value:

```
= {
    .match_table = irqchip_imsic_match,
    .cold_init = irqchip_imsic_cold_init,
}
```

30.304.2.2 irqchip_imsic_match

```
const struct fdt_match irqchip_imsic_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,imsics" },
    { },
}
```

30.305 lib/utils/irqchip/fdt_irqchip_plic.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/fdt_irqchip.h>
```

```
#include <sbi_utils/irqchip/plic.h>
```

Include dependency graph for fdt_irqchip_plic.c:

Functions

- static int irqchip_plic_update_context_map (const void *fdt, int nodeoff, struct plic_data *pd)
- static int irqchip_plic_cold_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match irqchip_plic_match [ ]
- struct fdt_irqchip fdt_irqchip_plic

30.305.1 Function Documentation

30.305.1.1 irqchip_plic_cold_init()

```
static int irqchip_plic_cold_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.305.1.2 irqchip_plic_update_context_map()

```
static int irqchip_plic_update_context_map (
            const void * fdt,
```

```
int nodeoff,
struct plic_data * pd )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.305.2 Variable Documentation

30.305.2.1 fdt_irqchip_plic

```
struct fdt_irqchip fdt_irqchip_plic
```

Initial value:

```
= {
    .match_table = irqchip_plic_match,
    .cold_init = irqchip_plic_cold_init,
}
```

30.305.2.2 irqchip_plic_match

```
const struct fdt_match irqchip_plic_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "andestech,nceplic100" },
    { .compatible = "riscv,plic0" },
    { .compatible = "sifive,plic-1.0.0" },
    { .compatible = "thead,c900-plic",
      .data = (void *)(PLIC_FLAG_THEAD_DELEGATION | PLIC_FLAG_ENABLE_PM) },
    {  }
}
```

30.306 lib/utils/irqchip/imsic.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_csr_detect.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_irqchip.h>
```

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/irqchip/imsic.h>
```

Include dependency graph for imsic.c:

Macros

- #define IMSIC_MMIO_PAGE_LE 0x00
- #define IMSIC_MMIO_PAGE_BE 0x04
- #define IMSIC_MIN_ID 63
- #define IMSIC_MAX_ID 2047
- #define IMSIC_EIDELIVERY 0x70
- #define IMSIC_EITHRESHOLD 0x72
- #define IMSIC_TOPEI 0x76
- #define IMSIC_TOPEI_ID_SHIFT 16
- #define IMSIC_TOPEI_ID_MASK 0x7ff
- #define IMSIC_TOPEI_PRIO_MASK 0x7ff
- #define IMSIC_EIP0 0x80
- #define IMSIC_EIP63 0xbf
- #define IMSIC_EIPx_BITS 32
- #define IMSIC_EIE0 0xc0
- #define IMSIC_EIE63 0xff
- #define IMSIC_EIEx_BITS 32
- #define IMSIC_DISABLE_EIDELIVERY 0
- #define IMSIC_ENABLE_EIDELIVERY 1
- #define IMSIC_DISABLE_EITHRESHOLD 1
- #define IMSIC_ENABLE_EITHRESHOLD 0
- #define IMSIC_IPI_ID 1
- #define imsic_csr_write(__c, __v)
- #define imsic_csr_read(__c)
- #define imsic_csr_set(__c, __v)
- #define imsic_csr_clear(__c, __v)
- #define imsic_get_hart_data_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, imsic_ptr_offset)
- #define imsic_set_hart_data_ptr(__scratch, __imsic) sbi_scratch_write_type((__scratch), void *, imsic_ptr_offset, (__imsic))
- #define imsic_get_hart_file(__scratch) sbi_scratch_read_type((__scratch), long, imsic_file_offset)
- #define imsic_set_hart_file(__scratch, __file) sbi_scratch_write_type((__scratch), long, imsic_file_offset, (↩ __file))

Functions

- int imsic_map_hartid_to_data (u32 hartid, struct imsic_data *imsic, int file)
- struct imsic_data * imsic_get_data (u32 hartindex)
- int imsic_get_target_file (u32 hartindex)
- static int imsic_external_irqfn (void)
- static void imsic_ipi_send (u32 hart_index)

- static void imsic_local_eix_update (unsigned long base_id, unsigned long num_id, bool pend, bool val)
- void imsic_local_irqchip_init (void)
- static int imsic_warm_irqchip_init (struct sbi_irqchip_device *dev)
- int imsic_data_check (struct imsic_data *imsic)
- int imsic_cold_irqchip_init (struct imsic_data *imsic)

Variables

- static unsigned long imsic_ptr_offset
- static unsigned long imsic_file_offset
- static struct sbi_ipi_device imsic_ipi_device
- static struct sbi_irqchip_device imsic_device

30.306.1 Macro Definition Documentation

30.306.1.1 imsic_csr_clear

```
#define imsic_csr_clear(
                __c,
                __v )
```

Value:

```
do { \
    csr_write(CSR_MISELECT, __c); \
    csr_clear(CSR_MIREG, __v); \
} while (0)
```

30.306.1.2 imsic_csr_read

```
#define imsic_csr_read(
                __c )
```

Value:

```
({ \
    unsigned long __v; \
    csr_write(CSR_MISELECT, __c); \
    __v = csr_read(CSR_MIREG); \
    __v; \
})
```

30.306.1.3 imsic_csr_set

```
#define imsic_csr_set(
                __c,
                __v )
```

Value:

```
do { \
    csr_write(CSR_MISELECT, __c); \
    csr_set(CSR_MIREG, __v); \
} while (0)
```

30.306.1.4 imsic_csr_write

```
#define imsic_csr_write(
                __c,
                __v )
```

Value:

```
do { \
    csr_write(CSR_MISELECT, __c); \
    csr_write(CSR_MIREG, __v); \
} while (0)
```

30.306.1.5 IMSIC_DISABLE_EIDELIVERY

```
#define IMSIC_DISABLE_EIDELIVERY 0
```

30.306.1.6 IMSIC_DISABLE_EITHRESHOLD

```
#define IMSIC_DISABLE_EITHRESHOLD 1
```

30.306.1.7 IMSIC_EIDELIVERY

```
#define IMSIC_EIDELIVERY 0x70
```

30.306.1.8 IMSIC_EIE0

```
#define IMSIC_EIE0 0xc0
```

30.306.1.9 IMSIC_EIE63

```
#define IMSIC_EIE63 0xff
```

30.306.1.10 IMSIC_EIEx_BITS

```
#define IMSIC_EIEx_BITS 32
```

30.306.1.11 IMSIC_EIP0

```
#define IMSIC_EIP0 0x80
```

30.306.1.12 IMSIC_EIP63

```
#define IMSIC_EIP63 0xbf
```

30.306.1.13 IMSIC_EIPx_BITS

```
#define IMSIC_EIPx_BITS 32
```

30.306.1.14 IMSIC_EITHRESHOLD

```
#define IMSIC_EITHRESHOLD 0x72
```

30.306.1.15 IMSIC_ENABLE_EIDELIVERY

```
#define IMSIC_ENABLE_EIDELIVERY 1
```

30.306.1.16 IMSIC_ENABLE_EITHRESHOLD

```
#define IMSIC_ENABLE_EITHRESHOLD 0
```

30.306.1.17 imsic_get_hart_data_ptr

```
#define imsic_get_hart_data_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, imsic_ptr_offset)
```

30.306.1.18 imsic_get_hart_file

```
#define imsic_get_hart_file(
            __scratch )  sbi_scratch_read_type((__scratch), long, imsic_file_offset)
```

30.306.1.19 IMSIC_IPI_ID

```
#define IMSIC_IPI_ID 1
```

30.306.1.20 IMSIC_MAX_ID

```
#define IMSIC_MAX_ID 2047
```

30.306.1.21 IMSIC_MIN_ID

```
#define IMSIC_MIN_ID 63
```

30.306.1.22 IMSIC_MMIO_PAGE_BE

```
#define IMSIC_MMIO_PAGE_BE 0x04
```

30.306.1.23 IMSIC_MMIO_PAGE_LE

```
#define IMSIC_MMIO_PAGE_LE 0x00
```

30.306.1.24 imsic_set_hart_data_ptr

```
#define imsic_set_hart_data_ptr(
            __scratch,
            __imsic )  sbi_scratch_write_type((__scratch), void *, imsic_ptr_offset, (__↩
imsic))
```

30.306.1.25 imsic_set_hart_file

```
#define imsic_set_hart_file(
            __scratch,
            __file )  sbi_scratch_write_type((__scratch), long, imsic_file_offset, (__file))
```

30.306.1.26 IMSIC_TOPEI

```
#define IMSIC_TOPEI 0x76
```

30.306.1.27 IMSIC_TOPEI_ID_MASK

```
#define IMSIC_TOPEI_ID_MASK 0x7ff
```

30.306.1.28 IMSIC_TOPEI_ID_SHIFT

```
#define IMSIC_TOPEI_ID_SHIFT 16
```

30.306.1.29 IMSIC_TOPEI_PRIO_MASK

```
#define IMSIC_TOPEI_PRIO_MASK 0x7ff
```

30.306.2 Function Documentation

30.306.2.1 imsic_cold_irqchip_init()

```
int imsic_cold_irqchip_init (
            struct imsic_data * imsic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.306.2.2 imsic_data_check()

```
int imsic_data_check (
            struct imsic_data * imsic )
```

Here is the caller graph for this function:

30.306.2.3 imsic_external_irqfn()

```
static int imsic_external_irqfn (
            void ) [static]
```

Here is the call graph for this function:

30.306.2.4 imsic_get_data()

```
struct imsic_data * imsic_get_data (
            u32 hartindex )
```

Here is the caller graph for this function:

30.306.2.5 imsic_get_target_file()

```
int imsic_get_target_file (
            u32 hartindex )
```

30.306.2.6 imsic_ipi_send()

```
static void imsic_ipi_send (
            u32 hart_index )  [static]
```

30.306.2.7 imsic_local_eix_update()

```
static void imsic_local_eix_update (
            unsigned long base_id,
            unsigned long num_id,
            bool pend,
            bool val )  [static]
```

Here is the caller graph for this function:

30.306.2.8 imsic_local_irqchip_init()

```
void imsic_local_irqchip_init (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.306.2.9 imsic_map_hartid_to_data()

```
int imsic_map_hartid_to_data (
            u32 hartid,
```

```
            struct imsic_data * imsic,
            int file )
```

Here is the caller graph for this function:

30.306.2.10 imsic_warm_irqchip_init()

```
static int imsic_warm_irqchip_init (
            struct sbi_irqchip_device * dev )  [static]
```

Here is the call graph for this function:

30.306.3 Variable Documentation

30.306.3.1 imsic_device

```
struct sbi_irqchip_device imsic_device  [static]
```

Initial value:

```
= {
    .warm_init  = imsic_warm_irqchip_init,
    .irq_handle = imsic_external_irqfn,
}
```

30.306.3.2 imsic_file_offset

```
unsigned long imsic_file_offset  [static]
```

30.306.3.3 imsic_ipi_device

```
struct sbi_ipi_device imsic_ipi_device  [static]
```

Initial value:

```
= {
    .name       = "aia-imsic",
    .ipi_send   = imsic_ipi_send
}
```

30.306.3.4 imsic_ptr_offset

```
unsigned long imsic_ptr_offset  [static]
```

30.307 lib/utils/irqchip/plic.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/riscv_encoding.h>
```

```
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_string.h>
#include <sbi_utils/irqchip/plic.h>
```

Include dependency graph for plic.c:

Macros

- #define PLIC_PRIORITY_BASE 0x0
- #define PLIC_PENDING_BASE 0x1000
- #define PLIC_ENABLE_BASE 0x2000
- #define PLIC_ENABLE_STRIDE 0x80
- #define PLIC_CONTEXT_BASE 0x200000
- #define PLIC_CONTEXT_STRIDE 0x1000
- #define THEAD_PLIC_CTRL_REG 0x1ffffc
- #define plic_get_hart_data_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, plic_ptr_offset)
- #define plic_set_hart_data_ptr(__scratch, __plic) sbi_scratch_write_type((__scratch), void *, plic_ptr_offset, (__plic))

Functions

- struct plic_data * plic_get (void)
- static u32 plic_get_priority (const struct plic_data *plic, u32 source)
- static void plic_set_priority (const struct plic_data *plic, u32 source, u32 val)
- static u32 plic_get_thresh (const struct plic_data *plic, u32 cntxid)
- static void plic_set_thresh (const struct plic_data *plic, u32 cntxid, u32 val)
- static u32 plic_get_ie (const struct plic_data *plic, u32 cntxid, u32 word_index)
- static void plic_set_ie (const struct plic_data *plic, u32 cntxid, u32 word_index, u32 val)
- static void plic_delegate (const struct plic_data *plic)
- static int plic_context_init (const struct plic_data *plic, int context_id, bool enable, u32 threshold)
- void plic_suspend (void)
- void plic_resume (void)
- static int plic_warm_irqchip_init (struct sbi_irqchip_device *dev)
- int plic_cold_irqchip_init (struct plic_data *plic)

Variables

- static unsigned long plic_ptr_offset

30.307.1 Macro Definition Documentation

30.307.1.1 PLIC_CONTEXT_BASE

```
#define PLIC_CONTEXT_BASE 0x200000
```

30.307.1.2 PLIC_CONTEXT_STRIDE

```
#define PLIC_CONTEXT_STRIDE 0x1000
```

30.307.1.3 PLIC_ENABLE_BASE

```
#define PLIC_ENABLE_BASE 0x2000
```

30.307.1.4 PLIC_ENABLE_STRIDE

```
#define PLIC_ENABLE_STRIDE 0x80
```

30.307.1.5 plic_get_hart_data_ptr

```
#define plic_get_hart_data_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, plic_ptr_offset)
```

30.307.1.6 PLIC_PENDING_BASE

```
#define PLIC_PENDING_BASE 0x1000
```

30.307.1.7 PLIC_PRIORITY_BASE

```
#define PLIC_PRIORITY_BASE 0x0
```

30.307.1.8 plic_set_hart_data_ptr

```
#define plic_set_hart_data_ptr(
            __scratch,
            __plic )  sbi_scratch_write_type((__scratch), void *, plic_ptr_offset, (__plic))
```

30.307.1.9 THEAD_PLIC_CTRL_REG

```
#define THEAD_PLIC_CTRL_REG 0x1ffffc
```

30.307.2 Function Documentation

30.307.2.1 plic_cold_irqchip_init()

```
int plic_cold_irqchip_init (
            struct plic_data * plic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.307.2.2 plic_context_init()

```
static int plic_context_init (
            const struct plic_data * plic,
            int context_id,
            bool enable,
            u32 threshold )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.307.2.3 plic_delegate()

```
static void plic_delegate (
            const struct plic_data * plic )  [static]
```

Here is the caller graph for this function:

30.307.2.4 plic_get()

```
struct plic_data * plic_get (
            void  )
```

30.307.2.5 plic_get_ie()

```
static u32 plic_get_ie (
            const struct plic_data * plic,
            u32 cntxid,
            u32 word_index )  [static]
```

Here is the caller graph for this function:

30.307.2.6 plic_get_priority()

```
static u32 plic_get_priority (
            const struct plic_data * plic,
            u32 source )  [static]
```

Here is the caller graph for this function:

30.307.2.7 plic_get_thresh()

```
static u32 plic_get_thresh (
            const struct plic_data * plic,
            u32 cntxid )  [static]
```

Here is the caller graph for this function:

30.307.2.8 plic_resume()

```
void plic_resume (
            void  )
```

Here is the call graph for this function:

30.307.2.9 plic_set_ie()

```
static void plic_set_ie (
            const struct plic_data * plic,
            u32 cntxid,
            u32 word_index,
            u32 val )  [static]
```

Here is the caller graph for this function:

30.307.2.10 plic_set_priority()

```
static void plic_set_priority (
            const struct plic_data * plic,
            u32 source,
            u32 val )  [static]
```

Here is the caller graph for this function:

30.307.2.11 plic_set_thresh()

```
static void plic_set_thresh (
            const struct plic_data * plic,
            u32 cntxid,
            u32 val )  [static]
```

Here is the caller graph for this function:

30.307.2.12 plic_suspend()

```
void plic_suspend (
            void  )
```

Here is the call graph for this function:

30.307.2.13 plic_warm_irqchip_init()

```
static int plic_warm_irqchip_init (
            struct sbi_irqchip_device * dev )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.307.3 Variable Documentation

30.307.3.1 plic_ptr_offset

```
unsigned long plic_ptr_offset  [static]
```

30.308 lib/utils/libfdt/fdt.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt.c:

Functions

- int32_t fdt_ro_probe_ (const void *fdt)
- static int check_off_ (uint32_t hdrsize, uint32_t totalsize, uint32_t off)
- static int check_block_ (uint32_t hdrsize, uint32_t totalsize, uint32_t base, uint32_t size)
- size_t fdt_header_size_ (uint32_t version)
- size_t fdt_header_size (const void *fdt)
- int fdt_check_header (const void *fdt)
- const void * fdt_offset_ptr (const void *fdt, int offset, unsigned int len)
- uint32_t fdt_next_tag (const void *fdt, int startoffset, int *nextoffset)
- int fdt_check_node_offset_ (const void *fdt, int offset)
- int fdt_check_prop_offset_ (const void *fdt, int offset)
- int fdt_next_node (const void *fdt, int offset, int *depth)
- int fdt_first_subnode (const void *fdt, int offset)
- int fdt_next_subnode (const void *fdt, int offset)
- const char * fdt_find_string_ (const char *strtab, int tabsize, const char *s)
- int fdt_move (const void *fdt, void *buf, int bufsize)

30.308.1 Function Documentation

30.308.1.1 check_block_()

```
static int check_block_ (
            uint32_t hdrsize,
            uint32_t totalsize,
            uint32_t base,
            uint32_t size )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.308.1.2 check_off_()

```
static int check_off_ (
            uint32_t hdrsize,
            uint32_t totalsize,
            uint32_t off )  [static]
```

Here is the caller graph for this function:

30.308.1.3 fdt_check_header()

```
int fdt_check_header (
            const void * fdt )
```

fdt_check_header - sanity check a device tree header @fdt: pointer to data which might be a flattened device tree fdt_check_header() checks that the given buffer contains what appears to be a flattened device tree, and that the header contains valid information (to the extent that can be determined from the header alone).
returns: 0, if the buffer appears to contain a valid device tree -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_TRUNCATED, standard meanings, as above Here is the call graph for this

function:

Here is the caller graph for this function:

30.308.1.4 fdt_check_node_offset_()

```
int fdt_check_node_offset_ (
            const void * fdt,
            int offset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.308.1.5 fdt_check_prop_offset_()

```
int fdt_check_prop_offset_ (
            const void * fdt,
            int offset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.308.1.6 fdt_find_string_()

```
const char * fdt_find_string_ (
            const char * strtab,
            int tabsize,
            const char * s )
```

Here is the caller graph for this function:

30.308.1.7 fdt_first_subnode()

```
int fdt_first_subnode (
            const void * fdt,
            int offset )
```

fdt_first_subnode() - get offset of first direct subnode @fdt: FDT blob @offset: Offset of node to check
Return: offset of first subnode, or -FDT_ERR_NOTFOUND if there is none Here is the call graph for this function:

30.308.1.8 fdt_header_size()

```
size_t fdt_header_size (
            const void * fdt )
```

fdt_header_size - return the size of the tree's header @fdt: pointer to a flattened device tree
Return: size of DTB header in bytes Here is the call graph for this function:

Here is the caller graph for this function:

30.308.1.9 fdt_header_size_()

```
size_t fdt_header_size_ (
            uint32_t version )
```

fdt_header_size_ - internal function to get header size from a version number

Version

: devicetree version number

Return: size of DTB header in bytes Here is the caller graph for this function:

30.308.1.10 fdt_move()

```
int fdt_move (
            const void * fdt,
            void * buf,
            int bufsize )
```

fdt_move - move a device tree around in memory @fdt: pointer to the device tree to move @buf: pointer to memory where the device is to be moved @bufsize: size of the memory space at buf

fdt_move() relocates, if possible, the device tree blob located at fdt to the buffer at buf of size bufsize. The buffer may overlap with the existing device tree blob at fdt. Therefore, fdt_move(fdt, fdt, fdt_totalsize(fdt)) should always succeed.

returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient to contain the device tree -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the caller graph for this function:

30.308.1.11 fdt_next_node()

```
int fdt_next_node (
            const void * fdt,
            int offset,
            int * depth )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.308.1.12 fdt_next_subnode()

```
int fdt_next_subnode (
            const void * fdt,
            int offset )
```

fdt_next_subnode() - get offset of next direct subnode @fdt: FDT blob @offset: Offset of previous subnode
After first calling fdt_first_subnode(), call this function repeatedly to get direct subnodes of a parent node.

Return: offset of next subnode, or -FDT_ERR_NOTFOUND if there are no more subnodes Here is the call graph for this function:

30.308.1.13 fdt_next_tag()

```
uint32_t fdt_next_tag (
            const void * fdt,
            int startoffset,
            int * nextoffset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.308.1.14 fdt_offset_ptr()

```
const void * fdt_offset_ptr (
            const void * fdt,
            int offset,
            unsigned int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.308.1.15 fdt_ro_probe_()

```
int32_t fdt_ro_probe_ (
            const void * fdt )
```

Here is the caller graph for this function:

30.309 lib/utils/libfdt/fdt.h File Reference

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_header
- struct fdt_reserve_entry
- struct fdt_node_header
- struct fdt_property

Macros

- #define FDT_MAGIC 0xd00dfeed /∗ 4: version, 4: total size ∗/
- #define FDT_TAGSIZE sizeof(fdt32_t)
- #define FDT_BEGIN_NODE 0x1 /∗ Start node: full name ∗/
- #define FDT_END_NODE 0x2 /∗ End node ∗/
- #define FDT_PROP
- #define FDT_NOP 0x4 /∗ nop ∗/
- #define FDT_END 0x9
- #define FDT_V1_SIZE (7∗sizeof(fdt32_t))
- #define FDT_V2_SIZE (FDT_V1_SIZE + sizeof(fdt32_t))
- #define FDT_V3_SIZE (FDT_V2_SIZE + sizeof(fdt32_t))
- #define FDT_V16_SIZE FDT_V3_SIZE
- #define FDT_V17_SIZE (FDT_V16_SIZE + sizeof(fdt32_t))

30.309.1 Macro Definition Documentation

30.309.1.1 FDT_BEGIN_NODE

```
#define FDT_BEGIN_NODE 0x1 /* Start node:  full name */
```

30.309.1.2 FDT_END

```
#define FDT_END 0x9
```

30.309.1.3 FDT_END_NODE

```
#define FDT_END_NODE 0x2 /* End node */
```

30.309.1.4 FDT_MAGIC

```
#define FDT_MAGIC 0xd00dfeed /* 4:  version, 4:  total size */
```

30.309.1.5 FDT_NOP

```
#define FDT_NOP 0x4 /* nop */
```

30.309.1.6 FDT_PROP

```
#define FDT_PROP
```

Value:

```
                    0x3      /* Property: name off,
                    size, content */
```

30.309.1.7 FDT_TAGSIZE

```
#define FDT_TAGSIZE sizeof(fdt32_t)
```

30.309.1.8 FDT_V16_SIZE

```
#define FDT_V16_SIZE FDT_V3_SIZE
```

30.309.1.9 FDT_V17_SIZE

```
#define FDT_V17_SIZE (FDT_V16_SIZE + sizeof(fdt32_t))
```

30.309.1.10 FDT_V1_SIZE

```
#define FDT_V1_SIZE (7*sizeof(fdt32_t))
```

30.309.1.11 FDT_V2_SIZE

```
#define FDT_V2_SIZE (FDT_V1_SIZE + sizeof(fdt32_t))
```

30.309.1.12 FDT_V3_SIZE

```
#define FDT_V3_SIZE (FDT_V2_SIZE + sizeof(fdt32_t))
```

30.310 fdt.h

Go to the documentation of this file.

```
/* SPDX-License-Identifier: GPL-2.0-or-later OR BSD-2-Clause */
#ifndef FDT_H
#define FDT_H
/*
 * libfdt - Flat Device Tree manipulation
 * Copyright (C) 2006 David Gibson, IBM Corporation.
 * Copyright 2012 Kim Phillips, Freescale Semiconductor.
 */

#ifndef __ASSEMBLY__

struct fdt_header {
    fdt32_t magic;           /* magic word FDT_MAGIC */
    fdt32_t totalsize;       /* total size of DT block */
    fdt32_t off_dt_struct;       /* offset to structure */
```

```
    fdt32_t off_dt_strings;         /* offset to strings */
    fdt32_t off_mem_rsvmap;         /* offset to memory reserve map */
    fdt32_t version;       /* format version */
    fdt32_t last_comp_version;   /* last compatible version */

    /* version 2 fields below */
    fdt32_t boot_cpuid_phys;   /* Which physical CPU id we're
                        booting on */
    /* version 3 fields below */
    fdt32_t size_dt_strings;   /* size of the strings block */

    /* version 17 fields below */
    fdt32_t size_dt_struct;         /* size of the structure block */
};

struct fdt_reserve_entry {
    fdt64_t address;
    fdt64_t size;
};

struct fdt_node_header {
    fdt32_t tag;
    char name[0];
};

struct fdt_property {
    fdt32_t tag;
    fdt32_t len;
    fdt32_t nameoff;
    char data[0];
};

#endif /* !__ASSEMBLY */

#define FDT_MAGIC   0xd00dfeed  /* 4: version, 4: total size */
#define FDT_TAGSIZE sizeof(fdt32_t)

#define FDT_BEGIN_NODE  0x1     /* Start node: full name */
#define FDT_END_NODE    0x2     /* End node */
#define FDT_PROP    0x3     /* Property: name off,
                        size, content */
#define FDT_NOP     0x4     /* nop */
#define FDT_END     0x9

#define FDT_V1_SIZE (7*sizeof(fdt32_t))
#define FDT_V2_SIZE (FDT_V1_SIZE + sizeof(fdt32_t))
#define FDT_V3_SIZE (FDT_V2_SIZE + sizeof(fdt32_t))
#define FDT_V16_SIZE    FDT_V3_SIZE
#define FDT_V17_SIZE    (FDT_V16_SIZE + sizeof(fdt32_t))

#endif /* FDT_H */
```

30.311 lib/utils/libfdt/fdt_addresses.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_addresses.c:

Functions

- static int fdt_cells (const void ∗fdt, int nodeoffset, const char ∗name)
- int fdt_address_cells (const void ∗fdt, int nodeoffset)
- int fdt_size_cells (const void ∗fdt, int nodeoffset)
- int fdt_appendprop_addrrange (void ∗fdt, int parent, int nodeoffset, const char ∗name, uint64_t addr, uint64↩ _t size)

30.311.1 Function Documentation

30.311.1.1 fdt_address_cells()

```
int fdt_address_cells (
            const void * fdt,
            int nodeoffset )
```

fdt_address_cells - retrieve address size for a bus represented in the tree @fdt: pointer to the device tree blob @nodeoffset: offset of the node to find the address size for

When the node has a valid #address-cells property, returns its value.

returns: 0 <= n < FDT_MAX_NCELLS, on success 2, if the node has no #address-cells property -FDT_ERR_↩ BADNCELLS, if the node has a badly formatted or invalid #address-cells property -FDT_ERR_BADMAGIC, -FDT↩ _ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.311.1.2 fdt_appendprop_addrrange()

```
int fdt_appendprop_addrrange (
            void * fdt,
            int parent,
            int nodeoffset,
            const char * name,
            uint64_t addr,
            uint64_t size )
```

Here is the call graph for this function:

30.311.1.3 fdt_cells()

```
static int fdt_cells (
            const void * fdt,
            int nodeoffset,
            const char * name )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.311.1.4 fdt_size_cells()

```
int fdt_size_cells (
            const void * fdt,
            int nodeoffset )
```

fdt_size_cells - retrieve address range size for a bus represented in the tree @fdt: pointer to the device tree blob @nodeoffset: offset of the node to find the address range size for

When the node has a valid #size-cells property, returns its value.

returns: 0 <= n < FDT_MAX_NCELLS, on success 1, if the node has no #size-cells property -FDT_ERR_↩ BADNCELLS, if the node has a badly formatted or invalid #size-cells property -FDT_ERR_BADMAGIC, -FDT↩ _ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, stan-

dard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.312 lib/utils/libfdt/fdt_check.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_check.c:

Functions

- int fdt_check_full (const void ∗fdt, size_t bufsize)

30.312.1 Function Documentation

30.312.1.1 fdt_check_full()

```
int fdt_check_full (
            const void * fdt,
            size_t bufsize )
```

Here is the call graph for this function:

30.313 lib/utils/libfdt/fdt_empty_tree.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_empty_tree.c:

Functions

- int fdt_create_empty_tree (void *buf, int bufsize)

30.313.1 Function Documentation

30.313.1.1 fdt_create_empty_tree()

```
int fdt_create_empty_tree (
            void * buf,
            int bufsize )
```

Here is the call graph for this function:

30.314 lib/utils/libfdt/fdt_overlay.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_overlay.c:

Functions

- static uint32_t overlay_get_target_phandle (const void *fdto, int fragment)

- static int overlay_get_target (const void *fdt, const void *fdto, int fragment, char const **pathp)

: Name of the property to modify (phandle or linux,phandle)

overlay_phandle_add_offset - Increases a phandle by an offset @fdt: Base device tree blob @node: Device tree overlay blob

@delta: offset to apply

overlay_phandle_add_offset() increments a node phandle by a given offset.

returns: 0 on success. Negative error code on error

- static int overlay_phandle_add_offset (void *fdt, int node, const char *name, uint32_t delta)
- static int overlay_adjust_node_phandles (void *fdto, int node, uint32_t delta)
- static int overlay_adjust_local_phandles (void *fdto, uint32_t delta)
- static int overlay_update_local_node_references (void *fdto, int tree_node, int fixup_node, uint32_t delta)
- static int overlay_update_local_references (void *fdto, uint32_t delta)

: Name of the property holding the phandle reference in the overlay

overlay_fixup_one_phandle - Set an overlay phandle to the base one @fdt: Base Device Tree blob @fdto↩ : Device tree overlay blob @symbols_off: Node offset of the symbols node in the base device tree @path: Path to a node holding a phandle in the overlay @path_len: number of path characters to consider

@name_len: number of name characters to consider @poffset: Offset within the overlay property where the phandle is stored @label: Label of the node referenced by the phandle

overlay_fixup_one_phandle() resolves an overlay phandle pointing to a node in the base device tree.

This is part of the device tree overlay application process, when you want all the phandles in the overlay to point to the actual base dt nodes.

returns: 0 on success Negative error code on failure

- static int overlay_fixup_one_phandle (void *fdt, void *fdto, int symbols_off, const char *path, uint32_t path_len, const char *name, uint32_t name_len, int poffset, const char *label)
- static int overlay_fixup_phandle (void *fdt, void *fdto, int symbols_off, int property)
- static int overlay_fixup_phandles (void *fdt, void *fdto)
- static int overlay_apply_node (void *fdt, int target, void *fdto, int node)
- static int overlay_merge (void *fdt, void *fdto)
- static int get_path_len (const void *fdt, int nodeoffset)
- static int overlay_symbol_update (void *fdt, void *fdto)
- int fdt_overlay_apply (void *fdt, void *fdto)

30.314.1 Function Documentation

30.314.1.1 fdt_overlay_apply()

```
int fdt_overlay_apply (
            void * fdt,
            void * fdto )
```

fdt_overlay_apply - Applies a DT overlay on a base DT @fdt: pointer to the base device tree blob @fdto: pointer to the device tree overlay blob

fdt_overlay_apply() will apply the given device tree overlay on the given base device tree.

Expect the base device tree to be modified, even if the function returns an error.

returns: 0, on success -FDT_ERR_NOSPACE, there's not enough space in the base device tree -FDT_ERR_↩ NOTFOUND, the overlay points to some inexistant nodes or properties in the base DT -FDT_ERR_BADPHANDLE, -FDT_ERR_BADOVERLAY, -FDT_ERR_NOPHANDLES, -FDT_ERR_INTERNAL, -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADOFFSET, -FDT_ERR_BADPATH, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTRUCTURE, -FDT_ERR_BADSTATE, -FDT_ERR_TRUNCATED, standard meanings Here is the call

graph for this function:

30.314.1.2 get_path_len()

```
static int get_path_len (
            const void * fdt,
            int nodeoffset )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.3 overlay_adjust_local_phandles()

```
static int overlay_adjust_local_phandles (
            void * fdto,
            uint32_t delta )  [static]
```

overlay_adjust_local_phandles - Adjust the phandles of a whole overlay @fdto: Device tree overlay blob @delta: Offset to shift the phandles of

overlay_adjust_local_phandles() adds a constant to all the phandles of an overlay. This is mainly use as part of the overlay application process, when we want to update all the overlay phandles to not conflict with the overlays of the base device tree.

returns: 0 on success Negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.4 overlay_adjust_node_phandles()

```
static int overlay_adjust_node_phandles (
            void * fdto,
            int node,
            uint32_t delta )  [static]
```

overlay_adjust_node_phandles - Offsets the phandles of a node @fdto: Device tree overlay blob @node: Offset of the node we want to adjust @delta: Offset to shift the phandles of

overlay_adjust_node_phandles() adds a constant to all the phandles of a given node. This is mainly use as part of the overlay application process, when we want to update all the overlay phandles to not conflict with the overlays of the base device tree.

returns: 0 on success Negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.5 overlay_apply_node()

```
static int overlay_apply_node (
            void * fdt,
```

```
                int target,
                void * fdto,
                int node )  [static]
```

overlay_apply_node - Merges a node into the base device tree @fdt: Base Device Tree blob @target: Node offset in the base device tree to apply the fragment to @fdto: Device tree overlay blob @node: Node offset in the overlay holding the changes to merge

overlay_apply_node() merges a node into a target base device tree node pointed.

This is part of the final step in the device tree overlay application process, when all the phandles have been adjusted and resolved and you just have to merge overlay into the base device tree.

returns: 0 on success Negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.6 overlay_fixup_one_phandle()

```
static int overlay_fixup_one_phandle (
                void * fdt,
                void * fdto,
                int symbols_off,
                const char * path,
                uint32_t path_len,
                const char * name,
                uint32_t name_len,
                int poffset,
                const char * label )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.7 overlay_fixup_phandle()

```
static int overlay_fixup_phandle (
            void * fdt,
            void * fdto,
            int symbols_off,
            int property )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.8 overlay_fixup_phandles()

```
static int overlay_fixup_phandles (
            void * fdt,
            void * fdto )  [static]
```

overlay_fixup_phandles - Resolve the overlay phandles to the base device tree @fdt: Base Device Tree blob @fdto: Device tree overlay blob

overlay_fixup_phandles() resolves all the overlay phandles pointing to nodes in the base device tree.

This is one of the steps of the device tree overlay application process, when you want all the phandles in the overlay to point to the actual base dt nodes.

returns: 0 on success Negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.9 overlay_get_target()

```
static int overlay_get_target (
            const void * fdt,
            const void * fdto,
            int fragment,
            char const ** pathp )  [static]
```

overlay_get_target - retrieves the offset of a fragment's target @fdt: Base device tree blob @fdto: Device tree overlay blob @fragment: node offset of the fragment in the overlay @pathp: pointer which receives the path of the target (or NULL)

overlay_get_target() retrieves the target offset in the base device tree of a fragment, no matter how the actual targeting is done (through a phandle or a path)

returns: the targeted node offset in the base device tree Negative error code on error Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.10 overlay_get_target_phandle()

```
static uint32_t overlay_get_target_phandle (
            const void * fdto,
            int fragment )  [static]
```

overlay_get_target_phandle - retrieves the target phandle of a fragment @fdto: pointer to the device tree overlay blob @fragment: node offset of the fragment in the overlay

overlay_get_target_phandle() retrieves the target phandle of an overlay fragment when that fragment uses a phandle (target property) instead of a path (target-path property).
returns: the phandle pointed by the target property 0, if the phandle was not found -1, if the phandle was malformed
Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.11 overlay_merge()

```
static int overlay_merge (
            void * fdt,
            void * fdto )  [static]
```

overlay_merge - Merge an overlay into its base device tree @fdt: Base Device Tree blob @fdto: Device tree overlay blob
overlay_merge() merges an overlay into its base device tree.
This is the next to last step in the device tree overlay application process, when all the phandles have been adjusted and resolved and you just have to merge overlay into the base device tree.
returns: 0 on success Negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.12 overlay_phandle_add_offset()

```
static int overlay_phandle_add_offset (
            void * fdt,
            int node,
            const char * name,
            uint32_t delta )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.13 overlay_symbol_update()

```
static int overlay_symbol_update (
            void * fdt,
            void * fdto )  [static]
```

overlay_symbol_update - Update the symbols of base tree after a merge @fdt: Base Device Tree blob @fdto: Device tree overlay blob

overlay_symbol_update() updates the symbols of the base tree with the symbols of the applied overlay

This is the last step in the device tree overlay application process, allowing the reference of overlay symbols by subsequent overlay operations.

returns: 0 on success Negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.14 overlay_update_local_node_references()

```
static int overlay_update_local_node_references (
            void * fdto,
            int tree_node,
            int fixup_node,
            uint32_t delta )  [static]
```

overlay_update_local_node_references - Adjust the overlay references @fdto: Device tree overlay blob @tree_↩node: Node offset of the node to operate on @fixup_node: Node offset of the matching local fixups node @delta: Offset to shift the phandles of

overlay_update_local_nodes_references() update the phandles pointing to a node within the device tree overlay by adding a constant delta.

This is mainly used as part of a device tree application process, where you want the device tree overlays phandles to not conflict with the ones from the base device tree before merging them.

returns: 0 on success Negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.314.1.15 overlay_update_local_references()

```
static int overlay_update_local_references (
            void * fdto,
            uint32_t delta )  [static]
```

overlay_update_local_references - Adjust the overlay references @fdto: Device tree overlay blob @delta: Offset to shift the phandles of

overlay_update_local_references() update all the phandles pointing to a node within the device tree overlay by adding a constant delta to not conflict with the base overlay.

This is mainly used as part of a device tree application process, where you want the device tree overlays phandles to not conflict with the ones from the base device tree before merging them.

returns: 0 on success Negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.315 lib/utils/libfdt/fdt_ro.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_ro.c:

Functions

- static int fdt_nodename_eq_ (const void *fdt, int offset, const char *s, int len)
- const char * fdt_get_string (const void *fdt, int stroffset, int *lenp)
- const char * fdt_string (const void *fdt, int stroffset)
- static int fdt_string_eq_ (const void *fdt, int stroffset, const char *s, int len)
- int fdt_find_max_phandle (const void *fdt, uint32_t *phandle)
- int fdt_generate_phandle (const void *fdt, uint32_t *phandle)
- static const struct fdt_reserve_entry * fdt_mem_rsv (const void *fdt, int n)
- int fdt_get_mem_rsv (const void *fdt, int n, uint64_t *address, uint64_t *size)
- int fdt_num_mem_rsv (const void *fdt)
- static int nextprop_ (const void *fdt, int offset)
- int fdt_subnode_offset_namelen (const void *fdt, int offset, const char *name, int namelen)
- int fdt_subnode_offset (const void *fdt, int parentoffset, const char *name)
- int fdt_path_offset_namelen (const void *fdt, const char *path, int namelen)
- int fdt_path_offset (const void *fdt, const char *path)
- const char * fdt_get_name (const void *fdt, int nodeoffset, int *len)
- int fdt_first_property_offset (const void *fdt, int nodeoffset)
- int fdt_next_property_offset (const void *fdt, int offset)
- static const struct fdt_property * fdt_get_property_by_offset_ (const void *fdt, int offset, int *lenp)
- const struct fdt_property * fdt_get_property_by_offset (const void *fdt, int offset, int *lenp)
- static const struct fdt_property * fdt_get_property_namelen_ (const void *fdt, int offset, const char *name, int namelen, int *lenp, int *poffset)
- const struct fdt_property * fdt_get_property_namelen (const void *fdt, int offset, const char *name, int namelen, int *lenp)
- const struct fdt_property * fdt_get_property (const void *fdt, int nodeoffset, const char *name, int *lenp)
- const void * fdt_getprop_namelen (const void *fdt, int nodeoffset, const char *name, int namelen, int *lenp)
- const void * fdt_getprop_by_offset (const void *fdt, int offset, const char **namep, int *lenp)
- const void * fdt_getprop (const void *fdt, int nodeoffset, const char *name, int *lenp)
- uint32_t fdt_get_phandle (const void *fdt, int nodeoffset)

- const char * fdt_get_alias_namelen (const void *fdt, const char *name, int namelen)
- const char * fdt_get_alias (const void *fdt, const char *name)
- int fdt_get_path (const void *fdt, int nodeoffset, char *buf, int buflen)
- int fdt_supernode_atdepth_offset (const void *fdt, int nodeoffset, int supernodedepth, int *nodedepth)
- int fdt_node_depth (const void *fdt, int nodeoffset)
- int fdt_parent_offset (const void *fdt, int nodeoffset)
- int fdt_node_offset_by_prop_value (const void *fdt, int startoffset, const char *propname, const void *propval, int proplen)
- int fdt_node_offset_by_phandle (const void *fdt, uint32_t phandle)
- int fdt_stringlist_contains (const char *strlist, int listlen, const char *str)
- int fdt_stringlist_count (const void *fdt, int nodeoffset, const char *property)
- int fdt_stringlist_search (const void *fdt, int nodeoffset, const char *property, const char *string)
- const char * fdt_stringlist_get (const void *fdt, int nodeoffset, const char *property, int idx, int *lenp)
- int fdt_node_check_compatible (const void *fdt, int nodeoffset, const char *compatible)
- int fdt_node_offset_by_compatible (const void *fdt, int startoffset, const char *compatible)

30.315.1 Function Documentation

30.315.1.1 fdt_find_max_phandle()

```
int fdt_find_max_phandle (
            const void * fdt,
            uint32_t * phandle )
```

fdt_find_max_phandle - find and return the highest phandle in a tree @fdt: pointer to the device tree blob @phandle: return location for the highest phandle value found in the tree

fdt_find_max_phandle() finds the highest phandle value in the given device tree. The value returned in @phandle is only valid if the function returns success.

returns: 0 on success or a negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.2 fdt_first_property_offset()

```
int fdt_first_property_offset (
            const void * fdt,
            int nodeoffset )
```

fdt_first_property_offset - find the offset of a node's first property @fdt: pointer to the device tree blob @nodeoffset: structure block offset of a node

fdt_first_property_offset() finds the first property of the node at the given structure block offset.

returns: structure block offset of the property (>=0), on success -FDT_ERR_NOTFOUND, if the requested node has no properties -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_BEGIN_NODE tag -FDT_ERR↩_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩_TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.3 fdt_generate_phandle()

```
int fdt_generate_phandle (
            const void * fdt,
            uint32_t * phandle )
```

fdt_generate_phandle - return a new, unused phandle for a device tree blob @fdt: pointer to the device tree blob @phandle: return location for the new phandle

Walks the device tree blob and looks for the highest phandle value. On success, the new, unused phandle value (one higher than the previously highest phandle value in the device tree blob) will be returned in the @phandle parameter.

Return: 0 on success or a negative error-code on failure Here is the call graph for this function:

30.315.1.4 fdt_get_alias()

```
const char * fdt_get_alias (
            const void * fdt,
            const char * name )
```

Here is the call graph for this function:

30.315.1.5 fdt_get_alias_namelen()

```
const char * fdt_get_alias_namelen (
            const void * fdt,
            const char * name,
            int namelen )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.6 fdt_get_mem_rsv()

```
int fdt_get_mem_rsv (
            const void * fdt,
            int n,
            uint64_t * address,
            uint64_t * size )
```

fdt_get_mem_rsv - retrieve one memory reserve map entry @fdt: pointer to the device tree blob
: index of reserve map entry @address: pointer to 64-bit variable to hold the start address @size: pointer to 64-bit variable to hold the size of the entry
On success, @address and @size will contain the address and size of the n-th reserve map entry from the device tree blob, in native-endian format.
returns: 0, on success -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:

30.315.1.7 fdt_get_name()

```
const char * fdt_get_name (
            const void * fdt,
            int nodeoffset,
            int * lenp )
```

fdt_get_name - retrieve the name of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of the starting node @lenp: pointer to an integer variable (will be overwritten) or NULL
fdt_get_name() retrieves the name (including unit address) of the device tree node at structure block offset nodeoffset. If lenp is non-NULL, the length of this name is also returned, in the integer pointed to by lenp.

returns: pointer to the node's name, on success If lenp is non-NULL, ∗lenp contains the length of that name (>=0) NULL, on error if lenp is non-NULL ∗lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.8 fdt_get_path()

```
int fdt_get_path (
            const void * fdt,
            int nodeoffset,
            char * buf,
            int buflen )
```

fdt_get_path - determine the full path of a node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose path to find @buf: character buffer to contain the returned path (will be overwritten) @buflen: size of the character buffer at buf

fdt_get_path() computes the full path of the node at offset nodeoffset, and records that path in the buffer at buf.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: 0, on success buf contains the absolute path of the node at nodeoffset, as a NUL-terminated string. -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOSPACE, the path of the

given node is longer than (bufsize-1) characters and will not fit in the given buffer. -FDT_ERR_BADMAGIC, -FDT_↩ ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.9 fdt_get_phandle()

```
uint32_t fdt_get_phandle (
            const void * fdt,
            int nodeoffset )
```

fdt_get_phandle - retrieve the phandle of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of the node

fdt_get_phandle() retrieves the phandle of the device tree node at structure block offset nodeoffset.

returns: the phandle of the node at nodeoffset, on success (!= 0, != -1) 0, if the node has no phandle, or another error occurs Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.10 fdt_get_property()

```
const struct fdt_property * fdt_get_property (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.11 fdt_get_property_by_offset()

```
const struct fdt_property * fdt_get_property_by_offset (
            const void * fdt,
            int offset,
            int * lenp )
```

fdt_get_property_by_offset - retrieve the property at a given offset @fdt: pointer to the device tree blob @offset: offset of the property to retrieve @lenp: pointer to an integer variable (will be overwritten) or NULL

fdt_get_property_by_offset() retrieves a pointer to the fdt_property structure within the device tree blob at the given offset. If lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by lenp.

Note that this code only works on device tree versions >= 16. fdt_getprop() works on all versions.

returns: pointer to the structure representing the property if lenp is non-NULL, ∗lenp contains the length of the property value (>=0) NULL, on error if lenp is non-NULL, ∗lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_PROP tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_↩ BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

30.315.1.12 fdt_get_property_by_offset_()

```
static const struct fdt_property * fdt_get_property_by_offset_ (
            const void * fdt,
            int offset,
            int * lenp )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.13 fdt_get_property_namelen()

```
const struct fdt_property * fdt_get_property_namelen (
            const void * fdt,
            int offset,
            const char * name,
            int namelen,
            int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.14 fdt_get_property_namelen_()

```
static const struct fdt_property * fdt_get_property_namelen_ (
            const void * fdt,
            int offset,
            const char * name,
            int namelen,
            int * lenp,
            int * poffset )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.15 fdt_get_string()

```
const char * fdt_get_string (
            const void * fdt,
            int stroffset,
            int * lenp )
```

fdt_get_string - retrieve a string from the strings block of a device tree @fdt: pointer to the device tree blob @stroffset: offset of the string within the strings block (native endian) @lenp: optional pointer to return the string's length

fdt_get_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt, and optionally also returns the string's length in ∗lenp.
returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.16 fdt_getprop()

```
const void * fdt_getprop (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:

30.315.1.17 fdt_getprop_by_offset()

```
const void * fdt_getprop_by_offset (
            const void * fdt,
            int offset,
            const char ** namep,
            int * lenp )
```

fdt_getprop_by_offset - retrieve the value of a property at a given offset @fdt: pointer to the device tree blob @offset: offset of the property to read @namep: pointer to a string variable (will be overwritten) or NULL @lenp: pointer to an integer variable (will be overwritten) or NULL

fdt_getprop_by_offset() retrieves a pointer to the value of the property at structure block offset 'offset' (this will be a pointer to within the device blob itself, not a copy of the value). If lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by lenp. If namep is non-NULL, the property's namne will also be returned in the char $*$ pointed to by namep (this will be a pointer to within the device tree's string block, not a new copy of the name).

returns: pointer to the property's value if lenp is non-NULL, $*$lenp contains the length of the property value ($>$=0) if namep is non-NULL $*$namep contiains a pointer to the property name. NULL, on error if lenp is non-NULL, $*$lenp contains an error code ($<$0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_PROP tag -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.18 fdt_getprop_namelen()

```
const void * fdt_getprop_namelen (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.19 fdt_mem_rsv()

```
static const struct fdt_reserve_entry * fdt_mem_rsv (
            const void * fdt,
            int n )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.20 fdt_next_property_offset()

```
int fdt_next_property_offset (
            const void * fdt,
            int offset )
```

fdt_next_property_offset - step through a node's properties @fdt: pointer to the device tree blob @offset: structure block offset of a property

fdt_next_property_offset() finds the property immediately after the one at the given structure block offset. This will be a property of the same node as the given property.

returns: structure block offset of the next property (>=0), on success -FDT_ERR_NOTFOUND, if the given property is the last in its node -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_PROP tag -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.21 fdt_node_check_compatible()

```
int fdt_node_check_compatible (
            const void * fdt,
            int nodeoffset,
            const char * compatible )
```

fdt_node_check_compatible - check a node's compatible property @fdt: pointer to the device tree blob @nodeoffset: offset of a tree node @compatible: string to match against

fdt_node_check_compatible() returns 0 if the given node contains a @compatible property with the given string as one of its elements, it returns non-zero otherwise, or on error.

returns: 0, if the node has a 'compatible' property listing the given string 1, if the node has a 'compatible' property, but it does not list the given string -FDT_ERR_NOTFOUND, if the given node has no 'compatible' property -FDT↩_ERR_BADOFFSET, if nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_↩BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph

for this function:

Here is the caller graph for this function:

30.315.1.22 fdt_node_depth()

```
int fdt_node_depth (
            const void * fdt,
            int nodeoffset )
```

fdt_node_depth - find the depth of a given node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find

fdt_node_depth() finds the depth of a given node. The root node has depth 0, its immediate subnodes depth 1 and so forth.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: depth of the node at nodeoffset (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.23 fdt_node_offset_by_compatible()

```
int fdt_node_offset_by_compatible (
            const void * fdt,
            int startoffset,
            const char * compatible )
```

fdt_node_offset_by_compatible - find nodes with a given 'compatible' value @fdt: pointer to the device tree blob @startoffset: only find nodes after this offset @compatible: 'compatible' string to match against

fdt_node_offset_by_compatible() returns the offset of the first node after startoffset, which has a 'compatible' property which lists the given compatible string; or if startoffset is -1, the very first such node in the tree.

To iterate through all nodes matching the criterion, the following idiom can be used: offset = fdt_node_offset_by_↩ compatible(fdt, -1, compatible); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_↩ offset_by_compatible(fdt, offset, compatible); }

Note the -1 in the first call to the function, if 0 is used here instead, the function will never locate the root node, even if it matches the criterion.

returns: structure block offset of the located node (>= 0, >startoffset), on success -FDT_ERR_NOTFOUND, no node matching the criterion exists in the tree after startoffset -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩ _BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.24 fdt_node_offset_by_phandle()

```
int fdt_node_offset_by_phandle (
            const void * fdt,
            uint32_t phandle )
```

fdt_node_offset_by_phandle - find the node with a given phandle @fdt: pointer to the device tree blob @phandle: phandle value

fdt_node_offset_by_phandle() returns the offset of the node which has the given phandle value. If there is more than one node in the tree with the given phandle (an invalid tree), results are undefined.

returns: structure block offset of the located node (>= 0), on success -FDT_ERR_NOTFOUND, no node with that phandle exists -FDT_ERR_BADPHANDLE, given phandle value was invalid (0 or -1) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.25 fdt_node_offset_by_prop_value()

```
int fdt_node_offset_by_prop_value (
            const void * fdt,
            int startoffset,
            const char * propname,
            const void * propval,
            int proplen )
```

fdt_node_offset_by_prop_value - find nodes with a given property value @fdt: pointer to the device tree blob @startoffset: only find nodes after this offset @propname: property name to check @propval: property value to search for @proplen: length of the value in propval

fdt_node_offset_by_prop_value() returns the offset of the first node after startoffset, which has a property named propname whose value is of length proplen and has value equal to propval; or if startoffset is -1, the very first such node in the tree.

To iterate through all nodes matching the criterion, the following idiom can be used: offset = fdt_node_offset_↩ by_prop_value(fdt, -1, propname, propval, proplen); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_offset_by_prop_value(fdt, offset, propname, propval, proplen); }

Note the -1 in the first call to the function, if 0 is used here instead, the function will never locate the root node, even if it matches the criterion.

returns: structure block offset of the located node (>= 0, >startoffset), on success -FDT_ERR_NOTFOUND, no node matching the criterion exists in the tree after startoffset -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩

_BADSTRUCTURE, standard meanings Here is the call graph for this function:

30.315.1.26 fdt_nodename_eq_()

```
static int fdt_nodename_eq_ (
            const void * fdt,
            int offset,
            const char * s,
            int len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.27 fdt_num_mem_rsv()

```
int fdt_num_mem_rsv (
            const void * fdt )
```

fdt_num_mem_rsv - retrieve the number of memory reserve map entries @fdt: pointer to the device tree blob
Returns the number of entries in the device tree blob's memory reservation map. This does not include the terminating 0,0 entry or any other (0,0) entries reserved for expansion.
returns: the number of entries Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.28 fdt_parent_offset()

```
int fdt_parent_offset (
            const void * fdt,
            int nodeoffset )
```

fdt_parent_offset - find the parent of a given node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find
fdt_parent_offset() locates the parent node of a given node (that is, it finds the offset of the node which contains the node at nodeoffset as a subnode).
NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset, *twice*.
returns: structure block offset of the parent of the node at nodeoffset (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.29 fdt_path_offset()

```
int fdt_path_offset (
            const void * fdt,
            const char * path )
```

fdt_path_offset - find a tree node by its full path @fdt: pointer to the device tree blob @path: full path of the node to locate

fdt_path_offset() finds a node of a given path in the device tree. Each path component may omit the unit address portion, but the results of this are undefined if any such path component is ambiguous (that is if there are multiple nodes at the relevant level matching the given component, differentiated only by unit address).

returns: structure block offset of the node with the requested path (>=0), on success -FDT_ERR_BADPATH, given path does not begin with '/' or is invalid -FDT_ERR_NOTFOUND, if the requested node does not exist -FDT_ERR↩_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩_TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.30 fdt_path_offset_namelen()

```
int fdt_path_offset_namelen (
            const void * fdt,
            const char * path,
            int namelen )
```

fdt_path_offset_namelen - find a tree node by its full path @fdt: pointer to the device tree blob @path: full path of the node to locate @namelen: number of characters of path to consider

Identical to fdt_path_offset(), but only consider the first namelen characters of path as the path name.

Return: offset of the node or negative libfdt error value otherwise Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.31 fdt_string()

```
const char * fdt_string (
            const void * fdt,
            int stroffset )
```

fdt_string - retrieve a string from the strings block of a device tree @fdt: pointer to the device tree blob @stroffset: offset of the string within the strings block (native endian)

fdt_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt.

returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.32 fdt_string_eq_()

```
static int fdt_string_eq_ (
            const void * fdt,
            int stroffset,
            const char * s,
            int len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.33 fdt_stringlist_contains()

```
int fdt_stringlist_contains (
            const char * strlist,
            int listlen,
            const char * str )
```

fdt_stringlist_contains - check a string list property for a string @strlist: Property containing a list of strings to check @listlen: Length of property @str: String to search for

This is a utility function provided for convenience. The list contains one or more strings, each terminated by \0, as is found in a device tree "compatible" property.

Return: 1 if the string is found in the list, 0 not found, or invalid list Here is the caller graph for this function:

30.315.1.34 fdt_stringlist_count()

```
int fdt_stringlist_count (
            const void * fdt,
            int nodeoffset,
            const char * property )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.35 fdt_stringlist_get()

```
const char * fdt_stringlist_get (
            const void * fdt,
            int nodeoffset,
            const char * property,
            int idx,
            int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.36 fdt_stringlist_search()

```
int fdt_stringlist_search (
            const void * fdt,
            int nodeoffset,
            const char * property,
            const char * string )
```

Here is the call graph for this function:

30.315.1.37 fdt_subnode_offset()

```
int fdt_subnode_offset (
            const void * fdt,
            int parentoffset,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.38 fdt_subnode_offset_namelen()

```
int fdt_subnode_offset_namelen (
            const void * fdt,
            int offset,
            const char * name,
            int namelen )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.39 fdt_supernode_atdepth_offset()

```
int fdt_supernode_atdepth_offset (
            const void * fdt,
            int nodeoffset,
            int supernodedepth,
            int * nodedepth )
```

fdt_supernode_atdepth_offset - find a specific ancestor of a node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find @supernodedepth: depth of the ancestor to find @nodedepth: pointer to an integer variable (will be overwritten) or NULL

fdt_supernode_atdepth_offset() finds an ancestor of the given node at a specific depth from the root (where the root itself has depth 0, its immediate subnodes depth 1 and so forth). So fdt_supernode_atdepth_offset(fdt, nodeoffset, 0, NULL); will always return 0, the offset of the root node. If the node at nodeoffset has depth D, then: fdt_supernode_atdepth_offset(fdt, nodeoffset, D, NULL); will return nodeoffset itself.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: structure block offset of the node at node offset's ancestor of depth supernodedepth (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOTFOUND, supernodedepth was greater than the depth of nodeoffset -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.315.1.40 nextprop_()

```
static int nextprop_ (
            const void * fdt,
            int offset )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316 lib/utils/libfdt/fdt_rw.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_rw.c:

Macros

- #define FDT_RW_PROBE(fdt)

Functions

- static int fdt_blocks_misordered_ (const void *fdt, int mem_rsv_size, int struct_size)
- static int fdt_rw_probe_ (void *fdt)
- static unsigned int fdt_data_size_ (void *fdt)
- static int fdt_splice_ (void *fdt, void *splicepoint, int oldlen, int newlen)
- static int fdt_splice_mem_rsv_ (void *fdt, struct fdt_reserve_entry *p, int oldn, int newn)
- static int fdt_splice_struct_ (void *fdt, void *p, int oldlen, int newlen)
- static void fdt_del_last_string_ (void *fdt, const char *s)
- static int fdt_splice_string_ (void *fdt, int newlen)
- static int fdt_find_add_string_ (void *fdt, const char *s, int *allocated)
- int fdt_add_mem_rsv (void *fdt, uint64_t address, uint64_t size)
- int fdt_del_mem_rsv (void *fdt, int n)
- static int fdt_resize_property_ (void *fdt, int nodeoffset, const char *name, int len, struct fdt_property **prop)
- static int fdt_add_property_ (void *fdt, int nodeoffset, const char *name, int len, struct fdt_property **prop)
- int fdt_set_name (void *fdt, int nodeoffset, const char *name)
- int fdt_setprop_placeholder (void *fdt, int nodeoffset, const char *name, int len, void **prop_data)
- int fdt_setprop (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- int fdt_appendprop (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- int fdt_delprop (void *fdt, int nodeoffset, const char *name)
- int fdt_add_subnode_namelen (void *fdt, int parentoffset, const char *name, int namelen)
- int fdt_add_subnode (void *fdt, int parentoffset, const char *name)
- int fdt_del_node (void *fdt, int nodeoffset)
- static void fdt_packblocks_ (const char *old, char *new, int mem_rsv_size, int struct_size, int strings_size)
- int fdt_open_into (const void *fdt, void *buf, int bufsize)
- int fdt_pack (void *fdt)

30.316.1 Macro Definition Documentation

30.316.1.1 FDT_RW_PROBE

```
#define FDT_RW_PROBE(
                fdt )
```

Value:

```
{ \
    int err_; \
    if ((err_ = fdt_rw_probe_(fdt)) != 0) \
        return err_; \
}
```

30.316.2 Function Documentation

30.316.2.1 fdt_add_mem_rsv()

```
int fdt_add_mem_rsv (
            void * fdt,
            uint64_t address,
            uint64_t size )
```

fdt_add_mem_rsv - add one memory reserve map entry @fdt: pointer to the device tree blob @address: 64-bit start address of the reserve map entry @size: 64-bit size of the reserved region

Adds a reserve map entry to the given blob reserving a region at address address of length size.

This function will insert data into the reserve map and will therefore change the indexes of some entries in the table. returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new reservation entry -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

30.316.2.2 fdt_add_property_()

```
static int fdt_add_property_ (
            void * fdt,
            int nodeoffset,
            const char * name,
            int len,
            struct fdt_property ** prop )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.3 fdt_add_subnode()

```
int fdt_add_subnode (
            void * fdt,
            int parentoffset,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.4 fdt_add_subnode_namelen()

```
int fdt_add_subnode_namelen (
            void * fdt,
            int parentoffset,
            const char * name,
            int namelen )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.5 fdt_appendprop()

```
int fdt_appendprop (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.6 fdt_blocks_misordered_()

```
static int fdt_blocks_misordered_ (
            const void * fdt,
            int mem_rsv_size,
            int struct_size )  [static]
```

Here is the caller graph for this function:

30.316.2.7 fdt_data_size_()

```
static unsigned int fdt_data_size_ (
            void * fdt )  [inline], [static]
```

Here is the caller graph for this function:

30.316.2.8 fdt_del_last_string_()

```
static void fdt_del_last_string_ (
            void * fdt,
            const char * s )  [static]
```

Here is the caller graph for this function:

30.316.2.9 fdt_del_mem_rsv()

```
int fdt_del_mem_rsv (
            void * fdt,
            int n )
```

fdt_del_mem_rsv - remove a memory reserve map entry @fdt: pointer to the device tree blob
: entry to remove
fdt_del_mem_rsv() removes the n-th memory reserve map entry from the blob.
This function will delete data from the reservation table and will therefore change the indexes of some entries in the table.
returns: 0, on success -FDT_ERR_NOTFOUND, there is no entry of the given index (i.e. there are less than n+1 reserve map entries) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph

for this function:

30.316.2.10 fdt_del_node()

```
int fdt_del_node (
            void * fdt,
            int nodeoffset )
```

fdt_del_node - delete a node (subtree) @fdt: pointer to the device tree blob @nodeoffset: offset of the node to nop fdt_del_node() will remove the given node, including all its subnodes if any, from the blob.
This function will delete data from the blob, and will therefore change the offsets of some existing nodes.
returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_↩ ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

30.316.2.11 fdt_delprop()

```
int fdt_delprop (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.12 fdt_find_add_string_()

```
static int fdt_find_add_string_ (
            void * fdt,
            const char * s,
            int * allocated )  [static]
```

fdt_find_add_string_() - Find or allocate a string
@fdt: pointer to the device tree to check/adjust @s: string to find/add @allocated: Set to 0 if the string was found, 1 if not found and so allocated. Ignored if can_assume(NO_ROLLBACK)

Returns

offset of string in the string table (whether found or added)

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.13 fdt_open_into()

```
int fdt_open_into (
            const void * fdt,
            void * buf,
            int bufsize )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.14 fdt_pack()

```
int fdt_pack (
            void * fdt )
```

Here is the call graph for this function:

30.316.2.15 fdt_packblocks_()

```
static void fdt_packblocks_ (
            const char * old,
            char * new,
            int mem_rsv_size,
```

```
            int struct_size,
            int strings_size )  [static]
```

Here is the caller graph for this function:

30.316.2.16 fdt_resize_property_()

```
static int fdt_resize_property_ (
            void * fdt,
            int nodeoffset,
            const char * name,
            int len,
            struct fdt_property ** prop )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.17 fdt_rw_probe_()

```
static int fdt_rw_probe_ (
            void * fdt )  [static]
```

Here is the call graph for this function:

30.316.2.18 fdt_set_name()

```
int fdt_set_name (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

30.316.2.19 fdt_setprop()

```
int fdt_setprop (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.20 fdt_setprop_placeholder()

```
int fdt_setprop_placeholder (
            void * fdt,
            int nodeoffset,
            const char * name,
            int len,
            void ** prop_data )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.21 fdt_splice_()

```
static int fdt_splice_ (
            void * fdt,
            void * splicepoint,
            int oldlen,
            int newlen )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.22 fdt_splice_mem_rsv_()

```
static int fdt_splice_mem_rsv_ (
            void * fdt,
            struct fdt_reserve_entry * p,
            int oldn,
            int newn )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.23 fdt_splice_string_()

```
static int fdt_splice_string_ (
            void * fdt,
            int newlen )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.316.2.24 fdt_splice_struct_()

```
static int fdt_splice_struct_ (
            void * fdt,
            void * p,
            int oldlen,
            int newlen )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.317 lib/utils/libfdt/fdt_strerror.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_strerror.c:

Macros

- #define FDT_ERRTABENT(val) [(val)] = { .str = #val, }
- #define FDT_ERRTABSIZE ((int)(sizeof(fdt_errtable) / sizeof(fdt_errtable[0])))

Functions

- const char * fdt_strerror (int errval)

Variables

- static struct fdt_errtabent fdt_errtable [ ]

30.317.1 Macro Definition Documentation

30.317.1.1 FDT_ERRTABENT

```
#define FDT_ERRTABENT(
            val )  [(val)] = { .str = #val, }
```

30.317.1.2 FDT_ERRTABSIZE

```
#define FDT_ERRTABSIZE ((int)(sizeof(fdt_errtable) / sizeof(fdt_errtable[0])))
```

30.317.2 Function Documentation

30.317.2.1 fdt_strerror()

```
const char * fdt_strerror (
            int errval )
```

30.317.3 Variable Documentation

30.317.3.1 fdt_errtable

```
struct fdt_errtabent fdt_errtable[]  [static]
```

Initial value:

```
= {
    FDT_ERRTABENT(FDT_ERR_NOTFOUND),
    FDT_ERRTABENT(FDT_ERR_EXISTS),
    FDT_ERRTABENT(FDT_ERR_NOSPACE),

    FDT_ERRTABENT(FDT_ERR_BADOFFSET),
    FDT_ERRTABENT(FDT_ERR_BADPATH),
    FDT_ERRTABENT(FDT_ERR_BADPHANDLE),
    FDT_ERRTABENT(FDT_ERR_BADSTATE),

    FDT_ERRTABENT(FDT_ERR_TRUNCATED),
    FDT_ERRTABENT(FDT_ERR_BADMAGIC),
    FDT_ERRTABENT(FDT_ERR_BADVERSION),
    FDT_ERRTABENT(FDT_ERR_BADSTRUCTURE),
    FDT_ERRTABENT(FDT_ERR_BADLAYOUT),
    FDT_ERRTABENT(FDT_ERR_INTERNAL),
    FDT_ERRTABENT(FDT_ERR_BADNCELLS),
    FDT_ERRTABENT(FDT_ERR_BADVALUE),
    FDT_ERRTABENT(FDT_ERR_BADOVERLAY),
    FDT_ERRTABENT(FDT_ERR_NOPHANDLES),
    FDT_ERRTABENT(FDT_ERR_BADFLAGS),
}
```

30.318 lib/utils/libfdt/fdt_sw.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_sw.c:

Macros

- #define FDT_SW_PROBE(fdt)
- #define FDT_SW_PROBE_MEMRSV(fdt)
- #define FDT_SW_PROBE_STRUCT(fdt)

Functions

- static int fdt_sw_probe_ (void ∗fdt)
- static int fdt_sw_probe_memrsv_ (void ∗fdt)
- static int fdt_sw_probe_struct_ (void ∗fdt)
- static uint32_t sw_flags (void ∗fdt)
- static void ∗ fdt_grab_space_ (void ∗fdt, size_t len)
- int fdt_create_with_flags (void ∗buf, int bufsize, uint32_t flags)
- int fdt_create (void ∗buf, int bufsize)
- int fdt_resize (void ∗fdt, void ∗buf, int bufsize)
- int fdt_add_reservemap_entry (void ∗fdt, uint64_t addr, uint64_t size)
- int fdt_finish_reservemap (void ∗fdt)
- int fdt_begin_node (void ∗fdt, const char ∗name)
- int fdt_end_node (void ∗fdt)
- static int fdt_add_string_ (void ∗fdt, const char ∗s)
- static void fdt_del_last_string_ (void ∗fdt, const char ∗s)
- static int fdt_find_add_string_ (void ∗fdt, const char ∗s, int ∗allocated)
- int fdt_property_placeholder (void ∗fdt, const char ∗name, int len, void ∗∗valp)
- int fdt_property (void ∗fdt, const char ∗name, const void ∗val, int len)
- int fdt_finish (void ∗fdt)

30.318.1 Macro Definition Documentation

30.318.1.1 FDT_SW_PROBE

```
#define FDT_SW_PROBE(
                fdt )
```

Value:

```
{ \
    int err; \
    if ((err = fdt_sw_probe_(fdt)) != 0) \
        return err; \
}
```

30.318.1.2 FDT_SW_PROBE_MEMRSV

```
#define FDT_SW_PROBE_MEMRSV(
                fdt )
```

Value:

```
{ \
    int err; \
    if ((err = fdt_sw_probe_memrsv_(fdt)) != 0) \
        return err; \
}
```

30.318.1.3 FDT_SW_PROBE_STRUCT

```
#define FDT_SW_PROBE_STRUCT(
                fdt )
```

Value:

```
{ \
    int err; \
    if ((err = fdt_sw_probe_struct_(fdt)) != 0) \
        return err; \
}
```

30.318.2 Function Documentation

30.318.2.1 fdt_add_reservemap_entry()

```
int fdt_add_reservemap_entry (
            void * fdt,
            uint64_t addr,
            uint64_t size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.318.2.2 fdt_add_string_()

```
static int fdt_add_string_ (
            void * fdt,
            const char * s )  [static]
```

Here is the caller graph for this function:

30.318.2.3 fdt_begin_node()

```
int fdt_begin_node (
            void * fdt,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.318.2.4 fdt_create()

```
int fdt_create (
            void * buf,
            int bufsize )
```

fdt_create - begin creation of a new fdt @buf: pointer to memory allocated where fdt will be created @bufsize: size of the memory space at fdt

fdt_create() is equivalent to fdt_create_with_flags() with flags=0.

returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient for a minimal fdt Here is the call graph for this

function:

Here is the caller graph for this function:

30.318.2.5 fdt_create_with_flags()

```
int fdt_create_with_flags (
            void * buf,
            int bufsize,
            uint32_t flags )
```

fdt_create_with_flags - begin creation of a new fdt @buf: pointer to memory allocated where fdt will be created @bufsize: size of the memory space at fdt @flags: a valid combination of FDT_CREATE_FLAG_ flags, or 0.
fdt_create_with_flags() begins the process of creating a new fdt with the sequential write interface.
fdt creation process must end with fdt_finished() to produce a valid fdt.
returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient for a minimal fdt -FDT_ERR_BADFLAGS, flags is not valid Here is the caller graph for this function:

30.318.2.6 fdt_del_last_string_()

```
static void fdt_del_last_string_ (
            void * fdt,
            const char * s )  [static]
```

Here is the caller graph for this function:

30.318.2.7 fdt_end_node()

```
int fdt_end_node (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.318.2.8 fdt_find_add_string_()

```
static int fdt_find_add_string_ (
            void * fdt,
            const char * s,
            int * allocated )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.318.2.9 fdt_finish()

```
int fdt_finish (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.318.2.10 fdt_finish_reservemap()

```
int fdt_finish_reservemap (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.318.2.11 fdt_grab_space_()

```
static void * fdt_grab_space_ (
            void * fdt,
            size_t len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.318.2.12 fdt_property()

```
int fdt_property (
            void * fdt,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

30.318.2.13 fdt_property_placeholder()

```
int fdt_property_placeholder (
            void * fdt,
            const char * name,
            int len,
            void ** valp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.318.2.14 fdt_resize()

```
int fdt_resize (
            void * fdt,
            void * buf,
            int bufsize )
```

30.318.2.15 fdt_sw_probe_()

```
static int fdt_sw_probe_ (
            void * fdt )  [static]
```

Here is the caller graph for this function:

30.318.2.16 fdt_sw_probe_memrsv_()

```
static int fdt_sw_probe_memrsv_ (
            void * fdt )  [static]
```

Here is the call graph for this function:

30.318.2.17 fdt_sw_probe_struct_()

```
static int fdt_sw_probe_struct_ (
            void * fdt )  [static]
```

Here is the call graph for this function:

30.318.2.18 sw_flags()

```
static uint32_t sw_flags (
            void * fdt )  [inline], [static]
```

Here is the caller graph for this function:

30.319 lib/utils/libfdt/fdt_wip.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_wip.c:

Functions

- int fdt_setprop_inplace_namelen_partial (void ∗fdt, int nodeoffset, const char ∗name, int namelen, uint32_t idx, const void ∗val, int len)
- int fdt_setprop_inplace (void ∗fdt, int nodeoffset, const char ∗name, const void ∗val, int len)
- static void fdt_nop_region_ (void ∗start, int len)
- int fdt_nop_property (void ∗fdt, int nodeoffset, const char ∗name)
- int fdt_node_end_offset_ (void ∗fdt, int offset)
- int fdt_nop_node (void ∗fdt, int nodeoffset)

30.319.1 Function Documentation

30.319.1.1 fdt_node_end_offset_()

```
int fdt_node_end_offset_ (
            void * fdt,
            int offset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.319.1.2 fdt_nop_node()

```
int fdt_nop_node (
            void * fdt,
            int nodeoffset )
```

fdt_nop_node - replace a node (subtree) with nop tags @fdt: pointer to the device tree blob @nodeoffset: offset of the node to nop

fdt_nop_node() will replace a given node's representation in the blob, including all its subnodes, if any, with FDT↩_NOP tags, effectively removing it from the tree.

This function will alter only the bytes in the blob which contain the node and its properties and subnodes, and will not alter or move any other part of the tree.

returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR↩_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩_TRUNCATED, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.319.1.3 fdt_nop_property()

```
int fdt_nop_property (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.319.1.4 fdt_nop_region_()

```
static void fdt_nop_region_ (
            void * start,
            int len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.319.1.5 fdt_setprop_inplace()

```
int fdt_setprop_inplace (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.319.1.6 fdt_setprop_inplace_namelen_partial()

```
int fdt_setprop_inplace_namelen_partial (
            void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            uint32_t idx,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320 lib/utils/libfdt/libfdt.h File Reference

```
#include <libfdt_env.h>
#include <fdt.h>
```

Include dependency graph for libfdt.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define FDT_FIRST_SUPPORTED_VERSION 0x02
- #define FDT_LAST_COMPATIBLE_VERSION 0x10
- #define FDT_LAST_SUPPORTED_VERSION 0x11
- #define FDT_ERR_NOTFOUND 1
- #define FDT_ERR_EXISTS 2
- #define FDT_ERR_NOSPACE 3
- #define FDT_ERR_BADOFFSET 4
- #define FDT_ERR_BADPATH 5
- #define FDT_ERR_BADPHANDLE 6
- #define FDT_ERR_BADSTATE 7

- #define FDT_ERR_TRUNCATED 8
- #define FDT_ERR_BADMAGIC 9
- #define FDT_ERR_BADVERSION 10
- #define FDT_ERR_BADSTRUCTURE 11
- #define FDT_ERR_BADLAYOUT 12
- #define FDT_ERR_INTERNAL 13
- #define FDT_ERR_BADNCELLS 14
- #define FDT_ERR_BADVALUE 15
- #define FDT_ERR_BADOVERLAY 16
- #define FDT_ERR_NOPHANDLES 17
- #define FDT_ERR_BADFLAGS 18
- #define FDT_ERR_ALIGNMENT 19
- #define FDT_ERR_MAX 19
- #define FDT_MAX_PHANDLE 0xfffffffe
- #define fdt_for_each_subnode(node, fdt, parent)
- #define fdt_get_header(fdt, field) (fdt32_ld(&((const struct fdt_header *)(fdt))->field))
- #define fdt_magic(fdt) (fdt_get_header(fdt, magic))
- #define fdt_totalsize(fdt) (fdt_get_header(fdt, totalsize))
- #define fdt_off_dt_struct(fdt) (fdt_get_header(fdt, off_dt_struct))
- #define fdt_off_dt_strings(fdt) (fdt_get_header(fdt, off_dt_strings))
- #define fdt_off_mem_rsvmap(fdt) (fdt_get_header(fdt, off_mem_rsvmap))
- #define fdt_version(fdt) (fdt_get_header(fdt, version))
- #define fdt_last_comp_version(fdt) (fdt_get_header(fdt, last_comp_version))
- #define fdt_boot_cpuid_phys(fdt) (fdt_get_header(fdt, boot_cpuid_phys))
- #define fdt_size_dt_strings(fdt) (fdt_get_header(fdt, size_dt_strings))
- #define fdt_size_dt_struct(fdt) (fdt_get_header(fdt, size_dt_struct))
- #define fdt_set_hdr_(name)

Functions

- const void * fdt_offset_ptr (const void *fdt, int offset, unsigned int checklen)
- static void * fdt_offset_ptr_w (void *fdt, int offset, int checklen)
- uint32_t fdt_next_tag (const void *fdt, int offset, int *nextoffset)
- static uint32_t fdt32_ld (const fdt32_t *p)
- static void fdt32_st (void *property, uint32_t value)
- static uint64_t fdt64_ld (const fdt64_t *p)
- static void fdt64_st (void *property, uint64_t value)
- int fdt_next_node (const void *fdt, int offset, int *depth)
- int fdt_first_subnode (const void *fdt, int offset)
- int fdt_next_subnode (const void *fdt, int offset)
- fdt_set_hdr_ (magic)
- fdt_set_hdr_ (totalsize)
- fdt_set_hdr_ (off_dt_struct)
- fdt_set_hdr_ (off_dt_strings)
- fdt_set_hdr_ (off_mem_rsvmap)
- fdt_set_hdr_ (version)
- fdt_set_hdr_ (last_comp_version)
- fdt_set_hdr_ (boot_cpuid_phys)
- fdt_set_hdr_ (size_dt_strings)
- fdt_set_hdr_ (size_dt_struct)
- size_t fdt_header_size (const void *fdt)
- size_t fdt_header_size_ (uint32_t version)
- int fdt_check_header (const void *fdt)
- int fdt_move (const void *fdt, void *buf, int bufsize)
- int fdt_check_full (const void *fdt, size_t bufsize)

- const char * fdt_get_string (const void *fdt, int stroffset, int *lenp)
- const char * fdt_string (const void *fdt, int stroffset)
- int fdt_find_max_phandle (const void *fdt, uint32_t *phandle)
- static uint32_t fdt_get_max_phandle (const void *fdt)
- int fdt_generate_phandle (const void *fdt, uint32_t *phandle)
- int fdt_num_mem_rsv (const void *fdt)
- int fdt_get_mem_rsv (const void *fdt, int n, uint64_t *address, uint64_t *size)

: name of the property to find

fdt_getprop - retrieve the value of a given property @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose property to find

@lenp: pointer to an integer variable (will be overwritten) or NULL

fdt_getprop() retrieves a pointer to the value of the property named

- const struct fdt_property * fdt_get_property_namelen (const void *fdt, int nodeoffset, const char *name, int namelen, int *lenp)
- const struct fdt_property * fdt_get_property (const void *fdt, int nodeoffset, const char *name, int *lenp)
- static struct fdt_property * fdt_get_property_w (void *fdt, int nodeoffset, const char *name, int *lenp)
- const void * fdt_getprop_by_offset (const void *fdt, int offset, const char **namep, int *lenp)
- const void * fdt_getprop_namelen (const void *fdt, int nodeoffset, const char *name, int namelen, int *lenp)
- static void * fdt_getprop_namelen_w (void *fdt, int nodeoffset, const char *name, int namelen, int *lenp)

of the node at offset @nodeoffset (this will be a

pointer to within the device blob itself, not a copy of the value). If @lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by @lenp.

*returns: pointer to the property's value if lenp is non-NULL, *lenp contains the length of the property value (>=0) NULL, on error if lenp is non-NULL, *lenp contains an error code (<0): -FDT_ERR_NOTFOUND, node does not have named property -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings*

- const void * fdt_getprop (const void *fdt, int nodeoffset, const char *name, int *lenp)
- static void * fdt_getprop_w (void *fdt, int nodeoffset, const char *name, int *lenp)
- uint32_t fdt_get_phandle (const void *fdt, int nodeoffset)

, if it exists,

NULL otherwise

- const char * fdt_get_alias_namelen (const void *fdt, const char *name, int namelen)

: name of the property to change the value of

fdt_setprop_inplace_cell - change the value of a single-cell property @fdt: pointer to the device tree blob @nodeoffset: offset of the node containing the property

@val: new value of the 32-bit cell

This is an alternative name for fdt_setprop_inplace_u32() Return: 0 on success, negative libfdt error number otherwise.

- static int fdt_setprop_inplace_cell (void *fdt, int nodeoffset, const char *name, uint32_t val)

: name to give the node

fdt_set_name - change the name of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of a node

fdt_set_name() replaces the name (including unit address, if any) of the given node with the given string. NOTE: this function can't efficiently check if the new name is unique amongst the given node's siblings; results are undefined if this function is invoked with a name equal to one of the given node's siblings.

This function may insert or delete data from the blob, and will therefore change the offsets of some existing nodes.

returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new name -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings

- int fdt_set_name (void *fdt, int nodeoffset, const char *name)

: name of the property to append to

fdt_appendprop - append to or create a property @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose property to change

@val: pointer to data to append to the property value @len: length of the data to append to the property value

fdt_appendprop() appends the value to the named property in the given node, creating the property if it does not already exist.

This function may insert data into the blob, and will therefore change the offsets of some existing nodes.

returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new property value -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings

- int fdt_appendprop (void *fdt, int nodeoffset, const char *name, const void *val, int len)

: name of property

fdt_appendprop_addrrange - append a address range property @fdt: pointer to the device tree blob @parent: offset of the parent node @nodeoffset: offset of the node to add a property at

@addr: start address of a given range @size: size of a given range

fdt_appendprop_addrrange() appends an address range value (start address and size) to the value of the named property in the given node, or creates a new property with that value if it does not already exist. If "name" is not specified, a default "reg" is used. Cell sizes are determined by parent's #address-cells and #size-cells.

This function may insert data into the blob, and will therefore change the offsets of some existing nodes.

returns: 0, on success -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADNCELLS, if the node has a badly formatted or invalid #address-cells property -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_BADVERSION, -FDT_ERR_BADVALUE, addr or size doesn't fit to respective cells size -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain a new property -FDT_ERR_TRUNCATED, standard meanings

- int fdt_appendprop_addrrange (void *fdt, int parent, int nodeoffset, const char *name, uint64_t addr, uint64_t size)

as the name of the new node. This is useful for

creating subnodes based on a portion of a larger string, such as a full path.

Return: structure block offset of the created subnode (>=0), negative libfdt error value otherwise

- int fdt_add_subnode_namelen (void *fdt, int parentoffset, const char *name, int namelen)

: name of the subnode to locate

fdt_add_subnode - creates a new node @fdt: pointer to the device tree blob @parentoffset: structure block offset of a node
fdt_add_subnode() creates a new node as a subnode of the node at structure block offset parentoffset, with the given name (which should include the unit address, if any).
This function will insert data into the blob, and will therefore change the offsets of some existing nodes.
returns: structure block offset of the created nodeequested subnode (>=0), on success -FDT_ERR_NOTFOUND, if the requested subnode does not exist -FDT_ERR_BADOFFSET, if parentoffset did not point to an FDT_BEGIN_NODE tag -FDT_ERR_EXISTS, if the node at parentoffset already has a subnode of the given name -FDT_ERR_NOSPACE, if there is insufficient free space in the blob to contain the new node -FDT_ERR_NOSPACE -FDT_ERR_BADLAYOUT -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings.

- #define fdt_for_each_property_offset(property, fdt, node)
- int fdt_subnode_offset_namelen (const void *fdt, int parentoffset, const char *name, int namelen)
- int fdt_subnode_offset (const void *fdt, int parentoffset, const char *name)
- int fdt_path_offset_namelen (const void *fdt, const char *path, int namelen)
- int fdt_path_offset (const void *fdt, const char *path)
- const char * fdt_get_name (const void *fdt, int nodeoffset, int *lenp)
- int fdt_first_property_offset (const void *fdt, int nodeoffset)
- int fdt_next_property_offset (const void *fdt, int offset)
- const struct fdt_property * fdt_get_property_by_offset (const void *fdt, int offset, int *lenp)
- int fdt_add_subnode (void *fdt, int parentoffset, const char *name)
- int fdt_del_node (void *fdt, int nodeoffset)
- int fdt_overlay_apply (void *fdt, void *fdto)
- const char * fdt_strerror (int errval)

in the node /aliases.

returns: a pointer to the expansion of the alias named 'name', if it exists NULL, if the given alias or the /aliases node does not exist

- #define FDT_MAX_NCELLS 4
- const char * fdt_get_alias (const void *fdt, const char *name)
- int fdt_get_path (const void *fdt, int nodeoffset, char *buf, int buflen)
- int fdt_supernode_atdepth_offset (const void *fdt, int nodeoffset, int supernodedepth, int *nodedepth)
- int fdt_node_depth (const void *fdt, int nodeoffset)
- int fdt_parent_offset (const void *fdt, int nodeoffset)
- int fdt_node_offset_by_prop_value (const void *fdt, int startoffset, const char *propname, const void *propval, int proplen)
- int fdt_node_offset_by_phandle (const void *fdt, uint32_t phandle)
- int fdt_node_check_compatible (const void *fdt, int nodeoffset, const char *compatible)
- int fdt_node_offset_by_compatible (const void *fdt, int startoffset, const char *compatible)
- int fdt_stringlist_contains (const char *strlist, int listlen, const char *str)
- int fdt_stringlist_count (const void *fdt, int nodeoffset, const char *property)
- int fdt_stringlist_search (const void *fdt, int nodeoffset, const char *property, const char *string)
- const char * fdt_stringlist_get (const void *fdt, int nodeoffset, const char *property, int index, int *lenp)
- int fdt_address_cells (const void *fdt, int nodeoffset)
- int fdt_size_cells (const void *fdt, int nodeoffset)

: name of the property to nop

fdt_delprop - delete a property @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose property to nop
fdt_del_property() will delete the given property.
This function will delete data from the blob, and will therefore change the offsets of some existing nodes.
returns: 0, on success -FDT_ERR_NOTFOUND, node does not have the named property -FDT_ERR_↩ BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADLAYOUT, -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings

- #define FDT_CREATE_FLAG_NO_NAME_DEDUP 0x1
- #define FDT_CREATE_FLAGS_ALL (FDT_CREATE_FLAG_NO_NAME_DEDUP)
- int fdt_nop_property (void *fdt, int nodeoffset, const char *name)
- int fdt_nop_node (void *fdt, int nodeoffset)
- int fdt_create_with_flags (void *buf, int bufsize, uint32_t flags)
- int fdt_create (void *buf, int bufsize)
- int fdt_resize (void *fdt, void *buf, int bufsize)
- int fdt_add_reservemap_entry (void *fdt, uint64_t addr, uint64_t size)
- int fdt_finish_reservemap (void *fdt)

- int fdt_begin_node (void *fdt, const char *name)
- int fdt_property (void *fdt, const char *name, const void *val, int len)
- static int fdt_property_u32 (void *fdt, const char *name, uint32_t val)
- static int fdt_property_u64 (void *fdt, const char *name, uint64_t val)
- static int fdt_property_cell (void *fdt, const char *name, uint32_t val)
- int fdt_delprop (void *fdt, int nodeoffset, const char *name)

: name of property to add

fdt_property_placeholder - add a new property and return a ptr to its value
@fdt: pointer to the device tree blob
@len: length of property value in bytes @valp: returns a pointer to where where the value should be placed
returns: 0, on success -FDT_ERR_BADMAGIC, -FDT_ERR_NOSPACE, standard meanings

- #define fdt_property_string(fdt, name, str) fdt_property(fdt, name, str, strlen(str)+1)
- int fdt_property_placeholder (void *fdt, const char *name, int len, void **valp)
- int fdt_end_node (void *fdt)
- int fdt_finish (void *fdt)
- int fdt_create_empty_tree (void *buf, int bufsize)
- int fdt_open_into (const void *fdt, void *buf, int bufsize)
- int fdt_pack (void *fdt)
- int fdt_add_mem_rsv (void *fdt, uint64_t address, uint64_t size)
- int fdt_del_mem_rsv (void *fdt, int n)

: name of the property to change

fdt_appendprop_string - append a string to a property @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose property to change
@str: string value to append to the property
fdt_appendprop_string() appends the given string to the value of the named property in the given node, or creates a new property with that value if it does not already exist.
This function may insert data into the blob, and will therefore change the offsets of some existing nodes.
returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new property value -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings

- #define fdt_setprop_string(fdt, nodeoffset, name, str) fdt_setprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
- #define fdt_setprop_empty(fdt, nodeoffset, name) fdt_setprop((fdt), (nodeoffset), (name), NULL, 0)
- #define fdt_appendprop_string(fdt, nodeoffset, name, str) fdt_appendprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
- int fdt_setprop_inplace_namelen_partial (void *fdt, int nodeoffset, const char *name, int namelen, uint32_t idx, const void *val, int len)
- int fdt_setprop_inplace (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- static int fdt_setprop_inplace_u32 (void *fdt, int nodeoffset, const char *name, uint32_t val)
- static int fdt_setprop_inplace_u64 (void *fdt, int nodeoffset, const char *name, uint64_t val)
- int fdt_setprop (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- int fdt_setprop_placeholder (void *fdt, int nodeoffset, const char *name, int len, void **prop_data)
- static int fdt_setprop_u32 (void *fdt, int nodeoffset, const char *name, uint32_t val)
- static int fdt_setprop_u64 (void *fdt, int nodeoffset, const char *name, uint64_t val)
- static int fdt_setprop_cell (void *fdt, int nodeoffset, const char *name, uint32_t val)
- static int fdt_appendprop_u32 (void *fdt, int nodeoffset, const char *name, uint32_t val)
- static int fdt_appendprop_u64 (void *fdt, int nodeoffset, const char *name, uint64_t val)
- static int fdt_appendprop_cell (void *fdt, int nodeoffset, const char *name, uint32_t val)

30.320.1 Macro Definition Documentation

30.320.1.1 fdt_appendprop_string

```
#define fdt_appendprop_string(
            fdt,
            nodeoffset,
            name,
            str )  fdt_appendprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
```

30.320.1.2 fdt_boot_cpuid_phys

```
#define fdt_boot_cpuid_phys(
            fdt ) (fdt_get_header(fdt, boot_cpuid_phys))
```

30.320.1.3 FDT_CREATE_FLAG_NO_NAME_DEDUP

```
#define FDT_CREATE_FLAG_NO_NAME_DEDUP 0x1
```

30.320.1.4 FDT_CREATE_FLAGS_ALL

```
#define FDT_CREATE_FLAGS_ALL (FDT_CREATE_FLAG_NO_NAME_DEDUP)
```

30.320.1.5 FDT_ERR_ALIGNMENT

```
#define FDT_ERR_ALIGNMENT 19
```

30.320.1.6 FDT_ERR_BADFLAGS

```
#define FDT_ERR_BADFLAGS 18
```

30.320.1.7 FDT_ERR_BADLAYOUT

```
#define FDT_ERR_BADLAYOUT 12
```

30.320.1.8 FDT_ERR_BADMAGIC

```
#define FDT_ERR_BADMAGIC 9
```

30.320.1.9 FDT_ERR_BADNCELLS

```
#define FDT_ERR_BADNCELLS 14
```

30.320.1.10 FDT_ERR_BADOFFSET

```
#define FDT_ERR_BADOFFSET 4
```

30.320.1.11 FDT_ERR_BADOVERLAY

```
#define FDT_ERR_BADOVERLAY 16
```

30.320.1.12 FDT_ERR_BADPATH

```
#define FDT_ERR_BADPATH 5
```

30.320.1.13 FDT_ERR_BADPHANDLE

```
#define FDT_ERR_BADPHANDLE 6
```

30.320.1.14 FDT_ERR_BADSTATE

```
#define FDT_ERR_BADSTATE 7
```

30.320.1.15 FDT_ERR_BADSTRUCTURE

```
#define FDT_ERR_BADSTRUCTURE 11
```

30.320.1.16 FDT_ERR_BADVALUE

```
#define FDT_ERR_BADVALUE 15
```

30.320.1.17 FDT_ERR_BADVERSION

```
#define FDT_ERR_BADVERSION 10
```

30.320.1.18 FDT_ERR_EXISTS

```
#define FDT_ERR_EXISTS 2
```

30.320.1.19 FDT_ERR_INTERNAL

```
#define FDT_ERR_INTERNAL 13
```

30.320.1.20 FDT_ERR_MAX

```
#define FDT_ERR_MAX 19
```

30.320.1.21 FDT_ERR_NOPHANDLES

```
#define FDT_ERR_NOPHANDLES 17
```

30.320.1.22 FDT_ERR_NOSPACE

```
#define FDT_ERR_NOSPACE 3
```

30.320.1.23 FDT_ERR_NOTFOUND

```
#define FDT_ERR_NOTFOUND 1
```

30.320.1.24 FDT_ERR_TRUNCATED

```
#define FDT_ERR_TRUNCATED 8
```

30.320.1.25 FDT_FIRST_SUPPORTED_VERSION

```
#define FDT_FIRST_SUPPORTED_VERSION 0x02
```

30.320.1.26 fdt_for_each_property_offset

```
#define fdt_for_each_property_offset(
            property,
            fdt,
            node )
```

Value:

```
for (property = fdt_first_property_offset(fdt, node);  \
     property >= 0;                                    \
     property = fdt_next_property_offset(fdt, property))
```

30.320.1.27 fdt_for_each_subnode

```
#define fdt_for_each_subnode(
            node,
            fdt,
            parent )
```

Value:

```
for (node = fdt_first_subnode(fdt, parent);    \
     node >= 0;                                \
     node = fdt_next_subnode(fdt, node))
```

fdt_for_each_subnode - iterate over all subnodes of a parent
@node: child node (int, lvalue) @fdt: FDT blob (const void ∗) @parent: parent node (int)
This is actually a wrapper around a for loop and would be used like so:
fdt_for_each_subnode(node, fdt, parent) { Use node ... }
if ((node < 0) && (node != -FDT_ERR_NOTFOUND)) { Error handling }
Note that this is implemented as a macro and @node is used as iterator in the loop. The parent variable be constant or even a literal.

30.320.1.28 fdt_get_header

```
#define fdt_get_header(
            fdt,
            field )  (fdt32_ld(&((const struct fdt_header *)(fdt))->field))
```

30.320.1.29 fdt_last_comp_version

```
#define fdt_last_comp_version(
            fdt ) (fdt_get_header(fdt, last_comp_version))
```

30.320.1.30 FDT_LAST_COMPATIBLE_VERSION

```
#define FDT_LAST_COMPATIBLE_VERSION 0x10
```

30.320.1.31 FDT_LAST_SUPPORTED_VERSION

```
#define FDT_LAST_SUPPORTED_VERSION 0x11
```

30.320.1.32 fdt_magic

```
#define fdt_magic(
            fdt ) (fdt_get_header(fdt, magic))
```

30.320.1.33 FDT_MAX_NCELLS

```
#define FDT_MAX_NCELLS 4
```

FDT_MAX_NCELLS - maximum value for #address-cells and #size-cells
This is the maximum value for #address-cells, #size-cells and similar properties that will be processed by libfdt. IEE1275 requires that OF implementations handle values up to 4. Implementations may support larger values, but in practice higher values aren't used.

30.320.1.34 FDT_MAX_PHANDLE

```
#define FDT_MAX_PHANDLE 0xfffffffe
```

30.320.1.35 fdt_off_dt_strings

```
#define fdt_off_dt_strings(
            fdt ) (fdt_get_header(fdt, off_dt_strings))
```

30.320.1.36 fdt_off_dt_struct

```
#define fdt_off_dt_struct(
            fdt ) (fdt_get_header(fdt, off_dt_struct))
```

30.320.1.37 fdt_off_mem_rsvmap

```
#define fdt_off_mem_rsvmap(
            fdt ) (fdt_get_header(fdt, off_mem_rsvmap))
```

30.320.1.38 fdt_property_string

```
#define fdt_property_string(
            fdt,
            name,
            str )  fdt_property(fdt, name, str, strlen(str)+1)
```

30.320.1.39 fdt_set_hdr_

```
#define fdt_set_hdr_(
            name )
```

Value:

```
static inline void fdt_set_##name(void *fdt, uint32_t val) \
{ \
    struct fdt_header *fdth = (struct fdt_header *)fdt; \
    fdth->name = cpu_to_fdt32(val); \
}
```

30.320.1.40 fdt_setprop_empty

```
#define fdt_setprop_empty(
            fdt,
            nodeoffset,
            name )  fdt_setprop((fdt), (nodeoffset), (name), NULL, 0)
```

30.320.1.41 fdt_setprop_string

```
#define fdt_setprop_string(
            fdt,
            nodeoffset,
            name,
            str )  fdt_setprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
```

30.320.1.42 fdt_size_dt_strings

```
#define fdt_size_dt_strings(
            fdt ) (fdt_get_header(fdt, size_dt_strings))
```

30.320.1.43 fdt_size_dt_struct

```
#define fdt_size_dt_struct(
            fdt ) (fdt_get_header(fdt, size_dt_struct))
```

30.320.1.44 fdt_totalsize

```
#define fdt_totalsize(
            fdt ) (fdt_get_header(fdt, totalsize))
```

30.320.1.45 fdt_version

```
#define fdt_version(
            fdt ) (fdt_get_header(fdt, version))
```

30.320.2 Function Documentation

30.320.2.1 fdt32_ld()

```
static uint32_t fdt32_ld (
            const fdt32_t * p )  [inline], [static]
```

30.320.2.2 fdt32_st()

```
static void fdt32_st (
            void * property,
            uint32_t value )  [inline], [static]
```

Here is the caller graph for this function:

30.320.2.3 fdt64_ld()

```
static uint64_t fdt64_ld (
            const fdt64_t * p )  [inline], [static]
```

30.320.2.4 fdt64_st()

```
static void fdt64_st (
            void * property,
            uint64_t value )  [inline], [static]
```

Here is the caller graph for this function:

30.320.2.5 fdt_add_mem_rsv()

```
int fdt_add_mem_rsv (
            void * fdt,
            uint64_t address,
            uint64_t size )
```

fdt_add_mem_rsv - add one memory reserve map entry @fdt: pointer to the device tree blob @address: 64-bit start address of the reserve map entry @size: 64-bit size of the reserved region

Adds a reserve map entry to the given blob reserving a region at address address of length size.

This function will insert data into the reserve map and will therefore change the indexes of some entries in the table.

returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new reservation entry -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

30.320.2.6 fdt_add_reservemap_entry()

```
int fdt_add_reservemap_entry (
            void * fdt,
            uint64_t addr,
            uint64_t size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.7 fdt_add_subnode()

```
int fdt_add_subnode (
            void * fdt,
            int parentoffset,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.8 fdt_add_subnode_namelen()

```
int fdt_add_subnode_namelen (
            void * fdt,
            int parentoffset,
            const char * name,
            int namelen )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.9 fdt_address_cells()

```
int fdt_address_cells (
            const void * fdt,
            int nodeoffset )
```

fdt_address_cells - retrieve address size for a bus represented in the tree @fdt: pointer to the device tree blob @nodeoffset: offset of the node to find the address size for

When the node has a valid #address-cells property, returns its value.

returns: 0 <= n < FDT_MAX_NCELLS, on success 2, if the node has no #address-cells property -FDT_ERR_↩ BADNCELLS, if the node has a badly formatted or invalid #address-cells property -FDT_ERR_BADMAGIC, -FDT↩ _ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.10 fdt_appendprop()

```
int fdt_appendprop (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.11 fdt_appendprop_addrrange()

```
int fdt_appendprop_addrrange (
            void * fdt,
            int parent,
            int nodeoffset,
            const char * name,
            uint64_t addr,
            uint64_t size )
```

Here is the call graph for this function:

30.320.2.12 fdt_appendprop_cell()

```
static int fdt_appendprop_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.13 fdt_appendprop_u32()

```
static int fdt_appendprop_u32 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.14 fdt_appendprop_u64()

```
static int fdt_appendprop_u64 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint64_t val )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.15 fdt_begin_node()

```
int fdt_begin_node (
            void * fdt,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.16 fdt_check_full()

```
int fdt_check_full (
            const void * fdt,
            size_t bufsize )
```

Here is the call graph for this function:

30.320.2.17 fdt_check_header()

```
int fdt_check_header (
            const void * fdt )
```

fdt_check_header - sanity check a device tree header @fdt: pointer to data which might be a flattened device tree fdt_check_header() checks that the given buffer contains what appears to be a flattened device tree, and that the header contains valid information (to the extent that can be determined from the header alone).
returns: 0, if the buffer appears to contain a valid device tree -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_TRUNCATED, standard meanings, as above Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.18 fdt_create()

```
int fdt_create (
            void * buf,
            int bufsize )
```

fdt_create - begin creation of a new fdt @buf: pointer to memory allocated where fdt will be created @bufsize: size of the memory space at fdt

fdt_create() is equivalent to fdt_create_with_flags() with flags=0.

returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient for a minimal fdt Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.19 fdt_create_empty_tree()

```
int fdt_create_empty_tree (
            void * buf,
            int bufsize )
```

Here is the call graph for this function:

30.320.2.20 fdt_create_with_flags()

```
int fdt_create_with_flags (
            void * buf,
            int bufsize,
            uint32_t flags )
```

fdt_create_with_flags - begin creation of a new fdt @buf: pointer to memory allocated where fdt will be created @bufsize: size of the memory space at fdt @flags: a valid combination of FDT_CREATE_FLAG_ flags, or 0.
fdt_create_with_flags() begins the process of creating a new fdt with the sequential write interface.
fdt creation process must end with fdt_finished() to produce a valid fdt.
returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient for a minimal fdt -FDT_ERR_BADFLAGS, flags is not valid Here is the caller graph for this function:

30.320.2.21 fdt_del_mem_rsv()

```
int fdt_del_mem_rsv (
            void * fdt,
            int n )
```

fdt_del_mem_rsv - remove a memory reserve map entry @fdt: pointer to the device tree blob
: entry to remove
fdt_del_mem_rsv() removes the n-th memory reserve map entry from the blob.
This function will delete data from the reservation table and will therefore change the indexes of some entries in the table.
returns: 0, on success -FDT_ERR_NOTFOUND, there is no entry of the given index (i.e. there are less than n+1 reserve map entries) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph

for this function:

30.320.2.22 fdt_del_node()

```
int fdt_del_node (
            void * fdt,
            int nodeoffset )
```

fdt_del_node - delete a node (subtree) @fdt: pointer to the device tree blob @nodeoffset: offset of the node to nop fdt_del_node() will remove the given node, including all its subnodes if any, from the blob.
This function will delete data from the blob, and will therefore change the offsets of some existing nodes.
returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_↩ ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

30.320.2.23 fdt_delprop()

```
int fdt_delprop (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.24 fdt_end_node()

```
int fdt_end_node (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.25 fdt_find_max_phandle()

```
int fdt_find_max_phandle (
            const void * fdt,
            uint32_t * phandle )
```

fdt_find_max_phandle - find and return the highest phandle in a tree @fdt: pointer to the device tree blob @phandle: return location for the highest phandle value found in the tree

fdt_find_max_phandle() finds the highest phandle value in the given device tree. The value returned in @phandle is only valid if the function returns success.

returns: 0 on success or a negative error code on failure Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.26 fdt_finish()

```
int fdt_finish (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.27 fdt_finish_reservemap()

```
int fdt_finish_reservemap (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.28 fdt_first_property_offset()

```
int fdt_first_property_offset (
            const void * fdt,
            int nodeoffset )
```

fdt_first_property_offset - find the offset of a node's first property @fdt: pointer to the device tree blob @nodeoffset: structure block offset of a node

fdt_first_property_offset() finds the first property of the node at the given structure block offset.

returns: structure block offset of the property (>=0), on success -FDT_ERR_NOTFOUND, if the requested node has no properties -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_BEGIN_NODE tag -FDT_ERR↩ _BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩

_TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.29 fdt_first_subnode()

```
int fdt_first_subnode (
            const void * fdt,
            int offset )
```

fdt_first_subnode() - get offset of first direct subnode @fdt: FDT blob @offset: Offset of node to check
Return: offset of first subnode, or -FDT_ERR_NOTFOUND if there is none Here is the call graph for this function:

30.320.2.30 fdt_generate_phandle()

```
int fdt_generate_phandle (
            const void * fdt,
            uint32_t * phandle )
```

fdt_generate_phandle - return a new, unused phandle for a device tree blob @fdt: pointer to the device tree blob @phandle: return location for the new phandle
Walks the device tree blob and looks for the highest phandle value. On success, the new, unused phandle value (one higher than the previously highest phandle value in the device tree blob) will be returned in the @phandle parameter.
Return: 0 on success or a negative error-code on failure Here is the call graph for this function:

30.320.2.31 fdt_get_alias()

```
const char * fdt_get_alias (
            const void * fdt,
            const char * name )
```

Here is the call graph for this function:

30.320.2.32 fdt_get_alias_namelen()

```
const char * fdt_get_alias_namelen (
            const void * fdt,
            const char * name,
            int namelen )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.33 fdt_get_max_phandle()

```
static uint32_t fdt_get_max_phandle (
            const void * fdt )  [inline], [static]
```

fdt_get_max_phandle - retrieves the highest phandle in a tree @fdt: pointer to the device tree blob

fdt_get_max_phandle retrieves the highest phandle in the given device tree. This will ignore badly formatted phandles, or phandles with a value of 0 or -1.

This function is deprecated in favour of fdt_find_max_phandle().

returns: the highest phandle on success 0, if no phandle was found in the device tree -1, if an error occurred Here is the call graph for this function:

30.320.2.34 fdt_get_mem_rsv()

```
int fdt_get_mem_rsv (
            const void * fdt,
            int n,
```

```
            uint64_t * address,
            uint64_t * size )
```

fdt_get_mem_rsv - retrieve one memory reserve map entry @fdt: pointer to the device tree blob
: index of reserve map entry @address: pointer to 64-bit variable to hold the start address @size: pointer to 64-bit variable to hold the size of the entry
On success, @address and @size will contain the address and size of the n-th reserve map entry from the device tree blob, in native-endian format.
returns: 0, on success -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:

30.320.2.35 fdt_get_name()

```
const char * fdt_get_name (
            const void * fdt,
            int nodeoffset,
            int * lenp )
```

fdt_get_name - retrieve the name of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of the starting node @lenp: pointer to an integer variable (will be overwritten) or NULL
fdt_get_name() retrieves the name (including unit address) of the device tree node at structure block offset nodeoffset. If lenp is non-NULL, the length of this name is also returned, in the integer pointed to by lenp.
returns: pointer to the node's name, on success If lenp is non-NULL, ∗lenp contains the length of that name (>=0) NULL, on error if lenp is non-NULL ∗lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.36 fdt_get_path()

```
int fdt_get_path (
            const void * fdt,
            int nodeoffset,
            char * buf,
            int buflen )
```

fdt_get_path - determine the full path of a node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose path to find @buf: character buffer to contain the returned path (will be overwritten) @buflen: size of the character buffer at buf

fdt_get_path() computes the full path of the node at offset nodeoffset, and records that path in the buffer at buf.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: 0, on success buf contains the absolute path of the node at nodeoffset, as a NUL-terminated string. -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOSPACE, the path of the given node is longer than (bufsize-1) characters and will not fit in the given buffer. -FDT_ERR_BADMAGIC, -FDT_↩ ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.37 fdt_get_phandle()

```
uint32_t fdt_get_phandle (
            const void * fdt,
            int nodeoffset )
```

fdt_get_phandle - retrieve the phandle of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of the node

fdt_get_phandle() retrieves the phandle of the device tree node at structure block offset nodeoffset.

returns: the phandle of the node at nodeoffset, on success (!= 0, != -1) 0, if the node has no phandle, or another error occurs Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.38 fdt_get_property()

```
const struct fdt_property * fdt_get_property (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.39 fdt_get_property_by_offset()

```
const struct fdt_property * fdt_get_property_by_offset (
            const void * fdt,
            int offset,
            int * lenp )
```

fdt_get_property_by_offset - retrieve the property at a given offset @fdt: pointer to the device tree blob @offset: offset of the property to retrieve @lenp: pointer to an integer variable (will be overwritten) or NULL fdt_get_property_by_offset() retrieves a pointer to the fdt_property structure within the device tree blob at the given offset. If lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by lenp.
Note that this code only works on device tree versions >= 16. fdt_getprop() works on all versions.
returns: pointer to the structure representing the property if lenp is non-NULL, ∗lenp contains the length of the property value (>=0) NULL, on error if lenp is non-NULL, ∗lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_PROP tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_↩ BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

30.320.2.40 fdt_get_property_namelen()

```
const struct fdt_property * fdt_get_property_namelen (
```

```
const void * fdt,
int nodeoffset,
const char * name,
int namelen,
int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.41 fdt_get_property_w()

```
static struct fdt_property * fdt_get_property_w (
            void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.42 fdt_get_string()

```
const char * fdt_get_string (
            const void * fdt,
            int stroffset,
            int * lenp )
```

fdt_get_string - retrieve a string from the strings block of a device tree @fdt: pointer to the device tree blob @stroffset: offset of the string within the strings block (native endian) @lenp: optional pointer to return the string's length

fdt_get_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt, and optionally also returns the string's length in ∗lenp.

returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.43 fdt_getprop()

```
const void * fdt_getprop (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:

30.320.2.44 fdt_getprop_by_offset()

```
const void * fdt_getprop_by_offset (
            const void * fdt,
            int offset,
            const char ** namep,
            int * lenp )
```

fdt_getprop_by_offset - retrieve the value of a property at a given offset @fdt: pointer to the device tree blob @offset: offset of the property to read @namep: pointer to a string variable (will be overwritten) or NULL @lenp: pointer to an integer variable (will be overwritten) or NULL

fdt_getprop_by_offset() retrieves a pointer to the value of the property at structure block offset 'offset' (this will be a pointer to within the device blob itself, not a copy of the value). If lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by lenp. If namep is non-NULL, the property's namne will also be returned in the char $*$ pointed to by namep (this will be a pointer to within the device tree's string block, not a new copy of the name).

returns: pointer to the property's value if lenp is non-NULL, $*$lenp contains the length of the property value ($>$=0) if namep is non-NULL $*$namep contiains a pointer to the property name. NULL, on error if lenp is non-NULL, $*$lenp contains an error code ($<$0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_PROP tag -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.45 fdt_getprop_namelen()

```
const void * fdt_getprop_namelen (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.46 fdt_getprop_namelen_w()

```
static void * fdt_getprop_namelen_w (
            void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            int * lenp )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.47 fdt_getprop_w()

```
static void * fdt_getprop_w (
            void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.48 fdt_header_size()

```
size_t fdt_header_size (
            const void * fdt )
```

fdt_header_size - return the size of the tree's header @fdt: pointer to a flattened device tree
Return: size of DTB header in bytes Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.49 fdt_header_size_()

```
size_t fdt_header_size_ (
            uint32_t version )
```

fdt_header_size_ - internal function to get header size from a version number

Version

: devicetree version number

Return: size of DTB header in bytes Here is the caller graph for this function:

30.320.2.50 fdt_move()

```
int fdt_move (
            const void * fdt,
            void * buf,
            int bufsize )
```

fdt_move - move a device tree around in memory @fdt: pointer to the device tree to move @buf: pointer to memory where the device is to be moved @bufsize: size of the memory space at buf

fdt_move() relocates, if possible, the device tree blob located at fdt to the buffer at buf of size bufsize. The buffer may overlap with the existing device tree blob at fdt. Therefore, fdt_move(fdt, fdt, fdt_totalsize(fdt)) should always succeed.

returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient to contain the device tree -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the caller graph for this function:

30.320.2.51 fdt_next_node()

```
int fdt_next_node (
            const void * fdt,
            int offset,
            int * depth )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.52 fdt_next_property_offset()

```
int fdt_next_property_offset (
            const void * fdt,
            int offset )
```

fdt_next_property_offset - step through a node's properties @fdt: pointer to the device tree blob @offset: structure block offset of a property

fdt_next_property_offset() finds the property immediately after the one at the given structure block offset. This will be a property of the same node as the given property.
returns: structure block offset of the next property (>=0), on success -FDT_ERR_NOTFOUND, if the given property is the last in its node -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_PROP tag -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.53 fdt_next_subnode()

```
int fdt_next_subnode (
            const void * fdt,
            int offset )
```

fdt_next_subnode() - get offset of next direct subnode @fdt: FDT blob @offset: Offset of previous subnode
After first calling fdt_first_subnode(), call this function repeatedly to get direct subnodes of a parent node.
Return: offset of next subnode, or -FDT_ERR_NOTFOUND if there are no more subnodes Here is the call graph for this function:

30.320.2.54 fdt_next_tag()

```
uint32_t fdt_next_tag (
            const void * fdt,
            int offset,
            int * nextoffset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.55 fdt_node_check_compatible()

```
int fdt_node_check_compatible (
            const void * fdt,
            int nodeoffset,
            const char * compatible )
```

fdt_node_check_compatible - check a node's compatible property @fdt: pointer to the device tree blob @nodeoffset: offset of a tree node @compatible: string to match against

fdt_node_check_compatible() returns 0 if the given node contains a @compatible property with the given string as one of its elements, it returns non-zero otherwise, or on error.

returns: 0, if the node has a 'compatible' property listing the given string 1, if the node has a 'compatible' property, but it does not list the given string -FDT_ERR_NOTFOUND, if the given node has no 'compatible' property -FDT↩_ERR_BADOFFSET, if nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_↩BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.56 fdt_node_depth()

```
int fdt_node_depth (
            const void * fdt,
            int nodeoffset )
```

fdt_node_depth - find the depth of a given node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find

fdt_node_depth() finds the depth of a given node. The root node has depth 0, its immediate subnodes depth 1 and so forth.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: depth of the node at nodeoffset (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.57 fdt_node_offset_by_compatible()

```
int fdt_node_offset_by_compatible (
            const void * fdt,
            int startoffset,
            const char * compatible )
```

fdt_node_offset_by_compatible - find nodes with a given 'compatible' value @fdt: pointer to the device tree blob @startoffset: only find nodes after this offset @compatible: 'compatible' string to match against

fdt_node_offset_by_compatible() returns the offset of the first node after startoffset, which has a 'compatible' property which lists the given compatible string; or if startoffset is -1, the very first such node in the tree.

To iterate through all nodes matching the criterion, the following idiom can be used: offset = fdt_node_offset_by_↩compatible(fdt, -1, compatible); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_↩offset_by_compatible(fdt, offset, compatible); }

Note the -1 in the first call to the function, if 0 is used here instead, the function will never locate the root node, even if it matches the criterion.

returns: structure block offset of the located node (>= 0, >startoffset), on success -FDT_ERR_NOTFOUND, no node matching the criterion exists in the tree after startoffset -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.58 fdt_node_offset_by_phandle()

```
int fdt_node_offset_by_phandle (
            const void * fdt,
            uint32_t phandle )
```

fdt_node_offset_by_phandle - find the node with a given phandle @fdt: pointer to the device tree blob @phandle: phandle value

fdt_node_offset_by_phandle() returns the offset of the node which has the given phandle value. If there is more than one node in the tree with the given phandle (an invalid tree), results are undefined.

returns: structure block offset of the located node (>= 0), on success -FDT_ERR_NOTFOUND, no node with that phandle exists -FDT_ERR_BADPHANDLE, given phandle value was invalid (0 or -1) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.59 fdt_node_offset_by_prop_value()

```
int fdt_node_offset_by_prop_value (
            const void * fdt,
            int startoffset,
            const char * propname,
            const void * propval,
            int proplen )
```

fdt_node_offset_by_prop_value - find nodes with a given property value @fdt: pointer to the device tree blob @startoffset: only find nodes after this offset @propname: property name to check @propval: property value to search for @proplen: length of the value in propval

fdt_node_offset_by_prop_value() returns the offset of the first node after startoffset, which has a property named propname whose value is of length proplen and has value equal to propval; or if startoffset is -1, the very first such node in the tree.

To iterate through all nodes matching the criterion, the following idiom can be used: offset = fdt_node_offset_↩ by_prop_value(fdt, -1, propname, propval, proplen); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_offset_by_prop_value(fdt, offset, propname, propval, proplen); }

Note the -1 in the first call to the function, if 0 is used here instead, the function will never locate the root node, even if it matches the criterion.

returns: structure block offset of the located node (>= 0, >startoffset), on success -FDT_ERR_NOTFOUND, no node matching the criterion exists in the tree after startoffset -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩

_BADSTRUCTURE, standard meanings Here is the call graph for this function:

30.320.2.60 fdt_nop_node()

```
int fdt_nop_node (
            void * fdt,
            int nodeoffset )
```

fdt_nop_node - replace a node (subtree) with nop tags @fdt: pointer to the device tree blob @nodeoffset: offset of the node to nop

fdt_nop_node() will replace a given node's representation in the blob, including all its subnodes, if any, with FDT↩_NOP tags, effectively removing it from the tree.

This function will alter only the bytes in the blob which contain the node and its properties and subnodes, and will not alter or move any other part of the tree.

returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR↩_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩_TRUNCATED, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.61 fdt_nop_property()

```
int fdt_nop_property (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.62 fdt_num_mem_rsv()

```
int fdt_num_mem_rsv (
            const void * fdt )
```

fdt_num_mem_rsv - retrieve the number of memory reserve map entries @fdt: pointer to the device tree blob
Returns the number of entries in the device tree blob's memory reservation map. This does not include the terminating 0,0 entry or any other (0,0) entries reserved for expansion.
returns: the number of entries Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.63 fdt_offset_ptr()

```
const void * fdt_offset_ptr (
            const void * fdt,
            int offset,
            unsigned int checklen )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.64 fdt_offset_ptr_w()

```
static void * fdt_offset_ptr_w (
            void * fdt,
            int offset,
            int checklen )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.65 fdt_open_into()

```
int fdt_open_into (
            const void * fdt,
            void * buf,
            int bufsize )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.66 fdt_overlay_apply()

```
int fdt_overlay_apply (
            void * fdt,
            void * fdto )
```

fdt_overlay_apply - Applies a DT overlay on a base DT @fdt: pointer to the base device tree blob @fdto: pointer to the device tree overlay blob

fdt_overlay_apply() will apply the given device tree overlay on the given base device tree.
Expect the base device tree to be modified, even if the function returns an error.
returns: 0, on success -FDT_ERR_NOSPACE, there's not enough space in the base device tree -FDT_ERR_↩ NOTFOUND, the overlay points to some inexistant nodes or properties in the base DT -FDT_ERR_BADPHANDLE, -FDT_ERR_BADOVERLAY, -FDT_ERR_NOPHANDLES, -FDT_ERR_INTERNAL, -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADOFFSET, -FDT_ERR_BADPATH, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTRUCTURE, -FDT_ERR_BADSTATE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

30.320.2.67 fdt_pack()

```
int fdt_pack (
            void * fdt )
```

Here is the call graph for this function:

30.320.2.68 fdt_parent_offset()

```
int fdt_parent_offset (
            const void * fdt,
            int nodeoffset )
```

fdt_parent_offset - find the parent of a given node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find
fdt_parent_offset() locates the parent node of a given node (that is, it finds the offset of the node which contains the node at nodeoffset as a subnode).
NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset, *twice*.

returns: structure block offset of the parent of the node at nodeoffset (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.69 fdt_path_offset()

```
int fdt_path_offset (
            const void * fdt,
            const char * path )
```

fdt_path_offset - find a tree node by its full path @fdt: pointer to the device tree blob @path: full path of the node to locate

fdt_path_offset() finds a node of a given path in the device tree. Each path component may omit the unit address portion, but the results of this are undefined if any such path component is ambiguous (that is if there are multiple nodes at the relevant level matching the given component, differentiated only by unit address).

returns: structure block offset of the node with the requested path (>=0), on success -FDT_ERR_BADPATH, given path does not begin with '/' or is invalid -FDT_ERR_NOTFOUND, if the requested node does not exist -FDT_ERR↩ _BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩

_TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.70 fdt_path_offset_namelen()

```
int fdt_path_offset_namelen (
            const void * fdt,
            const char * path,
            int namelen )
```

fdt_path_offset_namelen - find a tree node by its full path @fdt: pointer to the device tree blob @path: full path of the node to locate @namelen: number of characters of path to consider

Identical to fdt_path_offset(), but only consider the first namelen characters of path as the path name.

Return: offset of the node or negative libfdt error value otherwise Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.71 fdt_property()

```
int fdt_property (
            void * fdt,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

30.320.2.72 fdt_property_cell()

```
static int fdt_property_cell (
            void * fdt,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.73 fdt_property_placeholder()

```
int fdt_property_placeholder (
            void * fdt,
            const char * name,
            int len,
            void ** valp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.74 fdt_property_u32()

```
static int fdt_property_u32 (
            void * fdt,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.75 fdt_property_u64()

```
static int fdt_property_u64 (
            void * fdt,
            const char * name,
            uint64_t val )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.76 fdt_resize()

```
int fdt_resize (
            void * fdt,
            void * buf,
            int bufsize )
```

30.320.2.77 fdt_set_hdr_() [1/10]

```
fdt_set_hdr_ (
            boot_cpuid_phys  )
```

30.320.2.78 fdt_set_hdr_() [2/10]

```
fdt_set_hdr_ (
            last_comp_version  )
```

30.320.2.79 fdt_set_hdr_() [3/10]

```
fdt_set_hdr_ (
            magic  )
```

30.320.2.80 fdt_set_hdr_() [4/10]

```
fdt_set_hdr_ (
            off_dt_strings  )
```

30.320.2.81 fdt_set_hdr_() [5/10]

```
fdt_set_hdr_ (
            off_dt_struct  )
```

30.320.2.82 fdt_set_hdr_() [6/10]

```
fdt_set_hdr_ (
            off_mem_rsvmap  )
```

30.320.2.83 fdt_set_hdr_() [7/10]

```
fdt_set_hdr_ (
            size_dt_strings  )
```

30.320.2.84 fdt_set_hdr_() [8/10]

```
fdt_set_hdr_ (
            size_dt_struct  )
```

30.320.2.85 fdt_set_hdr_() [9/10]

```
fdt_set_hdr_ (
            totalsize  )
```

30.320.2.86 fdt_set_hdr_() [10/10]

```
fdt_set_hdr_ (
            version  )
```

30.320.2.87 fdt_set_name()

```
int fdt_set_name (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

30.320.2.88 fdt_setprop()

```
int fdt_setprop (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.89 fdt_setprop_cell()

```
static int fdt_setprop_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.90 fdt_setprop_inplace()

```
int fdt_setprop_inplace (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.91 fdt_setprop_inplace_cell()

```
static int fdt_setprop_inplace_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.92 fdt_setprop_inplace_namelen_partial()

```
int fdt_setprop_inplace_namelen_partial (
            void * fdt,
```

```
            int nodeoffset,
            const char * name,
            int namelen,
            uint32_t idx,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.93 fdt_setprop_inplace_u32()

```
static int fdt_setprop_inplace_u32 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.94 fdt_setprop_inplace_u64()

```
static int fdt_setprop_inplace_u64 (
            void * fdt,
            int nodeoffset,
```

```
const char * name,
uint64_t val )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.95 fdt_setprop_placeholder()

```
int fdt_setprop_placeholder (
            void * fdt,
            int nodeoffset,
            const char * name,
            int len,
            void ** prop_data )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.96 fdt_setprop_u32()

```
static int fdt_setprop_u32 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.97 fdt_setprop_u64()

```
static int fdt_setprop_u64 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint64_t val )  [inline], [static]
```

Here is the call graph for this function:

30.320.2.98 fdt_size_cells()

```
int fdt_size_cells (
            const void * fdt,
            int nodeoffset )
```

fdt_size_cells - retrieve address range size for a bus represented in the tree @fdt: pointer to the device tree blob @nodeoffset: offset of the node to find the address range size for

When the node has a valid #size-cells property, returns its value.

returns: 0 <= n < FDT_MAX_NCELLS, on success 1, if the node has no #size-cells property -FDT_ERR_↩ BADNCELLS, if the node has a badly formatted or invalid #size-cells property -FDT_ERR_BADMAGIC, -FDT↩ _ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, stan-

dard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.99 fdt_strerror()

```
const char * fdt_strerror (
            int errval )
```

30.320.2.100 fdt_string()

```
const char * fdt_string (
            const void * fdt,
            int stroffset )
```

fdt_string - retrieve a string from the strings block of a device tree @fdt: pointer to the device tree blob @stroffset: offset of the string within the strings block (native endian)

fdt_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt.

returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here

is the call graph for this function:

Here is the caller graph for this function:

30.320.2.101 fdt_stringlist_contains()

```
int fdt_stringlist_contains (
            const char * strlist,
            int listlen,
            const char * str )
```

fdt_stringlist_contains - check a string list property for a string @strlist: Property containing a list of strings to check @listlen: Length of property @str: String to search for

This is a utility function provided for convenience. The list contains one or more strings, each terminated by \0, as is found in a device tree "compatible" property.

Return: 1 if the string is found in the list, 0 not found, or invalid list Here is the caller graph for this function:

30.320.2.102 fdt_stringlist_count()

```
int fdt_stringlist_count (
            const void * fdt,
            int nodeoffset,
            const char * property )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.103 fdt_stringlist_get()

```
const char * fdt_stringlist_get (
            const void * fdt,
            int nodeoffset,
            const char * property,
            int index,
            int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.104 fdt_stringlist_search()

```
int fdt_stringlist_search (
            const void * fdt,
            int nodeoffset,
            const char * property,
            const char * string )
```

Here is the call graph for this function:

30.320.2.105 fdt_subnode_offset()

```
int fdt_subnode_offset (
            const void * fdt,
            int parentoffset,
            const char * name )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.106 fdt_subnode_offset_namelen()

```
int fdt_subnode_offset_namelen (
            const void * fdt,
            int parentoffset,
            const char * name,
            int namelen )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.320.2.107 fdt_supernode_atdepth_offset()

```
int fdt_supernode_atdepth_offset (
            const void * fdt,
            int nodeoffset,
            int supernodedepth,
            int * nodedepth )
```

fdt_supernode_atdepth_offset - find a specific ancestor of a node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find @supernodedepth: depth of the ancestor to find @nodedepth: pointer to an integer variable (will be overwritten) or NULL

fdt_supernode_atdepth_offset() finds an ancestor of the given node at a specific depth from the root (where the root itself has depth 0, its immediate subnodes depth 1 and so forth). So fdt_supernode_atdepth_offset(fdt, nodeoffset, 0, NULL); will always return 0, the offset of the root node. If the node at nodeoffset has depth D, then: fdt_supernode_atdepth_offset(fdt, nodeoffset, D, NULL); will return nodeoffset itself.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: structure block offset of the node at node offset's ancestor of depth supernodedepth (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOTFOUND, supernodedepth was greater than the depth of nodeoffset -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

30.321 libfdt.h

Go to the documentation of this file.

```
00001 /* SPDX-License-Identifier: GPL-2.0-or-later OR BSD-2-Clause */
00002 #ifndef LIBFDT_H
00003 #define LIBFDT_H
00004 /*
00005  * libfdt - Flat Device Tree manipulation
00006  * Copyright (C) 2006 David Gibson, IBM Corporation.
00007  */
00008
00009 #include <libfdt_env.h>
00010 #include <fdt.h>
00011
00012 #ifdef __cplusplus
00013 extern "C" {
00014 #endif
00015
00016 #define FDT_FIRST_SUPPORTED_VERSION 0x02
00017 #define FDT_LAST_COMPATIBLE_VERSION 0x10
00018 #define FDT_LAST_SUPPORTED_VERSION  0x11
00019
00020 /* Error codes: informative error codes */
00021 #define FDT_ERR_NOTFOUND    1
00022     /* FDT_ERR_NOTFOUND: The requested node or property does not exist */
00023 #define FDT_ERR_EXISTS      2
00024     /* FDT_ERR_EXISTS: Attempted to create a node or property which
00025      * already exists */
00026 #define FDT_ERR_NOSPACE     3
00027     /* FDT_ERR_NOSPACE: Operation needed to expand the device
00028      * tree, but its buffer did not have sufficient space to
00029      * contain the expanded tree. Use fdt_open_into() to move the
00030      * device tree to a buffer with more space. */
00031
00032 /* Error codes: codes for bad parameters */
00033 #define FDT_ERR_BADOFFSET   4
00034     /* FDT_ERR_BADOFFSET: Function was passed a structure block
00035      * offset which is out-of-bounds, or which points to an
00036      * unsuitable part of the structure for the operation. */
00037 #define FDT_ERR_BADPATH     5
00038     /* FDT_ERR_BADPATH: Function was passed a badly formatted path
00039      * (e.g. missing a leading / for a function which requires an
00040      * absolute path) */
00041 #define FDT_ERR_BADPHANDLE  6
```

```
    /* FDT_ERR_BADPHANDLE: Function was passed an invalid phandle.
     * This can be caused either by an invalid phandle property
     * length, or the phandle value was either 0 or -1, which are
     * not permitted. */
#define FDT_ERR_BADSTATE    7
    /* FDT_ERR_BADSTATE: Function was passed an incomplete device
     * tree created by the sequential-write functions, which is
     * not sufficiently complete for the requested operation. */

/* Error codes: codes for bad device tree blobs */
#define FDT_ERR_TRUNCATED   8
    /* FDT_ERR_TRUNCATED: FDT or a sub-block is improperly
     * terminated (overflows, goes outside allowed bounds, or
     * isn't properly terminated).  */
#define FDT_ERR_BADMAGIC    9
    /* FDT_ERR_BADMAGIC: Given "device tree" appears not to be a
     * device tree at all - it is missing the flattened device
     * tree magic number. */
#define FDT_ERR_BADVERSION  10
    /* FDT_ERR_BADVERSION: Given device tree has a version which
     * can't be handled by the requested operation.  For
     * read-write functions, this may mean that fdt_open_into() is
     * required to convert the tree to the expected version. */
#define FDT_ERR_BADSTRUCTURE    11
    /* FDT_ERR_BADSTRUCTURE: Given device tree has a corrupt
     * structure block or other serious error (e.g. misnested
     * nodes, or subnodes preceding properties). */
#define FDT_ERR_BADLAYOUT   12
    /* FDT_ERR_BADLAYOUT: For read-write functions, the given
     * device tree has it's sub-blocks in an order that the
     * function can't handle (memory reserve map, then structure,
     * then strings).  Use fdt_open_into() to reorganize the tree
     * into a form suitable for the read-write operations. */

/* "Can't happen" error indicating a bug in libfdt */
#define FDT_ERR_INTERNAL    13
    /* FDT_ERR_INTERNAL: libfdt has failed an internal assertion.
     * Should never be returned, if it is, it indicates a bug in
     * libfdt itself. */

/* Errors in device tree content */
#define FDT_ERR_BADNCELLS   14
    /* FDT_ERR_BADNCELLS: Device tree has a #address-cells, #size-cells
     * or similar property with a bad format or value */

#define FDT_ERR_BADVALUE    15
    /* FDT_ERR_BADVALUE: Device tree has a property with an unexpected
     * value. For example: a property expected to contain a string list
     * is not NUL-terminated within the length of its value. */

#define FDT_ERR_BADOVERLAY  16
    /* FDT_ERR_BADOVERLAY: The device tree overlay, while
     * correctly structured, cannot be applied due to some
     * unexpected or missing value, property or node. */

#define FDT_ERR_NOPHANDLES  17
    /* FDT_ERR_NOPHANDLES: The device tree doesn't have any
     * phandle available anymore without causing an overflow */

#define FDT_ERR_BADFLAGS    18
    /* FDT_ERR_BADFLAGS: The function was passed a flags field that
     * contains invalid flags or an invalid combination of flags. */

#define FDT_ERR_ALIGNMENT   19
    /* FDT_ERR_ALIGNMENT: The device tree base address is not 8-byte
     * aligned. */

#define FDT_ERR_MAX     19

/* constants */
#define FDT_MAX_PHANDLE 0xfffffffe
    /* Valid values for phandles range from 1 to 2^32-2. */

/**********************************************************************/
/* Low-level functions (you probably don't need these)                */
/**********************************************************************/

#ifndef SWIG /* This function is not useful in Python */
const void *fdt_offset_ptr(const void *fdt, int offset, unsigned int checklen);
#endif
static inline void *fdt_offset_ptr_w(void *fdt, int offset, int checklen)
{
    return (void *)(uintptr_t)fdt_offset_ptr(fdt, offset, checklen);
}

uint32_t fdt_next_tag(const void *fdt, int offset, int *nextoffset);

```

```
/*
 * External helpers to access words from a device tree blob. They're built
 * to work even with unaligned pointers on platforms (such as ARMv5) that don't
 * like unaligned loads and stores.
 */
static inline uint32_t fdt32_ld(const fdt32_t *p)
{
    const uint8_t *bp = (const uint8_t *)p;

    return ((uint32_t)bp[0] << 24)
        | ((uint32_t)bp[1] << 16)
        | ((uint32_t)bp[2] << 8)
        | bp[3];
}

static inline void fdt32_st(void *property, uint32_t value)
{
    uint8_t *bp = (uint8_t *)property;

    bp[0] = value >> 24;
    bp[1] = (value >> 16) & 0xff;
    bp[2] = (value >> 8) & 0xff;
    bp[3] = value & 0xff;
}

static inline uint64_t fdt64_ld(const fdt64_t *p)
{
    const uint8_t *bp = (const uint8_t *)p;

    return ((uint64_t)bp[0] << 56)
        | ((uint64_t)bp[1] << 48)
        | ((uint64_t)bp[2] << 40)
        | ((uint64_t)bp[3] << 32)
        | ((uint64_t)bp[4] << 24)
        | ((uint64_t)bp[5] << 16)
        | ((uint64_t)bp[6] << 8)
        | bp[7];
}

static inline void fdt64_st(void *property, uint64_t value)
{
    uint8_t *bp = (uint8_t *)property;

    bp[0] = value >> 56;
    bp[1] = (value >> 48) & 0xff;
    bp[2] = (value >> 40) & 0xff;
    bp[3] = (value >> 32) & 0xff;
    bp[4] = (value >> 24) & 0xff;
    bp[5] = (value >> 16) & 0xff;
    bp[6] = (value >> 8) & 0xff;
    bp[7] = value & 0xff;
}

/**********************************************************************/
/* Traversal functions                                                */
/**********************************************************************/

int fdt_next_node(const void *fdt, int offset, int *depth);

int fdt_first_subnode(const void *fdt, int offset);

int fdt_next_subnode(const void *fdt, int offset);

#define fdt_for_each_subnode(node, fdt, parent)     \
    for (node = fdt_first_subnode(fdt, parent); \
         node >= 0;                             \
         node = fdt_next_subnode(fdt, node))

/**********************************************************************/
/* General functions                                                  */
/**********************************************************************/
#define fdt_get_header(fdt, field) \
    (fdt32_ld(&((const struct fdt_header *)(fdt))->field))
#define fdt_magic(fdt)          (fdt_get_header(fdt, magic))
#define fdt_totalsize(fdt)      (fdt_get_header(fdt, totalsize))
#define fdt_off_dt_struct(fdt)      (fdt_get_header(fdt, off_dt_struct))
#define fdt_off_dt_strings(fdt)     (fdt_get_header(fdt, off_dt_strings))
#define fdt_off_mem_rsvmap(fdt)     (fdt_get_header(fdt, off_mem_rsvmap))
#define fdt_version(fdt)        (fdt_get_header(fdt, version))
#define fdt_last_comp_version(fdt)  (fdt_get_header(fdt, last_comp_version))
#define fdt_boot_cpuid_phys(fdt)    (fdt_get_header(fdt, boot_cpuid_phys))
#define fdt_size_dt_strings(fdt)    (fdt_get_header(fdt, size_dt_strings))
#define fdt_size_dt_struct(fdt)     (fdt_get_header(fdt, size_dt_struct))

#define fdt_set_hdr_(name) \
    static inline void fdt_set_##name(void *fdt, uint32_t val) \
    { \
```

```
        struct fdt_header *fdth = (struct fdt_header *)fdt; \
        fdth->name = cpu_to_fdt32(val); \
    }
fdt_set_hdr_(magic);
fdt_set_hdr_(totalsize);
fdt_set_hdr_(off_dt_struct);
fdt_set_hdr_(off_dt_strings);
fdt_set_hdr_(off_mem_rsvmap);
fdt_set_hdr_(version);
fdt_set_hdr_(last_comp_version);
fdt_set_hdr_(boot_cpuid_phys);
fdt_set_hdr_(size_dt_strings);
fdt_set_hdr_(size_dt_struct);
#undef fdt_set_hdr_

size_t fdt_header_size(const void *fdt);

size_t fdt_header_size_(uint32_t version);

int fdt_check_header(const void *fdt);

int fdt_move(const void *fdt, void *buf, int bufsize);

/**********************************************************************/
/* Read-only functions                                                */
/**********************************************************************/

int fdt_check_full(const void *fdt, size_t bufsize);

const char *fdt_get_string(const void *fdt, int stroffset, int *lenp);

const char *fdt_string(const void *fdt, int stroffset);

int fdt_find_max_phandle(const void *fdt, uint32_t *phandle);

static inline uint32_t fdt_get_max_phandle(const void *fdt)
{
    uint32_t phandle;
    int err;

    err = fdt_find_max_phandle(fdt, &phandle);
    if (err < 0)
        return (uint32_t)-1;

    return phandle;
}

int fdt_generate_phandle(const void *fdt, uint32_t *phandle);

int fdt_num_mem_rsv(const void *fdt);

int fdt_get_mem_rsv(const void *fdt, int n, uint64_t *address, uint64_t *size);

#ifndef SWIG /* Not available in Python */
int fdt_subnode_offset_namelen(const void *fdt, int parentoffset,
                    const char *name, int namelen);
#endif
int fdt_subnode_offset(const void *fdt, int parentoffset, const char *name);

#ifndef SWIG /* Not available in Python */
int fdt_path_offset_namelen(const void *fdt, const char *path, int namelen);
#endif

int fdt_path_offset(const void *fdt, const char *path);

const char *fdt_get_name(const void *fdt, int nodeoffset, int *lenp);

int fdt_first_property_offset(const void *fdt, int nodeoffset);

int fdt_next_property_offset(const void *fdt, int offset);

#define fdt_for_each_property_offset(property, fdt, node)   \
    for (property = fdt_first_property_offset(fdt, node);   \
         property >= 0;                                     \
         property = fdt_next_property_offset(fdt, property))

const struct fdt_property *fdt_get_property_by_offset(const void *fdt,
                              int offset,
                              int *lenp);

#ifndef SWIG /* Not available in Python */
const struct fdt_property *fdt_get_property_namelen(const void *fdt,
                            int nodeoffset,
                            const char *name,
                            int namelen, int *lenp);
#endif

```

```
const struct fdt_property *fdt_get_property(const void *fdt, int nodeoffset,
                        const char *name, int *lenp);
static inline struct fdt_property *fdt_get_property_w(void *fdt, int nodeoffset,
                              const char *name,
                              int *lenp)
{
    return (struct fdt_property *)(uintptr_t)
        fdt_get_property(fdt, nodeoffset, name, lenp);
}

#ifndef SWIG /* This function is not useful in Python */
const void *fdt_getprop_by_offset(const void *fdt, int offset,
                  const char **namep, int *lenp);
#endif

#ifndef SWIG /* Not available in Python */
const void *fdt_getprop_namelen(const void *fdt, int nodeoffset,
                const char *name, int namelen, int *lenp);
static inline void *fdt_getprop_namelen_w(void *fdt, int nodeoffset,
                      const char *name, int namelen,
                      int *lenp)
{
    return (void *)(uintptr_t)fdt_getprop_namelen(fdt, nodeoffset, name,
                              namelen, lenp);
}
#endif

const void *fdt_getprop(const void *fdt, int nodeoffset,
            const char *name, int *lenp);
static inline void *fdt_getprop_w(void *fdt, int nodeoffset,
                  const char *name, int *lenp)
{
    return (void *)(uintptr_t)fdt_getprop(fdt, nodeoffset, name, lenp);
}

uint32_t fdt_get_phandle(const void *fdt, int nodeoffset);

#ifndef SWIG /* Not available in Python */
const char *fdt_get_alias_namelen(const void *fdt,
                  const char *name, int namelen);
#endif

const char *fdt_get_alias(const void *fdt, const char *name);

int fdt_get_path(const void *fdt, int nodeoffset, char *buf, int buflen);

int fdt_supernode_atdepth_offset(const void *fdt, int nodeoffset,
                 int supernodedepth, int *nodedepth);

int fdt_node_depth(const void *fdt, int nodeoffset);

int fdt_parent_offset(const void *fdt, int nodeoffset);

int fdt_node_offset_by_prop_value(const void *fdt, int startoffset,
                  const char *propname,
                  const void *propval, int proplen);

int fdt_node_offset_by_phandle(const void *fdt, uint32_t phandle);

int fdt_node_check_compatible(const void *fdt, int nodeoffset,
                  const char *compatible);

int fdt_node_offset_by_compatible(const void *fdt, int startoffset,
                  const char *compatible);

int fdt_stringlist_contains(const char *strlist, int listlen, const char *str);

int fdt_stringlist_count(const void *fdt, int nodeoffset, const char *property);

int fdt_stringlist_search(const void *fdt, int nodeoffset, const char *property,
              const char *string);

const char *fdt_stringlist_get(const void *fdt, int nodeoffset,
                   const char *property, int index,
                   int *lenp);

/**********************************************************************/
/* Read-only functions (addressing related)                           */
/**********************************************************************/

#define FDT_MAX_NCELLS      4

int fdt_address_cells(const void *fdt, int nodeoffset);

int fdt_size_cells(const void *fdt, int nodeoffset);
```

```
/*********************************************************************/
/* Write-in-place functions                                          */
/*********************************************************************/

#ifndef SWIG /* Not available in Python */
int fdt_setprop_inplace_namelen_partial(void *fdt, int nodeoffset,
                    const char *name, int namelen,
                    uint32_t idx, const void *val,
                    int len);
#endif

#ifndef SWIG /* Not available in Python */
int fdt_setprop_inplace(void *fdt, int nodeoffset, const char *name,
            const void *val, int len);
#endif

static inline int fdt_setprop_inplace_u32(void *fdt, int nodeoffset,
                      const char *name, uint32_t val)
{
    fdt32_t tmp = cpu_to_fdt32(val);
    return fdt_setprop_inplace(fdt, nodeoffset, name, &tmp, sizeof(tmp));
}

static inline int fdt_setprop_inplace_u64(void *fdt, int nodeoffset,
                      const char *name, uint64_t val)
{
    fdt64_t tmp = cpu_to_fdt64(val);
    return fdt_setprop_inplace(fdt, nodeoffset, name, &tmp, sizeof(tmp));
}

static inline int fdt_setprop_inplace_cell(void *fdt, int nodeoffset,
                       const char *name, uint32_t val)
{
    return fdt_setprop_inplace_u32(fdt, nodeoffset, name, val);
}

int fdt_nop_property(void *fdt, int nodeoffset, const char *name);

int fdt_nop_node(void *fdt, int nodeoffset);

/*********************************************************************/
/* Sequential write functions                                        */
/*********************************************************************/

/* fdt_create_with_flags flags */
#define FDT_CREATE_FLAG_NO_NAME_DEDUP 0x1
    /* FDT_CREATE_FLAG_NO_NAME_DEDUP: Do not try to de-duplicate property
     * names in the fdt. This can result in faster creation times, but
     * a larger fdt. */

#define FDT_CREATE_FLAGS_ALL    (FDT_CREATE_FLAG_NO_NAME_DEDUP)

int fdt_create_with_flags(void *buf, int bufsize, uint32_t flags);

int fdt_create(void *buf, int bufsize);

int fdt_resize(void *fdt, void *buf, int bufsize);
int fdt_add_reservemap_entry(void *fdt, uint64_t addr, uint64_t size);
int fdt_finish_reservemap(void *fdt);
int fdt_begin_node(void *fdt, const char *name);
int fdt_property(void *fdt, const char *name, const void *val, int len);
static inline int fdt_property_u32(void *fdt, const char *name, uint32_t val)
{
    fdt32_t tmp = cpu_to_fdt32(val);
    return fdt_property(fdt, name, &tmp, sizeof(tmp));
}
static inline int fdt_property_u64(void *fdt, const char *name, uint64_t val)
{
    fdt64_t tmp = cpu_to_fdt64(val);
    return fdt_property(fdt, name, &tmp, sizeof(tmp));
}

#ifndef SWIG /* Not available in Python */
static inline int fdt_property_cell(void *fdt, const char *name, uint32_t val)
{
    return fdt_property_u32(fdt, name, val);
}
#endif

int fdt_property_placeholder(void *fdt, const char *name, int len, void **valp);

#define fdt_property_string(fdt, name, str) \
    fdt_property(fdt, name, str, strlen(str)+1)
int fdt_end_node(void *fdt);
int fdt_finish(void *fdt);

/*********************************************************************/
```

```
/* Read-write functions                                                 */
/************************************************************************/

int fdt_create_empty_tree(void *buf, int bufsize);
int fdt_open_into(const void *fdt, void *buf, int bufsize);
int fdt_pack(void *fdt);

int fdt_add_mem_rsv(void *fdt, uint64_t address, uint64_t size);

int fdt_del_mem_rsv(void *fdt, int n);

int fdt_set_name(void *fdt, int nodeoffset, const char *name);

int fdt_setprop(void *fdt, int nodeoffset, const char *name,
        const void *val, int len);

int fdt_setprop_placeholder(void *fdt, int nodeoffset, const char *name,
                int len, void **prop_data);

static inline int fdt_setprop_u32(void *fdt, int nodeoffset, const char *name,
                  uint32_t val)
{
    fdt32_t tmp = cpu_to_fdt32(val);
    return fdt_setprop(fdt, nodeoffset, name, &tmp, sizeof(tmp));
}

static inline int fdt_setprop_u64(void *fdt, int nodeoffset, const char *name,
                  uint64_t val)
{
    fdt64_t tmp = cpu_to_fdt64(val);
    return fdt_setprop(fdt, nodeoffset, name, &tmp, sizeof(tmp));
}

static inline int fdt_setprop_cell(void *fdt, int nodeoffset, const char *name,
                   uint32_t val)
{
    return fdt_setprop_u32(fdt, nodeoffset, name, val);
}

#define fdt_setprop_string(fdt, nodeoffset, name, str) \
    fdt_setprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)


#define fdt_setprop_empty(fdt, nodeoffset, name) \
    fdt_setprop((fdt), (nodeoffset), (name), NULL, 0)

int fdt_appendprop(void *fdt, int nodeoffset, const char *name,
           const void *val, int len);

static inline int fdt_appendprop_u32(void *fdt, int nodeoffset,
                     const char *name, uint32_t val)
{
    fdt32_t tmp = cpu_to_fdt32(val);
    return fdt_appendprop(fdt, nodeoffset, name, &tmp, sizeof(tmp));
}

static inline int fdt_appendprop_u64(void *fdt, int nodeoffset,
                     const char *name, uint64_t val)
{
    fdt64_t tmp = cpu_to_fdt64(val);
    return fdt_appendprop(fdt, nodeoffset, name, &tmp, sizeof(tmp));
}

static inline int fdt_appendprop_cell(void *fdt, int nodeoffset,
                      const char *name, uint32_t val)
{
    return fdt_appendprop_u32(fdt, nodeoffset, name, val);
}

#define fdt_appendprop_string(fdt, nodeoffset, name, str) \
    fdt_appendprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)

int fdt_appendprop_addrrange(void *fdt, int parent, int nodeoffset,
                  const char *name, uint64_t addr, uint64_t size);

int fdt_delprop(void *fdt, int nodeoffset, const char *name);

#ifndef SWIG /* Not available in Python */
int fdt_add_subnode_namelen(void *fdt, int parentoffset,
                const char *name, int namelen);
#endif

int fdt_add_subnode(void *fdt, int parentoffset, const char *name);

int fdt_del_node(void *fdt, int nodeoffset);

int fdt_overlay_apply(void *fdt, void *fdto);
```

```
/*********************************************************************/
/* Debugging / informational functions                               */
/*********************************************************************/

const char *fdt_strerror(int errval);

#ifdef __cplusplus
}
#endif

#endif /* LIBFDT_H */
```

30.322 lib/utils/libfdt/libfdt_env.h File Reference

```
#include <sbi/sbi_string.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_byteorder.h>
```

Include dependency graph for libfdt_env.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define INT_MAX ((int)(~0U >> 1))
- #define UINT_MAX ((unsigned int)~0U)
- #define INT32_MAX INT_MAX
- #define UINT32_MAX UINT_MAX
- #define FDT_FORCE
- #define FDT_BITWISE
- #define memmove sbi_memmove
- #define memcpy sbi_memcpy
- #define memcmp sbi_memcmp
- #define memchr sbi_memchr
- #define memset sbi_memset
- #define strchr sbi_strchr
- #define strrchr sbi_strrchr

- #define strcpy sbi_strcpy
- #define strncpy sbi_strncpy
- #define strcmp sbi_strcmp
- #define strncmp sbi_strncmp
- #define strlen sbi_strlen
- #define strnlen sbi_strnlen

Typedefs

- typedef be16_t FDT_BITWISE fdt16_t
- typedef be32_t FDT_BITWISE fdt32_t
- typedef be64_t FDT_BITWISE fdt64_t

Functions

- static uint16_t fdt16_to_cpu (fdt16_t x)
- static fdt16_t cpu_to_fdt16 (uint16_t x)
- static uint32_t fdt32_to_cpu (fdt32_t x)
- static fdt32_t cpu_to_fdt32 (uint32_t x)
- static uint64_t fdt64_to_cpu (fdt64_t x)
- static fdt64_t cpu_to_fdt64 (uint64_t x)

30.322.1 Macro Definition Documentation

30.322.1.1 FDT_BITWISE

```
#define FDT_BITWISE
```

30.322.1.2 FDT_FORCE

```
#define FDT_FORCE
```

30.322.1.3 INT32_MAX

```
#define INT32_MAX INT_MAX
```

30.322.1.4 INT_MAX

```
#define INT_MAX ((int)(~0U >> 1))
```

30.322.1.5 memchr

```
#define memchr sbi_memchr
```

30.322.1.6 memcmp

```
#define memcmp sbi_memcmp
```

30.322.1.7 memcpy

```
#define memcpy sbi_memcpy
```

30.322.1.8 memmove

```
#define memmove sbi_memmove
```

30.322.1.9 memset

```
#define memset sbi_memset
```

30.322.1.10 strchr

```
#define strchr sbi_strchr
```

30.322.1.11 strcmp

```
#define strcmp sbi_strcmp
```

30.322.1.12 strcpy

```
#define strcpy sbi_strcpy
```

30.322.1.13 strlen

```
#define strlen sbi_strlen
```

30.322.1.14 strncmp

```
#define strncmp sbi_strncmp
```

30.322.1.15 strncpy

```
#define strncpy sbi_strncpy
```

30.322.1.16 strnlen

```
#define strnlen sbi_strnlen
```

30.322.1.17 strrchr

```
#define strrchr sbi_strrchr
```

30.322.1.18 UINT32_MAX

```
#define UINT32_MAX UINT_MAX
```

30.322.1.19 UINT_MAX

```
#define UINT_MAX ((unsigned int)~0U)
```

30.322.2 Typedef Documentation

30.322.2.1 fdt16_t

```
typedef be16_t FDT_BITWISE fdt16_t
```

30.322.2.2 fdt32_t

```
typedef be32_t FDT_BITWISE fdt32_t
```

30.322.2.3 fdt64_t

```
typedef be64_t FDT_BITWISE fdt64_t
```

30.322.3 Function Documentation

30.322.3.1 cpu_to_fdt16()

```
static fdt16_t cpu_to_fdt16 (
            uint16_t x )  [inline], [static]
```

30.322.3.2 cpu_to_fdt32()

```
static fdt32_t cpu_to_fdt32 (
            uint32_t x ) [inline], [static]
```

Here is the caller graph for this function:

30.322.3.3 cpu_to_fdt64()

```
static fdt64_t cpu_to_fdt64 (
            uint64_t x ) [inline], [static]
```

Here is the caller graph for this function:

30.322.3.4 fdt16_to_cpu()

```
static uint16_t fdt16_to_cpu (
            fdt16_t x ) [inline], [static]
```

30.322.3.5 fdt32_to_cpu()

```
static uint32_t fdt32_to_cpu (
            fdt32_t x ) [inline], [static]
```

Here is the caller graph for this function:

30.322.3.6 fdt64_to_cpu()

```
static uint64_t fdt64_to_cpu (
            fdt64_t x )  [inline], [static]
```

Here is the caller graph for this function:

30.323 libfdt_env.h

Go to the documentation of this file.

```
/* SPDX-License-Identifier: GPL-2.0-or-later OR BSD-2-Clause */
#ifndef LIBFDT_ENV_H
#define LIBFDT_ENV_H
/*
 * libfdt - Flat Device Tree manipulation
 * Copyright (C) 2006 David Gibson, IBM Corporation.
 * Copyright 2012 Kim Phillips, Freescale Semiconductor.
 */

#include <sbi/sbi_string.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_byteorder.h>

#define INT_MAX     ((int)(~0U » 1))
#define UINT_MAX    ((unsigned int)~0U)
#define INT32_MAX   INT_MAX
#define UINT32_MAX  UINT_MAX

#ifdef __CHECKER__
#define FDT_FORCE __attribute__((force))
#define FDT_BITWISE __attribute__((bitwise))
#else
#define FDT_FORCE
#define FDT_BITWISE
#endif

#define memmove     sbi_memmove
#define memcpy      sbi_memcpy
#define memcmp      sbi_memcmp
#define memchr      sbi_memchr
#define memset      sbi_memset
#define strchr      sbi_strchr
#define strrchr     sbi_strrchr
#define strcpy      sbi_strcpy
#define strncpy     sbi_strncpy
#define strcmp      sbi_strcmp
#define strncmp     sbi_strncmp
#define strlen      sbi_strlen
#define strnlen     sbi_strnlen

typedef be16_t FDT_BITWISE fdt16_t;
typedef be32_t FDT_BITWISE fdt32_t;
typedef be64_t FDT_BITWISE fdt64_t;

static inline uint16_t fdt16_to_cpu(fdt16_t x)
{
    return (FDT_FORCE uint16_t)be16_to_cpu(x);
}

static inline fdt16_t cpu_to_fdt16(uint16_t x)
{
    return (FDT_FORCE fdt16_t)cpu_to_be16(x);
}

static inline uint32_t fdt32_to_cpu(fdt32_t x)
{
    return (FDT_FORCE uint32_t)be32_to_cpu(x);
}

static inline fdt32_t cpu_to_fdt32(uint32_t x)
{
    return (FDT_FORCE fdt32_t)cpu_to_be32(x);
}

static inline uint64_t fdt64_to_cpu(fdt64_t x)
{
    return (FDT_FORCE uint64_t)be64_to_cpu(x);
}

static inline fdt64_t cpu_to_fdt64(uint64_t x)
{
    return (FDT_FORCE fdt64_t)cpu_to_be64(x);
}

#ifdef __APPLE__
#include <AvailabilityMacros.h>

/* strnlen() is not available on Mac OS < 10.7 */
# if !defined(MAC_OS_X_VERSION_10_7) || (MAC_OS_X_VERSION_MAX_ALLOWED < \
                                         MAC_OS_X_VERSION_10_7)

#define strnlen fdt_strnlen

```

```
/*
 * fdt_strnlen: returns the length of a string or max_count - which ever is
 * smallest.
 * Input 1 string: the string whose size is to be determined
 * Input 2 max_count: the maximum value returned by this function
 * Output: length of the string or max_count (the smallest of the two)
 */
static inline size_t fdt_strnlen(const char *string, size_t max_count)
{
    const char *p = memchr(string, 0, max_count);
    return p ? p - string : max_count;
}

#endif /* !defined(MAC_OS_X_VERSION_10_7) || (MAC_OS_X_VERSION_MAX_ALLOWED <
          MAC_OS_X_VERSION_10_7) */

#endif /* __APPLE__ */

#endif /* LIBFDT_ENV_H */
```

30.324 lib/utils/libfdt/libfdt_internal.h File Reference

`#include <fdt.h>`

Include dependency graph for libfdt_internal.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define FDT_ALIGN(x, a) (((x) + (a) - 1) & ~((a) - 1))
- #define FDT_TAGALIGN(x) (FDT_ALIGN((x), FDT_TAGSIZE))
- #define FDT_RO_PROBE(fdt)
- #define FDT_SW_MAGIC (~FDT_MAGIC)
- #define FDT_ASSUME_MASK 0
- #define can_assume(_assume) can_assume_(ASSUME_ ## _assume)

Enumerations

- enum {
  ASSUME_PERFECT = 0xff , ASSUME_VALID_DTB = 1 << 0 , ASSUME_VALID_INPUT = 1 << 1 , ASSUME_LATEST = 1 << 2 ,
  ASSUME_NO_ROLLBACK = 1 << 3 , ASSUME_LIBFDT_ORDER = 1 << 4 , ASSUME_LIBFDT_FLAWLESS = 1 << 5 }

Functions

- int32_t fdt_ro_probe_ (const void *fdt)
- int fdt_check_node_offset_ (const void *fdt, int offset)
- int fdt_check_prop_offset_ (const void *fdt, int offset)
- const char * fdt_find_string_ (const char *strtab, int tabsize, const char *s)
- int fdt_node_end_offset_ (void *fdt, int nodeoffset)
- static const void * fdt_offset_ptr_ (const void *fdt, int offset)
- static void * fdt_offset_ptr_w_ (void *fdt, int offset)
- static const struct fdt_reserve_entry * fdt_mem_rsv_ (const void *fdt, int n)
- static struct fdt_reserve_entry * fdt_mem_rsv_w_ (void *fdt, int n)
- static uint32_t fdt32_ld_ (const fdt32_t *p)
- static uint64_t fdt64_ld_ (const fdt64_t *p)
- static bool can_assume_ (int mask)

30.324.1 Macro Definition Documentation

30.324.1.1 can_assume

```
#define can_assume(
            _assume ) can_assume_(ASSUME_ ## _assume)
```

helper macros for checking assumptions

30.324.1.2 FDT_ALIGN

```
#define FDT_ALIGN(
            x,
            a ) (((x) + (a) - 1) & ~((a) - 1))
```

30.324.1.3 FDT_ASSUME_MASK

```
#define FDT_ASSUME_MASK 0
```

30.324.1.4 FDT_RO_PROBE

```
#define FDT_RO_PROBE(
            fdt )
```

Value:

```
{                                   \
    int32_t totalsize_;             \
    if ((totalsize_ = fdt_ro_probe_(fdt)) < 0) \
        return totalsize_;          \
}
```

30.324.1.5 FDT_SW_MAGIC

```
#define FDT_SW_MAGIC (~FDT_MAGIC)
```

30.324.1.6 FDT_TAGALIGN

```
#define FDT_TAGALIGN(
            x ) (FDT_ALIGN((x), FDT_TAGSIZE))
```

30.324.2 Enumeration Type Documentation

30.324.2.1 anonymous enum

```
anonymous enum
```

Enumerator

ASSUME_PERFECT	

Enumerator

ASSUME_VALID_DTB	
ASSUME_VALID_INPUT	
ASSUME_LATEST	
ASSUME_NO_ROLLBACK	
ASSUME_LIBFDT_ORDER	
ASSUME_LIBFDT_FLAWLESS	

30.324.3 Function Documentation

30.324.3.1 can_assume_()

```
static bool can_assume_ (
            int mask )  [inline], [static]
```

can_assume_() - check if a particular assumption is enabled
@mask: Mask to check (ASSUME_...)

Returns

true if that assumption is enabled, else false

30.324.3.2 fdt32_ld_()

```
static uint32_t fdt32_ld_ (
            const fdt32_t * p )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.324.3.3 fdt64_ld_()

```
static uint64_t fdt64_ld_ (
            const fdt64_t * p )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.324.3.4 fdt_check_node_offset_()

```
int fdt_check_node_offset_ (
```

```
const void * fdt,
int offset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.324.3.5 fdt_check_prop_offset_()

```
int fdt_check_prop_offset_ (
            const void * fdt,
            int offset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.324.3.6 fdt_find_string_()

```
const char * fdt_find_string_ (
            const char * strtab,
            int tabsize,
            const char * s )
```

Here is the caller graph for this function:

30.324.3.7 fdt_mem_rsv_()

```
static const struct fdt_reserve_entry * fdt_mem_rsv_ (
            const void * fdt,
            int n )  [inline], [static]
```

Here is the caller graph for this function:

30.324.3.8 fdt_mem_rsv_w_()

```
static struct fdt_reserve_entry * fdt_mem_rsv_w_ (
            void * fdt,
            int n )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.324.3.9 fdt_node_end_offset_()

```
int fdt_node_end_offset_ (
            void * fdt,
            int nodeoffset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.324.3.10 fdt_offset_ptr_()

```
static const void * fdt_offset_ptr_ (
            const void * fdt,
            int offset )  [inline], [static]
```

Here is the caller graph for this function:

30.324.3.11 fdt_offset_ptr_w_()

```
static void * fdt_offset_ptr_w_ (
            void * fdt,
            int offset )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.324.3.12 fdt_ro_probe_()

```
int32_t fdt_ro_probe_ (
            const void * fdt )
```

Here is the caller graph for this function:

30.325 libfdt_internal.h

Go to the documentation of this file.

```
/* SPDX-License-Identifier: GPL-2.0-or-later OR BSD-2-Clause */
#ifndef LIBFDT_INTERNAL_H
#define LIBFDT_INTERNAL_H
/*
 * libfdt - Flat Device Tree manipulation
 * Copyright (C) 2006 David Gibson, IBM Corporation.
 */
#include <fdt.h>

#define FDT_ALIGN(x, a)      (((x) + (a) - 1) & ~((a) - 1))
#define FDT_TAGALIGN(x)      (FDT_ALIGN((x), FDT_TAGSIZE))

int32_t fdt_ro_probe_(const void *fdt);
#define FDT_RO_PROBE(fdt)                       \
    {                               \
        int32_t totalsize_;             \
        if ((totalsize_ = fdt_ro_probe_(fdt)) < 0)  \
            return totalsize_;          \
    }

int fdt_check_node_offset_(const void *fdt, int offset);
int fdt_check_prop_offset_(const void *fdt, int offset);
const char *fdt_find_string_(const char *strtab, int tabsize, const char *s);
int fdt_node_end_offset_(void *fdt, int nodeoffset);

static inline const void *fdt_offset_ptr_(const void *fdt, int offset)
{
    return (const char *)fdt + fdt_off_dt_struct(fdt) + offset;
}

static inline void *fdt_offset_ptr_w_(void *fdt, int offset)
{
    return (void *)(uintptr_t)fdt_offset_ptr_(fdt, offset);
}

static inline const struct fdt_reserve_entry *fdt_mem_rsv_(const void *fdt, int n)
{
    const struct fdt_reserve_entry *rsv_table =
        (const struct fdt_reserve_entry *)
        ((const char *)fdt + fdt_off_mem_rsvmap(fdt));
```

```

    return rsv_table + n;
}
static inline struct fdt_reserve_entry *fdt_mem_rsv_w_(void *fdt, int n)
{
    return (void *)(uintptr_t)fdt_mem_rsv_(fdt, n);
}

/*
 * Internal helpers to access tructural elements of the device tree
 * blob (rather than for exaple reading integers from within property
 * values).  We assume that we are either given a naturally aligned
 * address for the platform or if we are not, we are on a platform
 * where unaligned memory reads will be handled in a graceful manner.
 * If not the external helpers fdtXX_ld() from libfdt.h can be used
 * instead.
 */
static inline uint32_t fdt32_ld_(const fdt32_t *p)
{
    return fdt32_to_cpu(*p);
}

static inline uint64_t fdt64_ld_(const fdt64_t *p)
{
    return fdt64_to_cpu(*p);
}

#define FDT_SW_MAGIC        (~FDT_MAGIC)

/**********************************************************************/
/* Checking controls                                                  */
/**********************************************************************/

#ifndef FDT_ASSUME_MASK
#define FDT_ASSUME_MASK 0
#endif

/*
 * Defines assumptions which can be enabled. Each of these can be enabled
 * individually. For maximum safety, don't enable any assumptions!
 *
 * For minimal code size and no safety, use ASSUME_PERFECT at your own risk.
 * You should have another method of validating the device tree, such as a
 * signature or hash check before using libfdt.
 *
 * For situations where security is not a concern it may be safe to enable
 * ASSUME_SANE.
 */
enum {
    /*
     * This does essentially no checks. Only the latest device-tree
     * version is correctly handled. Inconsistencies or errors in the device
     * tree may cause undefined behaviour or crashes. Invalid parameters
     * passed to libfdt may do the same.
     *
     * If an error occurs when modifying the tree it may leave the tree in
     * an intermediate (but valid) state. As an example, adding a property
     * where there is insufficient space may result in the property name
     * being added to the string table even though the property itself is
     * not added to the struct section.
     *
     * Only use this if you have a fully validated device tree with
     * the latest supported version and wish to minimise code size.
     */
    ASSUME_PERFECT        = 0xff,

    /*
     * This assumes that the device tree is sane. i.e. header metadata
     * and basic hierarchy are correct.
     *
     * With this assumption enabled, normal device trees produced by libfdt
     * and the compiler should be handled safely. Malicious device trees and
     * complete garbage may cause libfdt to behave badly or crash. Truncated
     * device trees (e.g. those only partially loaded) can also cause
     * problems.
     *
     * Note: Only checks that relate exclusively to the device tree itself
     * (not the parameters passed to libfdt) are disabled by this
     * assumption. This includes checking headers, tags and the like.
     */
    ASSUME_VALID_DTB    = 1 << 0,

    /*
     * This builds on ASSUME_VALID_DTB and further assumes that libfdt
     * functions are called with valid parameters, i.e. not trigger
     * FDT_ERR_BADOFFSET or offsets that are out of bounds. It disables any
     * extensive checking of parameters and the device tree, making various
```

```
     * assumptions about correctness.
     *
     * It doesn't make sense to enable this assumption unless
     * ASSUME_VALID_DTB is also enabled.
     */
    ASSUME_VALID_INPUT    = 1 « 1,

    /*
     * This disables checks for device-tree version and removes all code
     * which handles older versions.
     *
     * Only enable this if you know you have a device tree with the latest
     * version.
     */
    ASSUME_LATEST      = 1 « 2,

    /*
     * This assumes that it is OK for a failed addition to the device tree,
     * due to lack of space or some other problem, to skip any rollback
     * steps (such as dropping the property name from the string table).
     * This is safe to enable in most circumstances, even though it may
     * leave the tree in a sub-optimal state.
     */
    ASSUME_NO_ROLLBACK    = 1 « 3,

    /*
     * This assumes that the device tree components appear in a 'convenient'
     * order, i.e. the memory reservation block first, then the structure
     * block and finally the string block.
     *
     * This order is not specified by the device-tree specification,
     * but is expected by libfdt. The device-tree compiler always created
     * device trees with this order.
     *
     * This assumption disables a check in fdt_open_into() and removes the
     * ability to fix the problem there. This is safe if you know that the
     * device tree is correctly ordered. See fdt_blocks_misordered_().
     */
    ASSUME_LIBFDT_ORDER  = 1 « 4,

    /*
     * This assumes that libfdt itself does not have any internal bugs. It
     * drops certain checks that should never be needed unless libfdt has an
     * undiscovered bug.
     *
     * This can generally be considered safe to enable.
     */
    ASSUME_LIBFDT_FLAWLESS    = 1 « 5,
};

static inline bool can_assume_(int mask)
{
    return FDT_ASSUME_MASK & mask;
}

#define can_assume(_assume) can_assume_(ASSUME_ ## _assume)

#endif /* LIBFDT_INTERNAL_H */
```

30.326 lib/utils/libquad/divdi3.c File Reference

```
#include <sys/cdefs.h>
#include "quad.h"
```

Include dependency graph for divdi3.c:

Functions

- __FBSDID ("$FreeBSD$")
- quad_t __divdi3 (quad_t a, quad_t b)

30.326.1 Function Documentation

30.326.1.1 __divdi3()

```
quad_t __divdi3 (
            quad_t a,
            quad_t b )
```

Here is the call graph for this function:

30.326.1.2 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.327 lib/utils/libquad/include/limits.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define CHAR_BIT 8

30.327.1 Macro Definition Documentation

30.327.1.1 CHAR_BIT

```
#define CHAR_BIT 8
```

30.328 limits.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Jessica Clarke <jrtc27@jrtc27.com>
 */

#ifndef __LIMITS_H__
#define __LIMITS_H__

#define CHAR_BIT 8

#endif
```

30.329 lib/utils/libquad/include/sys/cdefs.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define __FBSDID(s) struct __hack

30.329.1 Macro Definition Documentation

30.329.1.1 __FBSDID

```
#define __FBSDID(
            s ) struct __hack
```

30.330 cdefs.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Jessica Clarke <jrtc27@jrtc27.com>
 */

#ifndef __SYS_CDEFS_H__
#define __SYS_CDEFS_H__

#define __FBSDID(s) struct __hack

#endif
```

30.331 lib/utils/libquad/include/sys/types.h File Reference

`#include <sbi/sbi_types.h>`

Include dependency graph for types.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define _QUAD_LOWWORD 1
- #define _QUAD_HIGHWORD 0

Typedefs

- typedef unsigned long u_long
- typedef int64_t quad_t
- typedef uint64_t u_quad_t

30.331.1 Macro Definition Documentation

30.331.1.1 _QUAD_HIGHWORD

```
#define _QUAD_HIGHWORD 0
```

30.331.1.2 _QUAD_LOWWORD

```
#define _QUAD_LOWWORD 1
```

30.331.2 Typedef Documentation

30.331.2.1 quad_t

```
typedef int64_t quad_t
```

30.331.2.2 u_long

```
typedef unsigned long u_long
```

30.331.2.3 u_quad_t

```
typedef uint64_t u_quad_t
```

30.332 types.h

Go to the documentation of this file.

```
/*
 * SPDX-License-Identifier: BSD-2-Clause
 *
 * Copyright (c) 2021 Jessica Clarke <jrtc27@jrtc27.com>
 */

#ifndef __SYS_TYPES_H__
#define __SYS_TYPES_H__

#include <sbi/sbi_types.h>

typedef unsigned long u_long;

typedef int64_t quad_t;
typedef uint64_t u_quad_t;

#if __BYTE_ORDER__ == __ORDER_BIG_ENDIAN__
#define _QUAD_LOWWORD 1
#define _QUAD_HIGHWORD 0
#else
#define _QUAD_LOWWORD 0
#define _QUAD_HIGHWORD 1
#endif

#endif
```

30.333 lib/utils/libquad/moddi3.c File Reference

```
#include <sys/cdefs.h>
#include "quad.h"
```

Include dependency graph for moddi3.c:

Functions

- __FBSDID ("$FreeBSD$")
- quad_t __moddi3 (quad_t a, quad_t b)

30.333.1 Function Documentation

30.333.1.1 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.333.1.2 __moddi3()

```
quad_t __moddi3 (
            quad_t a,
            quad_t b )
```

Here is the call graph for this function:

30.334 lib/utils/libquad/qdivrem.c File Reference

```
#include <sys/cdefs.h>
```

```
#include "quad.h"
```

Include dependency graph for qdivrem.c:

Macros

- #define B (1L << HALF_BITS) /* digit base */
- #define COMBINE(a, b) (((u_long)(a) << HALF_BITS) | (b))

Typedefs

- typedef u_long digit

Functions

- __FBSDID ("$FreeBSD$")
- static void shl (digit *p, int len, int sh)
- u_quad_t __qdivrem (u_quad_t uq, u_quad_t vq, u_quad_t *arq)

30.334.1 Macro Definition Documentation

30.334.1.1 B

```
#define B (1L << HALF_BITS) /* digit base */
```

30.334.1.2 COMBINE

```
#define COMBINE(
            a,
            b ) (((u_long)(a) << HALF_BITS) | (b))
```

30.334.2 Typedef Documentation

30.334.2.1 digit

```
typedef u_long digit
```

30.334.3 Function Documentation

30.334.3.1 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.334.3.2 __qdivrem()

```
u_quad_t __qdivrem (
            u_quad_t uq,
            u_quad_t vq,
            u_quad_t * arq )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.334.3.3 shl()

```
static void shl (
            digit * p,
            int len,
            int sh )  [static]
```

Here is the caller graph for this function:

30.335 lib/utils/libquad/quad.h File Reference

```
#include <sys/types.h>
#include <limits.h>
```

Include dependency graph for quad.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- union uu

Macros

- #define H _QUAD_HIGHWORD
- #define L _QUAD_LOWWORD
- #define QUAD_BITS (sizeof(quad_t) ∗ CHAR_BIT)
- #define LONG_BITS (sizeof(long) ∗ CHAR_BIT)
- #define HALF_BITS (sizeof(long) ∗ CHAR_BIT / 2)
- #define HHALF(x) ((x) >> HALF_BITS)
- #define LHALF(x) ((x) & ((1L << HALF_BITS) - 1))
- #define LHUP(x) ((x) << HALF_BITS)

Typedefs

- typedef unsigned int qshift_t

Functions

- int __cmpdi2 (quad_t a, quad_t b)
- quad_t __divdi3 (quad_t a, quad_t b)
- quad_t __moddi3 (quad_t a, quad_t b)
- u_quad_t __qdivrem (u_quad_t u, u_quad_t v, u_quad_t ∗rem)
- int __ucmpdi2 (u_quad_t a, u_quad_t b)
- u_quad_t __udivdi3 (u_quad_t a, u_quad_t b)
- u_quad_t __umoddi3 (u_quad_t a, u_quad_t b)

30.335.1 Macro Definition Documentation

30.335.1.1 H

```
#define H _QUAD_HIGHWORD
```

30.335.1.2 HALF_BITS

```
#define HALF_BITS (sizeof(long) * CHAR_BIT / 2)
```

30.335.1.3 HHALF

```
#define HHALF(
            x ) ((x) >> HALF_BITS)
```

30.335.1.4 L

```
#define L _QUAD_LOWWORD
```

30.335.1.5 LHALF

```
#define LHALF(
            x ) ((x) & ((1L << HALF_BITS) - 1))
```

30.335.1.6 LHUP

```
#define LHUP(
            x ) ((x) << HALF_BITS)
```

30.335.1.7 LONG_BITS

```
#define LONG_BITS (sizeof(long) * CHAR_BIT)
```

30.335.1.8 QUAD_BITS

```
#define QUAD_BITS (sizeof(quad_t) * CHAR_BIT)
```

30.335.2 Typedef Documentation

30.335.2.1 qshift_t

```
typedef unsigned int qshift_t
```

30.335.3 Function Documentation

30.335.3.1 __cmpdi2()

```
int __cmpdi2 (
            quad_t a,
            quad_t b )
```

30.335.3.2 __divdi3()

```
quad_t __divdi3 (
            quad_t a,
            quad_t b )
```

Here is the call graph for this function:

30.335.3.3 __moddi3()

```
quad_t __moddi3 (
            quad_t a,
            quad_t b )
```

Here is the call graph for this function:

30.335.3.4 __qdivrem()

```
u_quad_t __qdivrem (
            u_quad_t u,
            u_quad_t v,
            u_quad_t * rem )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.335.3.5 __ucmpdi2()

```
int __ucmpdi2 (
            u_quad_t a,
            u_quad_t b )
```

30.335.3.6 __udivdi3()

```
u_quad_t __udivdi3 (
            u_quad_t a,
            u_quad_t b )
```

Here is the call graph for this function:

30.335.3.7 __umoddi3()

```
u_quad_t __umoddi3 (
            u_quad_t a,
            u_quad_t b )
```

Here is the call graph for this function:

30.336 quad.h

Go to the documentation of this file.

```
/*-
 * SPDX-License-Identifier: BSD-3-Clause
 *
 * Copyright (c) 1992, 1993
 *  The Regents of the University of California.  All rights reserved.
 *
 * This software was developed by the Computer Systems Engineering group
 * at Lawrence Berkeley Laboratory under DARPA contract BG 91-66 and
 * contributed to Berkeley.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 *  @(#)quad.h  8.1 (Berkeley) 6/4/93
 * $FreeBSD$
```

```
 */

/*
 * Quad arithmetic.
 *
 * This library makes the following assumptions:
 *
 *  - The type long long (aka quad_t) exists.
 *
 *  - A quad variable is exactly twice as long as `long'.
 *
 *  - The machine's arithmetic is two's complement.
 *
 * This library can provide 128-bit arithmetic on a machine with 128-bit
 * quads and 64-bit longs, for instance, or 96-bit arithmetic on machines
 * with 48-bit longs.
 */

#include <sys/types.h>
#include <limits.h>

/*
 * Depending on the desired operation, we view a `long long' (aka quad_t) in
 * one or more of the following formats.
 */
union uu {
    quad_t    q;     /* as a (signed) quad */
    quad_t    uq;       /* as an unsigned quad */
    long    sl[2];        /* as two signed longs */
    u_long    ul[2];        /* as two unsigned longs */
};

/*
 * Define high and low longwords.
 */
#define H       _QUAD_HIGHWORD
#define L       _QUAD_LOWWORD

/*
 * Total number of bits in a quad_t and in the pieces that make it up.
 * These are used for shifting, and also below for halfword extraction
 * and assembly.
 */
#define QUAD_BITS   (sizeof(quad_t) * CHAR_BIT)
#define LONG_BITS   (sizeof(long) * CHAR_BIT)
#define HALF_BITS   (sizeof(long) * CHAR_BIT / 2)

/*
 * Extract high and low shortwords from longword, and move low shortword of
 * longword to upper half of long, i.e., produce the upper longword of
 * ((quad_t)(x) « (number_of_bits_in_long/2)).  ('x' must actually be u_long.)
 *
 * These are used in the multiply code, to split a longword into upper
 * and lower halves, and to reassemble a product as a quad_t, shifted left
 * (sizeof(long)*CHAR_BIT/2).
 */
#define HHALF(x)    ((x) » HALF_BITS)
#define LHALF(x)    ((x) & ((1L « HALF_BITS) - 1))
#define LHUP(x)     ((x) « HALF_BITS)

int     __cmpdi2(quad_t a, quad_t b);
quad_t        __divdi3(quad_t a, quad_t b);
quad_t        __moddi3(quad_t a, quad_t b);
u_quad_t    __qdivrem(u_quad_t u, u_quad_t v, u_quad_t *rem);
int     __ucmpdi2(u_quad_t a, u_quad_t b);
u_quad_t    __udivdi3(u_quad_t a, u_quad_t b);
u_quad_t    __umoddi3(u_quad_t a, u_quad_t b);

typedef unsigned int    qshift_t;
```

30.337 lib/utils/libquad/udivdi3.c File Reference

```
#include <sys/cdefs.h>
#include "quad.h"
```

Include dependency graph for udivdi3.c:

Functions

- __FBSDID ("$FreeBSD$")
- u_quad_t __udivdi3 (u_quad_t a, u_quad_t b)

30.337.1 Function Documentation

30.337.1.1 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.337.1.2 __udivdi3()

```
u_quad_t __udivdi3 (
            u_quad_t a,
            u_quad_t b )
```

Here is the call graph for this function:

30.338 lib/utils/libquad/umoddi3.c File Reference

```
#include <sys/cdefs.h>
```

```
#include "quad.h"
```

Include dependency graph for umoddi3.c:

Functions

- __FBSDID ("$FreeBSD$")
- u_quad_t __umoddi3 (u_quad_t a, u_quad_t b)

30.338.1 Function Documentation

30.338.1.1 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.338.1.2 __umoddi3()

```
u_quad_t __umoddi3 (
            u_quad_t a,
            u_quad_t b )
```

Here is the call graph for this function:

30.339 lib/utils/mailbox/fdt_mailbox.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/mailbox/fdt_mailbox.h>
```

Include dependency graph for fdt_mailbox.c:

Functions

- static int fdt_mbox_controller_find (const void ∗fdt, int nodeoff, struct mbox_controller ∗∗out_mbox)
- int fdt_mailbox_request_chan (const void ∗fdt, int nodeoff, int index, struct mbox_chan ∗∗out_chan)
- int fdt_mailbox_simple_xlate (struct mbox_controller ∗mbox, const struct fdt_phandle_args ∗pargs, u32 ∗out_chan_args)

Variables

- const struct fdt_driver ∗const fdt_mailbox_drivers [ ]

30.339.1 Function Documentation

30.339.1.1 fdt_mailbox_request_chan()

```
int fdt_mailbox_request_chan (
            const void * fdt,
            int nodeoff,
            int index,
            struct mbox_chan ** out_chan )
```

Request a mailbox channel using "mboxes" DT property of client DT node Here is the call graph for this function:

Here is the caller graph for this function:

30.339.1.2 fdt_mailbox_simple_xlate()

```
int fdt_mailbox_simple_xlate (
            struct mbox_controller * mbox,
            const struct fdt_phandle_args * pargs,
            u32 * out_chan_args )
```

Simple xlate function to convert one mailbox FDT cell into channel args

30.339.1.3 fdt_mbox_controller_find()

```
static int fdt_mbox_controller_find (
            const void * fdt,
            int nodeoff,
            struct mbox_controller ** out_mbox )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.339.2 Variable Documentation

30.339.2.1 fdt_mailbox_drivers

```
const struct fdt_driver* const fdt_mailbox_drivers[]  [extern]
```

30.340 lib/utils/mailbox/fdt_mailbox_rpmi_shmem.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_timer.h>
#include <sbi/riscv_io.h>
#include <sbi/riscv_locks.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/mailbox/mailbox.h>
#include <sbi_utils/mailbox/fdt_mailbox.h>
#include <sbi_utils/mailbox/rpmi_mailbox.h>
```

Include dependency graph for fdt_mailbox_rpmi_shmem.c:

Macros

- #define RPMI_BASE_VERSION_MIN RPMI_VERSION(1, 0)

- #define GET_SERVICEGROUP_ID(msg)
- #define GET_SERVICE_ID(msg)
- #define GET_FLAGS(msg)
- #define GET_MESSAGE_ID(msg)
- #define MAKE_MESSAGE_ID(__group_id, __service_id, __flags)
- #define GET_DLEN(msg)
- #define GET_TOKEN(msg)
- #define GET_MESSAGE_TYPE(msg)
- #define to_srvgrp_chan(mbox_chan) container_of(mbox_chan, struct rpmi_srvgrp_chan, chan);

Enumerations

- enum rpmi_queue_type { RPMI_QUEUE_TYPE_REQ = 0 , RPMI_QUEUE_TYPE_ACK = 1 }
- enum rpmi_queue_idx { RPMI_QUEUE_IDX_A2P_REQ = 0 , RPMI_QUEUE_IDX_P2A_ACK = 1 , RPMI_QUEUE_IDX_P2A_REQ = 2 , RPMI_QUEUE_IDX_A2P_ACK = 3 , RPMI_QUEUE_IDX_MAX_COUNT }
- enum rpmi_reg_idx { RPMI_REG_IDX_DB_REG = 0 , RPMI_REG_IDX_MAX_COUNT }

Functions

- static bool __smq_queue_full (struct smq_queue_ctx ∗qctx)
- static bool __smq_queue_empty (struct smq_queue_ctx ∗qctx)
- static int __smq_rx (struct smq_queue_ctx ∗qctx, u32 slot_size, u32 service_group_id, struct mbox_xfer ∗xfer)
- static int __smq_tx (struct smq_queue_ctx ∗qctx, struct rpmi_mb_regs ∗mb_regs, u32 slot_size, u32 service_group_id, struct mbox_xfer ∗xfer)
- static int smq_rx (struct rpmi_shmem_mbox_controller ∗mctl, u32 queue_id, u32 service_group_id, struct mbox_xfer ∗xfer)
- static int smq_tx (struct rpmi_shmem_mbox_controller ∗mctl, u32 queue_id, u32 service_group_id, struct mbox_xfer ∗xfer)
- static int rpmi_get_platform_info (struct rpmi_shmem_mbox_controller ∗mctl)
- static int smq_base_get_two_u32 (struct rpmi_shmem_mbox_controller ∗mctl, u32 service_id, u32 ∗inarg, u32 ∗outvals)
- static int rpmi_shmem_mbox_xfer (struct mbox_chan ∗chan, struct mbox_xfer ∗xfer)
- static struct mbox_chan ∗ rpmi_shmem_mbox_request_chan (struct mbox_controller ∗mbox, u32 ∗chan_↩ args)
- static void rpmi_shmem_mbox_free_chan (struct mbox_controller ∗mbox, struct mbox_chan ∗chan)
- static int rpmi_shmem_transport_init (struct rpmi_shmem_mbox_controller ∗mctl, const void ∗fdt, int nodeoff)
- static int rpmi_shmem_mbox_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- struct rpmi_mb_regs __packed
- struct fdt_mailbox fdt_mailbox_rpmi_shmem
- static const struct fdt_match rpmi_shmem_mbox_match [ ]

30.340.1 Macro Definition Documentation

30.340.1.1 GET_DLEN

```
#define GET_DLEN(
            msg )
```

Value:

```
({                          \
    struct rpmi_message *mbuf = msg;     \
    le16_to_cpu(mbuf->header.datalen); \
})
```

30.340.1.2 GET_FLAGS

```
#define GET_FLAGS(
            msg )
```

Value:

```
({                          \
    struct rpmi_message *mbuf = msg;     \
    mbuf->header.flags;           \
})
```

30.340.1.3 GET_MESSAGE_ID

```
#define GET_MESSAGE_ID(
            msg )
```

Value:

```
({                          \
    struct rpmi_message *mbuf = msg;     \
    ((u32)mbuf->header.flags « (RPMI_MSG_FLAGS_OFFSET * 8)) | \
    ((u32)mbuf->header.service_id « (RPMI_MSG_SERVICE_ID_OFFSET * 8)) | \
    ((u32)le16_to_cpu(mbuf->header.servicegroup_id)); \
})
```

30.340.1.4 GET_MESSAGE_TYPE

```
#define GET_MESSAGE_TYPE(
            msg )
```

Value:

```
({                                  \
    uint8_t flags = *((uint8_t *)msg + RPMI_MSG_FLAGS_OFFSET);    \
    ((flags & RPMI_MSG_FLAGS_TYPE) » RPMI_MSG_FLAGS_TYPE_POS));   \
})
```

30.340.1.5 GET_SERVICE_ID

```
#define GET_SERVICE_ID(
            msg )
```

Value:

```
({                          \
    struct rpmi_message *mbuf = msg;     \
    mbuf->header.service_id;         \
})
```

30.340.1.6 GET_SERVICEGROUP_ID

```
#define GET_SERVICEGROUP_ID(
            msg )
```

Value:

```
({                          \
    struct rpmi_message *mbuf = msg;     \
    le16_to_cpu(mbuf->header.servicegroup_id);\
})
```

30.340.1.7 GET_TOKEN

```
#define GET_TOKEN(
            msg )
```

Value:

```
({                          \
    struct rpmi_message *mbuf = msg;     \
    le16_to_cpu(mbuf->header.token); \
})
```

30.340.1.8 MAKE_MESSAGE_ID

```
#define MAKE_MESSAGE_ID(
            __group_id,
            __service_id,
            __flags )
```

Value:

```
({                                  \
    u32 __ret = 0;                  \
    __ret |= (u32)(__group_id) « (RPMI_MSG_SERVICEGROUP_ID_OFFSET * 8); \
    __ret |= (u32)(__service_id) « (RPMI_MSG_SERVICE_ID_OFFSET * 8); \
    __ret |= (u32)(__flags) « (RPMI_MSG_FLAGS_OFFSET * 8); \
    __ret;                          \
})
```

30.340.1.9 RPMI_BASE_VERSION_MIN

```
#define RPMI_BASE_VERSION_MIN RPMI_VERSION(1, 0)
```

Minimum Base group version required

30.340.1.10 to_srvgrp_chan

```
#define to_srvgrp_chan(
            mbox_chan )  container_of(mbox_chan, struct rpmi_srvgrp_chan, chan);
```

30.340.2 Enumeration Type Documentation

30.340.2.1 rpmi_queue_idx

```
enum rpmi_queue_idx
```

Enumerator

RPMI_QUEUE_IDX_A2P_REQ	
RPMI_QUEUE_IDX_P2A_ACK	
RPMI_QUEUE_IDX_P2A_REQ	
RPMI_QUEUE_IDX_A2P_ACK	
RPMI_QUEUE_IDX_MAX_COUNT	

30.340.2.2 rpmi_queue_type

```
enum rpmi_queue_type
```

Enumerator

RPMI_QUEUE_TYPE_REQ	
RPMI_QUEUE_TYPE_ACK	

30.340.2.3 rpmi_reg_idx

```
enum rpmi_reg_idx
```

Enumerator

RPMI_REG_IDX_DB_REG	
RPMI_REG_IDX_MAX_COUNT	

30.340.3 Function Documentation

30.340.3.1 __smq_queue_empty()

```
static bool __smq_queue_empty (
            struct smq_queue_ctx * qctx )  [static]
```

Here is the caller graph for this function:

30.340.3.2 __smq_queue_full()

```
static bool __smq_queue_full (
            struct smq_queue_ctx * qctx )  [static]
```

Here is the caller graph for this function:

30.340.3.3 __smq_rx()

```
static int __smq_rx (
            struct smq_queue_ctx * qctx,
            u32 slot_size,
            u32 service_group_id,
            struct mbox_xfer * xfer )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.3.4 __smq_tx()

```
static int __smq_tx (
            struct smq_queue_ctx * qctx,
```

```
struct rpmi_mb_regs * mb_regs,
u32 slot_size,
u32 service_group_id,
struct mbox_xfer * xfer )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.3.5 rpmi_get_platform_info()

```
static int rpmi_get_platform_info (
            struct rpmi_shmem_mbox_controller * mctl )  [static]
```

platform string can occupy max possible size max possible space in the message data as per the formatHere is the call graph for this function:

Here is the caller graph for this function:

30.340.3.6 rpmi_shmem_mbox_free_chan()

```
static void rpmi_shmem_mbox_free_chan (
            struct mbox_controller * mbox,
            struct mbox_chan * chan )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.3.7 rpmi_shmem_mbox_init()

```
static int rpmi_shmem_mbox_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.340.3.8 rpmi_shmem_mbox_request_chan()

```
static struct mbox_chan * rpmi_shmem_mbox_request_chan (
            struct mbox_controller * mbox,
            u32 * chan_args )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.3.9 rpmi_shmem_mbox_xfer()

```
static int rpmi_shmem_mbox_xfer (
            struct mbox_chan * chan,
            struct mbox_xfer * xfer )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.3.10 rpmi_shmem_transport_init()

```
static int rpmi_shmem_transport_init (
            struct rpmi_shmem_mbox_controller * mctl,
            const void * fdt,
            int nodeoff )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.3.11 smq_base_get_two_u32()

```
static int smq_base_get_two_u32 (
            struct rpmi_shmem_mbox_controller * mctl,
            u32 service_id,
            u32 * inarg,
            u32 * outvals )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.3.12 smq_rx()

```
static int smq_rx (
            struct rpmi_shmem_mbox_controller * mctl,
            u32 queue_id,
            u32 service_group_id,
            struct mbox_xfer * xfer )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.3.13 smq_tx()

```
static int smq_tx (
            struct rpmi_shmem_mbox_controller * mctl,
            u32 queue_id,
            u32 service_group_id,
            struct mbox_xfer * xfer )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.340.4 Variable Documentation

30.340.4.1 __packed

```
struct rpmi_mb_regs __packed
```

30.340.4.2 fdt_mailbox_rpmi_shmem

```
struct fdt_mailbox fdt_mailbox_rpmi_shmem
```

Initial value:

```
= {
    .driver = {
        .match_table = rpmi_shmem_mbox_match,
        .init = rpmi_shmem_mbox_init,
        .experimental = true,
    },
    .xlate = fdt_mailbox_simple_xlate,
}
```

30.340.4.3 rpmi_shmem_mbox_match

```
const struct fdt_match rpmi_shmem_mbox_match[] [static]
```

Initial value:

```
= {
    { .compatible = "riscv,rpmi-shmem-mbox" },
    { },
}
```

30.341 lib/utils/mailbox/mailbox.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_string.h>
#include <sbi_utils/mailbox/mailbox.h>
```

Include dependency graph for mailbox.c:

Functions

- static SBI_LIST_HEAD (mbox_list)
- struct mbox_controller ∗ mbox_controller_find (unsigned int id)
- int mbox_controller_add (struct mbox_controller ∗mbox)
- void mbox_controller_remove (struct mbox_controller ∗mbox)
- struct mbox_chan ∗ mbox_controller_request_chan (struct mbox_controller ∗mbox, u32 ∗chan_args)
- void mbox_controller_free_chan (struct mbox_chan ∗chan)
- int mbox_chan_xfer (struct mbox_chan ∗chan, struct mbox_xfer ∗xfer)
- int mbox_chan_get_attribute (struct mbox_chan ∗chan, int attr_id, void ∗out_value)
- int mbox_chan_set_attribute (struct mbox_chan ∗chan, int attr_id, void ∗new_value)

30.341.1 Function Documentation

30.341.1.1 mbox_chan_get_attribute()

```
int mbox_chan_get_attribute (
            struct mbox_chan * chan,
            int attr_id,
            void * out_value )
```

Get an attribute of mailbox channel

30.341.1.2 mbox_chan_set_attribute()

```
int mbox_chan_set_attribute (
            struct mbox_chan * chan,
            int attr_id,
            void * new_value )
```

Set an attribute of mailbox channel

30.341.1.3 mbox_chan_xfer()

```
int mbox_chan_xfer (
            struct mbox_chan * chan,
            struct mbox_xfer * xfer )
```

Data transfer over mailbox channel Here is the call graph for this function:

Here is the caller graph for this function:

30.341.1.4 mbox_controller_add()

```
int mbox_controller_add (
            struct mbox_controller * mbox )
```

Register mailbox controller Here is the call graph for this function:

Here is the caller graph for this function:

30.341.1.5 mbox_controller_find()

```
struct mbox_controller * mbox_controller_find (
            unsigned int id )
```

Find a registered mailbox controller Here is the caller graph for this function:

30.341.1.6 mbox_controller_free_chan()

```
void mbox_controller_free_chan (
            struct mbox_chan * chan )
```

Free a mailbox channel Here is the call graph for this function:

Here is the caller graph for this function:

30.341.1.7 mbox_controller_remove()

```
void mbox_controller_remove (
            struct mbox_controller * mbox )
```

Un-register mailbox controller Here is the call graph for this function:

Here is the caller graph for this function:

30.341.1.8 mbox_controller_request_chan()

```
struct mbox_chan * mbox_controller_request_chan (
            struct mbox_controller * mbox,
            u32 * chan_args )
```

Request a mailbox channel Here is the call graph for this function:

Here is the caller graph for this function:

30.341.1.9 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            mbox_list  )  [static]
```

30.342 lib/utils/mailbox/rpmi_mailbox.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/mailbox/mailbox.h>
#include <sbi_utils/mailbox/rpmi_mailbox.h>
```

Include dependency graph for rpmi_mailbox.c:

Functions

- int rpmi_xlate_error (enum rpmi_error error)

- int rpmi_normal_request_with_status (struct mbox_chan ∗chan, u32 service_id, void ∗req, u32 req_words, u32 req_endian_words, void ∗resp, u32 resp_words, u32 resp_endian_words)
- int rpmi_posted_request (struct mbox_chan ∗chan, u32 service_id, void ∗req, u32 req_words, u32 req_↩ endian_words)

30.342.1 Function Documentation

30.342.1.1 rpmi_normal_request_with_status()

```
int rpmi_normal_request_with_status (
            struct mbox_chan * chan,
            u32 service_id,
            void * req,
            u32 req_words,
            u32 req_endian_words,
            void * resp,
            u32 resp_words,
            u32 resp_endian_words )
```

Typical RPMI normal request with at least status code in response Here is the call graph for this function:

Here is the caller graph for this function:

30.342.1.2 rpmi_posted_request()

```
int rpmi_posted_request (
            struct mbox_chan * chan,
            u32 service_id,
            void * req,
            u32 req_words,
            u32 req_endian_words )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.342.1.3 rpmi_xlate_error()

```
int rpmi_xlate_error (
            enum rpmi_error error )
```

Convert RPMI error to SBI error Here is the caller graph for this function:

30.343 lib/utils/mpxy/fdt_mpxy.c File Reference

```
#include <sbi_utils/mpxy/fdt_mpxy.h>
```

Include dependency graph for fdt_mpxy.c:

Functions

- void fdt_mpxy_init (const void ∗fdt)

Variables

- const struct fdt_driver ∗const fdt_mpxy_drivers [ ]

30.343.1 Function Documentation

30.343.1.1 fdt_mpxy_init()

```
void fdt_mpxy_init (
            const void * fdt )
```

Here is the call graph for this function:

30.343.2 Variable Documentation

30.343.2.1 fdt_mpxy_drivers

```
const struct fdt_driver* const fdt_mpxy_drivers[] [extern]
```

30.344 lib/utils/mpxy/fdt_mpxy_rpmi_mbox.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_mpxy.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/mpxy/fdt_mpxy.h>
#include <sbi_utils/mailbox/fdt_mailbox.h>
#include <sbi_utils/mailbox/rpmi_msgprot.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for fdt_mpxy_rpmi_mbox.c:

Macros

- #define RPMI_MAJOR_VER (1)
- #define RPMI_MINOR_VER (0)
- #define attr_id2index(attr_id) (attr_id - SBI_MPXY_ATTR_MSGPROTO_ATTR_START)
- #define assert_field_offset(field, attr_offset)

Enumerations

- enum mpxy_msgprot_rpmi_attr_id { MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_ID = SBI_MPXY↩_ATTR_MSGPROTO_ATTR_START , MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_VERSION , MPXY_MSGPROT_RPMI_ATTR_MAX_ID }

Functions

- assert_field_offset (servicegrp_id, MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_ID)
- static struct rpmi_service_data ∗ mpxy_find_rpmi_srvid (u32 message_id, struct mpxy_mbox_data ∗mbox↩_data)
- static void mpxy_copy_attrs (u32 ∗outmem, u32 ∗inmem, u32 count)
- static int mpxy_mbox_read_attributes (struct sbi_mpxy_channel ∗channel, u32 ∗outmem, u32 base_attr_id, u32 attr_count)
- static int mpxy_check_write_attr (u32 attr_id, u32 attr_val)
- static void mpxy_write_attr (struct mpxy_rpmi_channel_attrs ∗attrs, u32 attr_id, u32 attr_val)
- static int mpxy_mbox_write_attributes (struct sbi_mpxy_channel ∗channel, u32 ∗outmem, u32 base_attr_id, u32 attr_count)
- static int __mpxy_mbox_send_message (struct sbi_mpxy_channel ∗channel, u32 message_id, void ∗tx, u32 tx_len, void ∗rx, u32 rx_max_len, unsigned long ∗ack_len)
- static int mpxy_mbox_send_message_withresp (struct sbi_mpxy_channel ∗channel, u32 message_id, void ∗tx, u32 tx_len, void ∗rx, u32 rx_max_len, unsigned long ∗ack_len)

- static int mpxy_mbox_send_message_withoutresp (struct sbi_mpxy_channel *channel, u32 message_id, void *tx, u32 tx_len)
- static int mpxy_mbox_get_notifications (struct sbi_mpxy_channel *channel, void *eventsbuf, u32 bufsize, unsigned long *events_len)
- static int mpxy_mbox_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static struct rpmi_service_data clock_services [ ]
- static struct mpxy_mbox_data clock_data
- static const struct fdt_match mpxy_mbox_match [ ]
- struct fdt_driver fdt_mpxy_rpmi_mbox

30.344.1 Macro Definition Documentation

30.344.1.1 assert_field_offset

```
#define assert_field_offset(
            field,
            attr_offset )
```

Value:

```
    _Static_assert(                          \
        ((offsetof(struct mpxy_rpmi_channel_attrs, field)) /    \
         sizeof(u32)) == (attr_offset - SBI_MPXY_ATTR_MSGPROTO_ATTR_START),\
        "field " #field                      \
        " from struct mpxy_rpmi_channel_attrs invalid offset, expected " #attr_offset)
```

Make sure all attributes are packed for direct memcpy

30.344.1.2 attr_id2index

```
#define attr_id2index(
            attr_id ) (attr_id - SBI_MPXY_ATTR_MSGPROTO_ATTR_START)
```

Convert the mpxy attribute ID to attribute array index

30.344.1.3 RPMI_MAJOR_VER

```
#define RPMI_MAJOR_VER (1)
```

30.344.1.4 RPMI_MINOR_VER

```
#define RPMI_MINOR_VER (0)
```

30.344.2 Enumeration Type Documentation

30.344.2.1 mpxy_msgprot_rpmi_attr_id

```
enum mpxy_msgprot_rpmi_attr_id
```

Enumerator

MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_ID	
MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_VERSION	
MPXY_MSGPROT_RPMI_ATTR_MAX_ID	

30.344.3 Function Documentation

30.344.3.1 __mpxy_mbox_send_message()

```
static int __mpxy_mbox_send_message (
            struct sbi_mpxy_channel * channel,
```

```
            u32 message_id,
            void * tx,
            u32 tx_len,
            void * rx,
            u32 rx_max_len,
            unsigned long * ack_len )  [static]
```

message data size to be sent is in the supported service data size range
MPXY buffer cannnot hold service min response dataHere is the call graph for this function:

Here is the caller graph for this function:

30.344.3.2 assert_field_offset()

```
assert_field_offset (
            servicegrp_id ,
            MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_ID  )
```

30.344.3.3 mpxy_check_write_attr()

```
static int mpxy_check_write_attr (
            u32 attr_id,
            u32 attr_val )  [static]
```

Verify the channel standard attribute wrt to write permission and the value to be set if valid or not. Only attributes needs to be checked which are defined Read/Write permission. Other with Readonly permission will result in error. Attributes values to be written must also be checked because before writing a range of attributes, we need to make sure that either complete range of attributes is written successfully or not at all. All RO access attributes falls under defaultHere is the caller graph for this function:

30.344.3.4 mpxy_copy_attrs()

```
static void mpxy_copy_attrs (
            u32 * outmem,
            u32 * inmem,
            u32 count )  [static]
```

Copy attributes word size Here is the caller graph for this function:

30.344.3.5 mpxy_find_rpmi_srvid()

```
static struct rpmi_service_data * mpxy_find_rpmi_srvid (
            u32 message_id,
            struct mpxy_mbox_data * mbox_data )  [static]
```

Discover the RPMI service data using message_id MPXY message_id == RPMI service_id Here is the caller graph for this function:

30.344.3.6 mpxy_mbox_get_notifications()

```
static int mpxy_mbox_get_notifications (
            struct sbi_mpxy_channel * channel,
            void * eventsbuf,
            u32 bufsize,
            unsigned long * events_len )  [static]
```

Here is the caller graph for this function:

30.344.3.7 mpxy_mbox_init()

```
static int mpxy_mbox_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.344.3.8 mpxy_mbox_read_attributes()

```
static int mpxy_mbox_read_attributes (
            struct sbi_mpxy_channel * channel,
            u32 * outmem,
            u32 base_attr_id,
            u32 attr_count )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.344.3.9 mpxy_mbox_send_message_withoutresp()

```
static int mpxy_mbox_send_message_withoutresp (
            struct sbi_mpxy_channel * channel,
            u32 message_id,
            void * tx,
            u32 tx_len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.344.3.10 mpxy_mbox_send_message_withresp()

```
static int mpxy_mbox_send_message_withresp (
            struct sbi_mpxy_channel * channel,
            u32 message_id,
            void * tx,
            u32 tx_len,
            void * rx,
            u32 rx_max_len,
            unsigned long * ack_len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.344.3.11 mpxy_mbox_write_attributes()

```
static int mpxy_mbox_write_attributes (
            struct sbi_mpxy_channel * channel,
```

```
            u32 * outmem,
            u32 base_attr_id,
            u32 attr_count )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.344.3.12 mpxy_write_attr()

```
static void mpxy_write_attr (
            struct mpxy_rpmi_channel_attrs * attrs,
            u32 attr_id,
            u32 attr_val )  [static]
```

Here is the caller graph for this function:

30.344.4 Variable Documentation

30.344.4.1 clock_data

```
struct mpxy_mbox_data clock_data  [static]
```

Initial value:

```
= {
    .servicegrp_id = RPMI_SRVGRP_CLOCK,
    .num_services = RPMI_CLOCK_SRV_MAX_COUNT,
    .notifications_support = 1,
    .priv_data = clock_services,
}
```

30.344.4.2 clock_services

```
struct rpmi_service_data clock_services[]  [static]
```

30.344.4.3 fdt_mpxy_rpmi_mbox

```
struct fdt_driver fdt_mpxy_rpmi_mbox
```

Initial value:

```
= {
    .match_table = mpxy_mbox_match,
    .init = mpxy_mbox_init,
    .experimental = true,
}
```

30.344.4.4 mpxy_mbox_match

```
const struct fdt_match mpxy_mbox_match[] [static]
```

Initial value:

```
= {
    { .compatible = "riscv,rpmi-mpxy-clock", .data = &clock_data },
    { },
}
```

30.345 lib/utils/regmap/fdt_regmap.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/regmap/fdt_regmap.h>
```

Include dependency graph for fdt_regmap.c:

Functions

- static int fdt_regmap_find (const void ∗fdt, int nodeoff, struct regmap ∗∗out_rmap)
- int fdt_regmap_get_by_phandle (const void ∗fdt, u32 phandle, struct regmap ∗∗out_rmap)
- int fdt_regmap_get (const void ∗fdt, int nodeoff, struct regmap ∗∗out_rmap)

Variables

- const struct fdt_driver ∗const fdt_regmap_drivers [ ]

30.345.1 Function Documentation

30.345.1.1 fdt_regmap_find()

```
static int fdt_regmap_find (
            const void * fdt,
```

```
                int nodeoff,
                struct regmap ** out_rmap )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.345.1.2 fdt_regmap_get()

```
int fdt_regmap_get (
                const void * fdt,
                int nodeoff,
                struct regmap ** out_rmap )
```

Get regmap instance based on "regmap" property of the specified DT node Here is the call graph for this function:

Here is the caller graph for this function:

30.345.1.3 fdt_regmap_get_by_phandle()

```
int fdt_regmap_get_by_phandle (
            const void * fdt,
            u32 phandle,
            struct regmap ** out_rmap )
```

Get regmap instance based on phandle Here is the call graph for this function:

Here is the caller graph for this function:

30.345.2 Variable Documentation

30.345.2.1 fdt_regmap_drivers

```
const struct fdt_driver* const fdt_regmap_drivers[] [extern]
```

30.346 lib/utils/regmap/fdt_regmap_syscon.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_byteorder.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/regmap/fdt_regmap.h>
```

Include dependency graph for fdt_regmap_syscon.c:

Macros

- #define to_syscon_regmap(__rmap) container_of((__rmap), struct syscon_regmap, rmap)

Enumerations

- enum syscon_regmap_endian { SYSCON_ENDIAN_NATIVE = 0 , SYSCON_ENDIAN_LITTLE , SYSCON_ENDIAN_BIG , SYSCON_ENDIAN_MAX }

Functions

- static int regmap_syscon_read_8 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_8 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_16 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_16 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_32 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_32 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_le16 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_le16 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_le32 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_le32 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_be16 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_be16 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_be32 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_be32 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match regmap_syscon_match [ ]
- const struct fdt_driver fdt_regmap_syscon

30.346.1 Macro Definition Documentation

30.346.1.1 to_syscon_regmap

```
#define to_syscon_regmap(
            __rmap )  container_of((__rmap), struct syscon_regmap, rmap)
```

30.346.2 Enumeration Type Documentation

30.346.2.1 syscon_regmap_endian

```
enum syscon_regmap_endian
```

Enumerator

SYSCON_ENDIAN_NATIVE	
SYSCON_ENDIAN_LITTLE	
SYSCON_ENDIAN_BIG	
SYSCON_ENDIAN_MAX	

30.346.3 Function Documentation

30.346.3.1 regmap_syscon_init()

```
static int regmap_syscon_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.346.3.2 regmap_syscon_read_16()

```
static int regmap_syscon_read_16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

Here is the caller graph for this function:

30.346.3.3 regmap_syscon_read_32()

```
static int regmap_syscon_read_32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

Here is the caller graph for this function:

30.346.3.4 regmap_syscon_read_8()

```
static int regmap_syscon_read_8 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

Here is the caller graph for this function:

30.346.3.5 regmap_syscon_read_be16()

```
static int regmap_syscon_read_be16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

Here is the caller graph for this function:

30.346.3.6 regmap_syscon_read_be32()

```
static int regmap_syscon_read_be32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

Here is the caller graph for this function:

30.346.3.7 regmap_syscon_read_le16()

```
static int regmap_syscon_read_le16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

Here is the caller graph for this function:

30.346.3.8 regmap_syscon_read_le32()

```
static int regmap_syscon_read_le32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

Here is the caller graph for this function:

30.346.3.9 regmap_syscon_write_16()

```
static int regmap_syscon_write_16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )  [static]
```

Here is the caller graph for this function:

30.346.3.10 regmap_syscon_write_32()

```
static int regmap_syscon_write_32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )  [static]
```

Here is the caller graph for this function:

30.346.3.11 regmap_syscon_write_8()

```
static int regmap_syscon_write_8 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )  [static]
```

Here is the caller graph for this function:

30.346.3.12 regmap_syscon_write_be16()

```
static int regmap_syscon_write_be16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )  [static]
```

Here is the caller graph for this function:

30.346.3.13 regmap_syscon_write_be32()

```
static int regmap_syscon_write_be32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )  [static]
```

Here is the caller graph for this function:

30.346.3.14 regmap_syscon_write_le16()

```
static int regmap_syscon_write_le16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )  [static]
```

Here is the caller graph for this function:

30.346.3.15 regmap_syscon_write_le32()

```
static int regmap_syscon_write_le32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )  [static]
```

Here is the caller graph for this function:

30.346.4 Variable Documentation

30.346.4.1 fdt_regmap_syscon

```
const struct fdt_driver fdt_regmap_syscon
```

Initial value:

```
= {
    .match_table = regmap_syscon_match,
    .init = regmap_syscon_init,
}
```

30.346.4.2 regmap_syscon_match

```
const struct fdt_match regmap_syscon_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "syscon" },
    { },
}
```

30.347 lib/utils/regmap/regmap.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/regmap/regmap.h>
```

Include dependency graph for regmap.c:

Functions

- static SBI_LIST_HEAD (regmap_list)
- struct regmap ∗ regmap_find (unsigned int id)
- int regmap_add (struct regmap ∗rmap)
- void regmap_remove (struct regmap ∗rmap)
- static bool regmap_reg_valid (struct regmap ∗rmap, unsigned int reg)
- static unsigned int regmap_reg_addr (struct regmap ∗rmap, unsigned int reg)
- int regmap_read (struct regmap ∗rmap, unsigned int reg, unsigned int ∗val)
- int regmap_write (struct regmap ∗rmap, unsigned int reg, unsigned int val)
- int regmap_update_bits (struct regmap ∗rmap, unsigned int reg, unsigned int mask, unsigned int val)

30.347.1 Function Documentation

30.347.1.1 regmap_add()

```
int regmap_add (
            struct regmap * rmap )
```

Register a regmap instance Here is the call graph for this function:

Here is the caller graph for this function:

30.347.1.2 regmap_find()

```
struct regmap * regmap_find (
            unsigned int id )
```

Find a registered regmap instance Here is the call graph for this function:

Here is the caller graph for this function:

30.347.1.3 regmap_read()

```
int regmap_read (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )
```

Read a register in a regmap instance Here is the call graph for this function:

30.347.1.4 regmap_reg_addr()

```
static unsigned int regmap_reg_addr (
            struct regmap * rmap,
            unsigned int reg )  [static]
```

Here is the caller graph for this function:

30.347.1.5 regmap_reg_valid()

```
static bool regmap_reg_valid (
            struct regmap * rmap,
            unsigned int reg )  [static]
```

Here is the caller graph for this function:

30.347.1.6 regmap_remove()

```
void regmap_remove (
            struct regmap * rmap )
```

Un-register a regmap instance Here is the call graph for this function:

30.347.1.7 regmap_update_bits()

```
int regmap_update_bits (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int mask,
            unsigned int val )
```

Read-modify-write a register in a regmap instance Here is the call graph for this function:

Here is the caller graph for this function:

30.347.1.8 regmap_write()

```
int regmap_write (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )
```

Write a register in a regmap instance Here is the call graph for this function:

30.347.1.9 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            regmap_list  )  [static]
```

30.348 lib/utils/reset/fdt_reset.c File Reference

```
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset.c:

Functions

- void fdt_reset_init (const void *fdt)

Variables

- const struct fdt_driver *const fdt_reset_drivers [ ]

30.348.1 Function Documentation

30.348.1.1 fdt_reset_init()

```
void fdt_reset_init (
            const void * fdt )
```

Here is the call graph for this function:

30.348.2 Variable Documentation

30.348.2.1 fdt_reset_drivers

```
const struct fdt_driver* const fdt_reset_drivers[] [extern]
```

30.349 lib/utils/reset/fdt_reset_atcwdt200.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_system.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
#include <sbi_utils/sys/atcsmu.h>
```

Include dependency graph for fdt_reset_atcwdt200.c:

Macros

- #define ATCWDT200_WP_NUM 0x5aa5
- #define WREN_REG 0x18
- #define CTRL_REG 0x10
- #define RST_TIME_OFF 8
- #define RST_TIME_MSK (0x3 << RST_TIME_OFF)

- #define RST_CLK_128 (0 << RST_TIME_OFF)
- #define RST_CLK_256 (1 << RST_TIME_OFF)
- #define RST_CLK_512 (2 << RST_TIME_OFF)
- #define RST_CLK_1024 (3 << RST_TIME_OFF)
- #define INT_TIME_OFF 4
- #define INT_TIME_MSK (0xf << INT_TIME_OFF)
- #define INT_CLK_64 (0 << INT_TIME_OFF)
- #define INT_CLK_256 (1 << INT_TIME_OFF)
- #define INT_CLK_1024 (2 << INT_TIME_OFF)
- #define INT_CLK_2048 (3 << INT_TIME_OFF)
- #define INT_CLK_4096 (4 << INT_TIME_OFF)
- #define INT_CLK_8192 (5 << INT_TIME_OFF)
- #define INT_CLK_16384 (6 << INT_TIME_OFF)
- #define INT_CLK_32768 (7 << INT_TIME_OFF)
- #define RST_EN (1 << 3)
- #define INT_EN (1 << 2)
- #define CLK_PCLK (1 << 1)
- #define WDT_EN (1 << 0)

Functions

- static int ae350_system_reset_check (u32 type, u32 reason)
- static void ae350_system_reset (u32 type, u32 reason)
- static int atcwdt200_reset_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static volatile char * wdt_addr = NULL
- static struct smu_data smu = { 0 }
- static struct sbi_system_reset_device atcwdt200_reset
- static const struct fdt_match atcwdt200_reset_match [ ]
- const struct fdt_driver fdt_reset_atcwdt200

30.349.1 Macro Definition Documentation

30.349.1.1 ATCWDT200_WP_NUM

```
#define ATCWDT200_WP_NUM 0x5aa5
```

30.349.1.2 CLK_PCLK

```
#define CLK_PCLK (1 << 1)
```

30.349.1.3 CTRL_REG

```
#define CTRL_REG 0x10
```

30.349.1.4 INT_CLK_1024

```
#define INT_CLK_1024 (2 << INT_TIME_OFF)
```

30.349.1.5 INT_CLK_16384

```
#define INT_CLK_16384 (6 << INT_TIME_OFF)
```

30.349.1.6 INT_CLK_2048

```
#define INT_CLK_2048 (3 << INT_TIME_OFF)
```

30.349.1.7 INT_CLK_256

```
#define INT_CLK_256 (1 << INT_TIME_OFF)
```

30.349.1.8 INT_CLK_32768

```
#define INT_CLK_32768 (7 << INT_TIME_OFF)
```

30.349.1.9 INT_CLK_4096

```
#define INT_CLK_4096 (4 << INT_TIME_OFF)
```

30.349.1.10 INT_CLK_64

```
#define INT_CLK_64 (0 << INT_TIME_OFF)
```

30.349.1.11 INT_CLK_8192

```
#define INT_CLK_8192 (5 << INT_TIME_OFF)
```

30.349.1.12 INT_EN

```
#define INT_EN (1 << 2)
```

30.349.1.13 INT_TIME_MSK

```
#define INT_TIME_MSK (0xf << INT_TIME_OFF)
```

30.349.1.14 INT_TIME_OFF

```
#define INT_TIME_OFF 4
```

30.349.1.15 RST_CLK_1024

```
#define RST_CLK_1024 (3 << RST_TIME_OFF)
```

30.349.1.16 RST_CLK_128

```
#define RST_CLK_128 (0 << RST_TIME_OFF)
```

30.349.1.17 RST_CLK_256

```
#define RST_CLK_256 (1 << RST_TIME_OFF)
```

30.349.1.18 RST_CLK_512

```
#define RST_CLK_512 (2 << RST_TIME_OFF)
```

30.349.1.19 RST_EN

```
#define RST_EN (1 << 3)
```

30.349.1.20 RST_TIME_MSK

```
#define RST_TIME_MSK (0x3 << RST_TIME_OFF)
```

30.349.1.21 RST_TIME_OFF

```
#define RST_TIME_OFF 8
```

30.349.1.22 WDT_EN

```
#define WDT_EN (1 << 0)
```

30.349.1.23 WREN_REG

```
#define WREN_REG 0x18
```

30.349.2 Function Documentation

30.349.2.1 ae350_system_reset()

```
static void ae350_system_reset (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.349.2.2 ae350_system_reset_check()

```
static int ae350_system_reset_check (
            u32 type,
            u32 reason )  [static]
```

30.349.2.3 atcwdt200_reset_init()

```
static int atcwdt200_reset_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.349.3 Variable Documentation

30.349.3.1 atcwdt200_reset

```
struct sbi_system_reset_device atcwdt200_reset  [static]
```

Initial value:

```
= {
    .name           = "atcwdt200",
    .system_reset_check = ae350_system_reset_check,
    .system_reset       = ae350_system_reset,
}
```

30.349.3.2 atcwdt200_reset_match

```
const struct fdt_match atcwdt200_reset_match[] [static]
```

Initial value:

```
= {
    { .compatible = "andestech,atcwdt200" },
    {},
}
```

30.349.3.3 fdt_reset_atcwdt200

```
const struct fdt_driver fdt_reset_atcwdt200
```

Initial value:

```
= {
    .match_table = atcwdt200_reset_match,
    .init        = atcwdt200_reset_init,
}
```

30.349.3.4 smu

```
struct smu_data smu = { 0 } [static]
```

30.349.3.5 wdt_addr

```
volatile char* wdt_addr = NULL [static]
```

30.350 lib/utils/reset/fdt_reset_gpio.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset_gpio.c:

Functions

- static struct gpio_reset * gpio_reset_get (bool is_restart, u32 type)
- static void gpio_reset_exec (struct gpio_reset *reset)
- static int gpio_system_poweroff_check (u32 type, u32 reason)
- static void gpio_system_poweroff (u32 type, u32 reason)

- static int gpio_system_restart_check (u32 type, u32 reason)
- static void gpio_system_restart (u32 type, u32 reason)
- static int gpio_reset_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static struct gpio_reset poweroff
- static struct gpio_reset restart
- static struct sbi_system_reset_device gpio_poweroff
- static struct sbi_system_reset_device gpio_restart
- static const struct fdt_match gpio_poweroff_match [ ]
- const struct fdt_driver fdt_poweroff_gpio
- static const struct fdt_match gpio_reset_match [ ]
- const struct fdt_driver fdt_reset_gpio

30.350.1 Function Documentation

30.350.1.1 gpio_reset_exec()

```
static void gpio_reset_exec (
            struct gpio_reset * reset )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.350.1.2 gpio_reset_get()

```
static struct gpio_reset * gpio_reset_get (
            bool is_restart,
            u32 type )  [static]
```

Here is the caller graph for this function:

30.350.1.3 gpio_reset_init()

```
static int gpio_reset_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.350.1.4 gpio_system_poweroff()

```
static void gpio_system_poweroff (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.350.1.5 gpio_system_poweroff_check()

```
static int gpio_system_poweroff_check (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.350.1.6 gpio_system_restart()

```
static void gpio_system_restart (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.350.1.7 gpio_system_restart_check()

```
static int gpio_system_restart_check (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.350.2 Variable Documentation

30.350.2.1 fdt_poweroff_gpio

```
const struct fdt_driver fdt_poweroff_gpio
```

Initial value:

```
= {
    .match_table = gpio_poweroff_match,
```

```
    .init = gpio_reset_init,
}
```

30.350.2.2 fdt_reset_gpio

```
const struct fdt_driver fdt_reset_gpio
```

Initial value:

```
= {
    .match_table = gpio_reset_match,
    .init = gpio_reset_init,
}
```

30.350.2.3 gpio_poweroff

```
struct sbi_system_reset_device gpio_poweroff  [static]
```

Initial value:

```
= {
    .name = "gpio-poweroff",
    .system_reset_check = gpio_system_poweroff_check,
    .system_reset = gpio_system_poweroff
}
```

30.350.2.4 gpio_poweroff_match

```
const struct fdt_match gpio_poweroff_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "gpio-poweroff", .data = (const void *)false },
    { },
}
```

30.350.2.5 gpio_reset_match

```
const struct fdt_match gpio_reset_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "gpio-restart", .data = (const void *)true },
    { },
}
```

30.350.2.6 gpio_restart

```
struct sbi_system_reset_device gpio_restart  [static]
```

Initial value:

```
= {
    .name = "gpio-restart",
    .system_reset_check = gpio_system_restart_check,
    .system_reset = gpio_system_restart
}
```

30.350.2.7 poweroff

```
struct gpio_reset poweroff  [static]
```

Initial value:

```
= {
    .active_delay = 100,
    .inactive_delay = 100
}
```

30.350.2.8 restart

```
struct gpio_reset restart  [static]
```

Initial value:

```
= {
    .active_delay = 100,
    .inactive_delay = 100
}
```

30.351 lib/utils/reset/fdt_reset_htif.c File Reference

```
#include <sbi_utils/reset/fdt_reset.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/sys/htif.h>
```

Include dependency graph for fdt_reset_htif.c:

Functions

- static int htif_reset_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match htif_reset_match [ ]
- const struct fdt_driver fdt_reset_htif

30.351.1 Function Documentation

30.351.1.1 htif_reset_init()

```
static int htif_reset_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.351.2 Variable Documentation

30.351.2.1 fdt_reset_htif

```
const struct fdt_driver fdt_reset_htif
```

Initial value:

```
= {
    .match_table = htif_reset_match,
    .init = htif_reset_init
}
```

30.351.2.2 htif_reset_match

```
const struct fdt_match htif_reset_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "ucb,htif0" },
    { },
}
```

30.352 lib/utils/reset/fdt_reset_rpmi.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
#include <sbi_utils/mailbox/fdt_mailbox.h>
#include <sbi_utils/mailbox/rpmi_msgprot.h>
#include <sbi_utils/mailbox/rpmi_mailbox.h>
```

Include dependency graph for fdt_reset_rpmi.c:

Functions

- static int rpmi_system_reset_type_check (u32 reset_type)
- static void rpmi_do_system_reset (u32 reset_type)
- static int rpmi_system_reset_check (u32 type, u32 reason)
- static void rpmi_system_reset (u32 type, u32 reason)
- static int rpmi_reset_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static struct rpmi_sysreset sysreset_ctx
- static struct sbi_system_reset_device rpmi_reset_dev
- static const struct fdt_match rpmi_reset_match [ ]
- struct fdt_driver fdt_reset_rpmi

30.352.1 Function Documentation

30.352.1.1 rpmi_do_system_reset()

```
static void rpmi_do_system_reset (
            u32 reset_type )  [static]
```

rpmi_do_system_reset: Do system reset
@reset_type: RPMI System Reset Type Here is the call graph for this function:

Here is the caller graph for this function:

30.352.1.2 rpmi_reset_init()

```
static int rpmi_reset_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.352.1.3 rpmi_system_reset()

```
static void rpmi_system_reset (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.352.1.4 rpmi_system_reset_check()

```
static int rpmi_system_reset_check (
            u32 type,
            u32 reason )  [static]
```

rpmi_system_reset_check: Check the support for various reset types
@type: SBI System Reset Type @reason: Reason for system reset

30.352.1.5 rpmi_system_reset_type_check()

```
static int rpmi_system_reset_type_check (
            u32 reset_type )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.352.2 Variable Documentation

30.352.2.1 fdt_reset_rpmi

```
struct fdt_driver fdt_reset_rpmi
```

Initial value:

```
= {
    .match_table = rpmi_reset_match,
    .init = rpmi_reset_init,
    .experimental = true,
}
```

30.352.2.2 rpmi_reset_dev

```
struct sbi_system_reset_device rpmi_reset_dev  [static]
```

Initial value:

```
= {
    .name = "rpmi-system-reset",
    .system_reset_check = rpmi_system_reset_check,
    .system_reset = rpmi_system_reset,
}
```

30.352.2.3 rpmi_reset_match

```
const struct fdt_match rpmi_reset_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,rpmi-system-reset" },
    {},
}
```

30.352.2.4 sysreset_ctx

```
struct rpmi_sysreset sysreset_ctx  [static]
```

30.353 lib/utils/reset/fdt_reset_sg2042_hwmon_mcu.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
#include <sbi_utils/i2c/fdt_i2c.h>
```

Include dependency graph for fdt_reset_sg2042_hwmon_mcu.c:

Macros

- #define MANGO_BOARD_TYPE_MASK 0x80
- #define REG_BOARD_TYPE 0x00
- #define REG_CMD 0x03
- #define CMD_POWEROFF 0x02
- #define CMD_RESET 0x03
- #define CMD_REBOOT 0x07

Functions

- static int sg2042_mcu_reset_check (u32 type, u32 reason)
- static void sg2042_mcu_reset (u32 type, u32 reason)
- static int sg2042_mcu_reset_check_board (struct i2c_adapter ∗adap, uint32_t reg)
- static int sg2042_mcu_reset_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static struct i2c_adapter ∗ mcu_adapter = NULL
- static uint32_t mcu_reg = 0
- static struct sbi_system_reset_device sg2042_mcu_reset_device
- static const struct fdt_match sg2042_mcu_reset_match [ ]
- const struct fdt_driver fdt_reset_sg2042_mcu

30.353.1 Macro Definition Documentation

30.353.1.1 CMD_POWEROFF

```
#define CMD_POWEROFF 0x02
```

30.353.1.2 CMD_REBOOT

```
#define CMD_REBOOT 0x07
```

30.353.1.3 CMD_RESET

```
#define CMD_RESET 0x03
```

30.353.1.4 MANGO_BOARD_TYPE_MASK

```
#define MANGO_BOARD_TYPE_MASK 0x80
```

30.353.1.5 REG_BOARD_TYPE

```
#define REG_BOARD_TYPE 0x00
```

30.353.1.6 REG_CMD

```
#define REG_CMD 0x03
```

30.353.2 Function Documentation

30.353.2.1 sg2042_mcu_reset()

```
static void sg2042_mcu_reset (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.353.2.2 sg2042_mcu_reset_check()

```
static int sg2042_mcu_reset_check (
            u32 type,
            u32 reason )  [static]
```

30.353.2.3 sg2042_mcu_reset_check_board()

```
static int sg2042_mcu_reset_check_board (
            struct i2c_adapter * adap,
            uint32_t reg )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.353.2.4 sg2042_mcu_reset_init()

```
static int sg2042_mcu_reset_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.353.3 Variable Documentation

30.353.3.1 fdt_reset_sg2042_mcu

```
const struct fdt_driver fdt_reset_sg2042_mcu
```

Initial value:

```
= {
    .match_table = sg2042_mcu_reset_match,
    .init = sg2042_mcu_reset_init,
}
```

30.353.3.2 mcu_adapter

```
struct i2c_adapter* mcu_adapter = NULL  [static]
```

30.353.3.3 mcu_reg

```
uint32_t mcu_reg = 0  [static]
```

30.353.3.4 sg2042_mcu_reset_device

```
struct sbi_system_reset_device sg2042_mcu_reset_device  [static]
```

Initial value:

```
= {
    .name = "sg2042-mcu-reset",
    .system_reset_check = sg2042_mcu_reset_check,
    .system_reset = sg2042_mcu_reset
}
```

30.353.3.5 sg2042_mcu_reset_match

```
const struct fdt_match sg2042_mcu_reset_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "sophgo,sg2042-hwmon-mcu", .data = (void *)true},
    { },
}
```

30.354 lib/utils/reset/fdt_reset_sunxi_wdt.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_error.h>
```

```
#include <sbi/sbi_system.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset_sunxi_wdt.c:

Macros

- #define WDT_KEY_VAL 0x16aa0000
- #define WDT_SOFT_RST_REG 0x08
- #define WDT_SOFT_RST_EN BIT(0)
- #define WDT_MODE_REG 0x18

Functions

- static int sunxi_wdt_system_reset_check (u32 type, u32 reason)
- static void sunxi_wdt_system_reset (u32 type, u32 reason)
- static int sunxi_wdt_reset_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static volatile char * sunxi_wdt_base
- static struct sbi_system_reset_device sunxi_wdt_reset
- static const struct fdt_match sunxi_wdt_reset_match [ ]
- const struct fdt_driver fdt_reset_sunxi_wdt

30.354.1 Macro Definition Documentation

30.354.1.1 WDT_KEY_VAL

```
#define WDT_KEY_VAL 0x16aa0000
```

30.354.1.2 WDT_MODE_REG

```
#define WDT_MODE_REG 0x18
```

30.354.1.3 WDT_SOFT_RST_EN

```
#define WDT_SOFT_RST_EN BIT(0)
```

30.354.1.4 WDT_SOFT_RST_REG

```
#define WDT_SOFT_RST_REG 0x08
```

30.354.2 Function Documentation

30.354.2.1 sunxi_wdt_reset_init()

```
static int sunxi_wdt_reset_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.354.2.2 sunxi_wdt_system_reset()

```
static void sunxi_wdt_system_reset (
            u32 type,
            u32 reason )  [static]
```

30.354.2.3 sunxi_wdt_system_reset_check()

```
static int sunxi_wdt_system_reset_check (
            u32 type,
            u32 reason )  [static]
```

30.354.3 Variable Documentation

30.354.3.1 fdt_reset_sunxi_wdt

```
const struct fdt_driver fdt_reset_sunxi_wdt
```

Initial value:

```
= {
    .match_table = sunxi_wdt_reset_match,
    .init = sunxi_wdt_reset_init,
}
```

30.354.3.2 sunxi_wdt_base

```
volatile char* sunxi_wdt_base  [static]
```

30.354.3.3 sunxi_wdt_reset

```
struct sbi_system_reset_device sunxi_wdt_reset  [static]
```

Initial value:

```
= {
    .name = "sunxi-wdt-reset",
    .system_reset_check = sunxi_wdt_system_reset_check,
    .system_reset = sunxi_wdt_system_reset,
}
```

30.354.3.4 sunxi_wdt_reset_match

```
const struct fdt_match sunxi_wdt_reset_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "allwinner,sun20i-d1-wdt-reset" },
    { },
}
```

30.355 lib/utils/reset/fdt_reset_syscon.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_system.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/regmap/fdt_regmap.h>
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset_syscon.c:

Functions

- static struct syscon_reset ∗ syscon_reset_get (bool is_reboot, u32 type)
- static void syscon_reset_exec (struct syscon_reset ∗reset)
- static int syscon_poweroff_check (u32 type, u32 reason)
- static void syscon_do_poweroff (u32 type, u32 reason)
- static int syscon_reboot_check (u32 type, u32 reason)
- static void syscon_do_reboot (u32 type, u32 reason)
- static int syscon_reset_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static struct syscon_reset poweroff
- static struct syscon_reset reboot
- static struct sbi_system_reset_device syscon_poweroff
- static struct sbi_system_reset_device syscon_reboot
- static const struct fdt_match syscon_poweroff_match [ ]
- const struct fdt_driver fdt_syscon_poweroff
- static const struct fdt_match syscon_reboot_match [ ]
- const struct fdt_driver fdt_syscon_reboot

30.355.1 Function Documentation

30.355.1.1 syscon_do_poweroff()

```
static void syscon_do_poweroff (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.355.1.2 syscon_do_reboot()

```
static void syscon_do_reboot (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.355.1.3 syscon_poweroff_check()

```
static int syscon_poweroff_check (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.355.1.4 syscon_reboot_check()

```
static int syscon_reboot_check (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.355.1.5 syscon_reset_exec()

```
static void syscon_reset_exec (
            struct syscon_reset * reset )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.355.1.6 syscon_reset_get()

```
static struct syscon_reset * syscon_reset_get (
            bool is_reboot,
            u32 type )  [static]
```

Here is the caller graph for this function:

30.355.1.7 syscon_reset_init()

```
static int syscon_reset_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.355.2 Variable Documentation

30.355.2.1 fdt_syscon_poweroff

```
const struct fdt_driver fdt_syscon_poweroff
```

Initial value:

```
= {
    .match_table = syscon_poweroff_match,
    .init = syscon_reset_init,
}
```

30.355.2.2 fdt_syscon_reboot

```
const struct fdt_driver fdt_syscon_reboot
```

Initial value:

```
= {
    .match_table = syscon_reboot_match,
    .init = syscon_reset_init,
}
```

30.355.2.3 poweroff

```
struct syscon_reset poweroff [static]
```

30.355.2.4 reboot

```
struct syscon_reset reboot [static]
```

30.355.2.5 syscon_poweroff

```
struct sbi_system_reset_device syscon_poweroff [static]
```

Initial value:

```
= {
    .name = "syscon-poweroff",
    .system_reset_check = syscon_poweroff_check,
    .system_reset = syscon_do_poweroff
}
```

30.355.2.6 syscon_poweroff_match

```
const struct fdt_match syscon_poweroff_match[] [static]
```

Initial value:

```
= {
    { .compatible = "syscon-poweroff", .data = (const void *)false },
    { },
}
```

30.355.2.7 syscon_reboot

```
struct sbi_system_reset_device syscon_reboot [static]
```

Initial value:

```
= {
    .name = "syscon-reboot",
    .system_reset_check = syscon_reboot_check,
    .system_reset = syscon_do_reboot
}
```

30.355.2.8 syscon_reboot_match

```
const struct fdt_match syscon_reboot_match[] [static]
```

Initial value:

```
= {
    { .compatible = "syscon-reboot", .data = (const void *)true },
    { },
}
```

30.356 lib/utils/serial/cadence-uart.c File Reference

```
#include <sbi/sbi_domain.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/cadence-uart.h>
```

Include dependency graph for cadence-uart.c:

Macros

- #define UART_REG_CTRL 0x00
- #define UART_REG_MODE 0x04
- #define UART_REG_IDR 0x0C
- #define UART_REG_BRGR 0x18
- #define UART_REG_CSR 0x2C
- #define UART_REG_RFIFO_TFIFO 0x30
- #define UART_REG_BDIVR 0x34
- #define UART_CTRL_RXRES 0x00000001
- #define UART_CTRL_TXRES 0x00000002
- #define UART_CTRL_RXEN 0x00000004
- #define UART_CTRL_RXDIS 0x00000008
- #define UART_CTRL_TXEN 0x00000010
- #define UART_CTRL_TXDIS 0x00000020
- #define UART_MODE_PAR_NONE 0x00000020 /∗ No parity set ∗/
- #define UART_BRGR_CD_CLKDIVISOR 0x00000001 /∗ baud_sample = sel_clk ∗/
- #define UART_CSR_REMPTY 0x00000002
- #define UART_CSR_TFUL 0x00000010

Functions

- static unsigned int uart_min_clk_divisor (uint64_t in_freq, uint64_t max_target_hz)
- static u32 get_reg (u32 offset)
- static void set_reg (u32 offset, u32 val)
- static void cadence_uart_putc (char ch)
- static int cadence_uart_getc (void)
- int cadence_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile void ∗ uart_base
- static u32 uart_in_freq
- static u32 uart_baudrate
- static struct sbi_console_device cadence_console

30.356.1 Macro Definition Documentation

30.356.1.1 UART_BRGR_CD_CLKDIVISOR

```
#define UART_BRGR_CD_CLKDIVISOR 0x00000001 /∗ baud_sample = sel_clk ∗/
```

30.356.1.2 UART_CSR_REMPTY

```
#define UART_CSR_REMPTY 0x00000002
```

30.356.1.3 UART_CSR_TFUL

```
#define UART_CSR_TFUL 0x00000010
```

30.356.1.4 UART_CTRL_RXDIS

```
#define UART_CTRL_RXDIS 0x00000008
```

30.356.1.5 UART_CTRL_RXEN

```
#define UART_CTRL_RXEN 0x00000004
```

30.356.1.6 UART_CTRL_RXRES

```
#define UART_CTRL_RXRES 0x00000001
```

30.356.1.7 UART_CTRL_TXDIS

```
#define UART_CTRL_TXDIS 0x00000020
```

30.356.1.8 UART_CTRL_TXEN

```
#define UART_CTRL_TXEN 0x00000010
```

30.356.1.9 UART_CTRL_TXRES

```
#define UART_CTRL_TXRES 0x00000002
```

30.356.1.10 UART_MODE_PAR_NONE

```
#define UART_MODE_PAR_NONE 0x00000020 /* No parity set */
```

30.356.1.11 UART_REG_BDIVR

```
#define UART_REG_BDIVR 0x34
```

30.356.1.12 UART_REG_BRGR

```
#define UART_REG_BRGR 0x18
```

30.356.1.13 UART_REG_CSR

```
#define UART_REG_CSR 0x2C
```

30.356.1.14 UART_REG_CTRL

```
#define UART_REG_CTRL 0x00
```

30.356.1.15 UART_REG_IDR

```
#define UART_REG_IDR 0x0C
```

30.356.1.16 UART_REG_MODE

```
#define UART_REG_MODE 0x04
```

30.356.1.17 UART_REG_RFIFO_TFIFO

```
#define UART_REG_RFIFO_TFIFO 0x30
```

30.356.2 Function Documentation

30.356.2.1 cadence_uart_getc()

```
static int cadence_uart_getc (
            void ) [static]
```

Here is the call graph for this function:

30.356.2.2 cadence_uart_init()

```
int cadence_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.356.2.3 cadence_uart_putc()

```
static void cadence_uart_putc (
            char ch ) [static]
```

Here is the call graph for this function:

30.356.2.4 get_reg()

```
static u32 get_reg (
            u32 offset )  [static]
```

Here is the caller graph for this function:

30.356.2.5 set_reg()

```
static void set_reg (
            u32 offset,
            u32 val )  [static]
```

Here is the caller graph for this function:

30.356.2.6 uart_min_clk_divisor()

```
static unsigned int uart_min_clk_divisor (
```

```
uint64_t in_freq,
uint64_t max_target_hz )  [inline], [static]
```

Here is the caller graph for this function:

30.356.3 Variable Documentation

30.356.3.1 cadence_console

```
struct sbi_console_device cadence_console  [static]
```

Initial value:

```
= {
    .name = "cadence_uart",
    .console_putc = cadence_uart_putc,
    .console_getc = cadence_uart_getc
}
```

30.356.3.2 uart_base

```
volatile void* uart_base  [static]
```

30.356.3.3 uart_baudrate

```
u32 uart_baudrate  [static]
```

30.356.3.4 uart_in_freq

```
u32 uart_in_freq  [static]
```

30.357 lib/utils/serial/fdt_serial.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
```

Include dependency graph for fdt_serial.c:

Functions

- int fdt_serial_init (const void *fdt)

Variables

- const struct fdt_driver *const fdt_serial_drivers [ ]

30.357.1 Function Documentation

30.357.1.1 fdt_serial_init()

```
int fdt_serial_init (
            const void * fdt )
```

Here is the call graph for this function:

30.357.2 Variable Documentation

30.357.2.1 fdt_serial_drivers

```
const struct fdt_driver* const fdt_serial_drivers[] [extern]
```

30.358 lib/utils/serial/fdt_serial_cadence.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
```

```
#include <sbi_utils/serial/cadence-uart.h>
```

Include dependency graph for fdt_serial_cadence.c:

Functions

- static int serial_cadence_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_cadence_match [ ]
- const struct fdt_driver fdt_serial_cadence

30.358.1 Function Documentation

30.358.1.1 serial_cadence_init()

```
static int serial_cadence_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.358.2 Variable Documentation

30.358.2.1 fdt_serial_cadence

```
const struct fdt_driver fdt_serial_cadence
```

Initial value:

```
= {
    .match_table = serial_cadence_match,
    .init = serial_cadence_init
}
```

30.358.2.2 serial_cadence_match

```
const struct fdt_match serial_cadence_match[] [static]
```

Initial value:

```
= {
    { .compatible = "cdns,uart-r1p8", },
    { .compatible = "cdns,uart-r1p12" },
    { .compatible = "starfive,jh8100-uart" },
    { },
}
```

30.359 lib/utils/serial/fdt_serial_gaisler.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/gaisler-uart.h>
```

Include dependency graph for fdt_serial_gaisler.c:

Functions

- static int serial_gaisler_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_gaisler_match [ ]
- const struct fdt_driver fdt_serial_gaisler

30.359.1 Function Documentation

30.359.1.1 serial_gaisler_init()

```
static int serial_gaisler_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.359.2 Variable Documentation

30.359.2.1 fdt_serial_gaisler

```
const struct fdt_driver fdt_serial_gaisler
```

Initial value:

```
= {
    .match_table = serial_gaisler_match,
    .init = serial_gaisler_init
}
```

30.359.2.2 serial_gaisler_match

```
const struct fdt_match serial_gaisler_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "gaisler,apbuart" },
    {},
}
```

30.360 lib/utils/serial/fdt_serial_htif.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_domain.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/sys/htif.h>
```

Include dependency graph for fdt_serial_htif.c:

Functions

- static int serial_htif_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match serial_htif_match [ ]
- const struct fdt_driver fdt_serial_htif

30.360.1 Function Documentation

30.360.1.1 serial_htif_init()

```
static int serial_htif_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.360.2 Variable Documentation

30.360.2.1 fdt_serial_htif

```
const struct fdt_driver fdt_serial_htif
```

Initial value:

```
= {
    .match_table = serial_htif_match,
    .init = serial_htif_init
}
```

30.360.2.2 serial_htif_match

```
const struct fdt_match serial_htif_match[] [static]
```

Initial value:

```
= {
    { .compatible = "ucb,htif0" },
    { },
}
```

30.361 lib/utils/serial/fdt_serial_litex.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/litex-uart.h>
```

Include dependency graph for fdt_serial_litex.c:

Functions

- static int serial_litex_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_litex_match [ ]
- const struct fdt_driver fdt_serial_litex

30.361.1 Function Documentation

30.361.1.1 serial_litex_init()

```
static int serial_litex_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.361.2 Variable Documentation

30.361.2.1 fdt_serial_litex

```
const struct fdt_driver fdt_serial_litex
```

Initial value:

```
= {
    .match_table = serial_litex_match,
    .init = serial_litex_init
}
```

30.361.2.2 serial_litex_match

```
const struct fdt_match serial_litex_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "litex,liteuart" },
    { },
}
```

30.362 lib/utils/serial/fdt_serial_renesas_scif.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/renesas-scif.h>
```

Include dependency graph for fdt_serial_renesas_scif.c:

Functions

- static int serial_renesas_scif_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match serial_renesas_scif_match [ ]
- const struct fdt_driver fdt_serial_renesas_scif

30.362.1 Function Documentation

30.362.1.1 serial_renesas_scif_init()

```
static int serial_renesas_scif_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.362.2 Variable Documentation

30.362.2.1 fdt_serial_renesas_scif

```
const struct fdt_driver fdt_serial_renesas_scif
```

Initial value:

```
= {
    .match_table = serial_renesas_scif_match,
    .init = serial_renesas_scif_init
}
```

30.362.2.2 serial_renesas_scif_match

```
const struct fdt_match serial_renesas_scif_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "renesas,scif-r9a07g043" },
    {  }
}
```

30.363 lib/utils/serial/fdt_serial_shakti.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/shakti-uart.h>
```

Include dependency graph for fdt_serial_shakti.c:

Functions

- static int serial_shakti_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_shakti_match [ ]
- const struct fdt_driver fdt_serial_shakti

30.363.1 Function Documentation

30.363.1.1 serial_shakti_init()

```
static int serial_shakti_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.363.2 Variable Documentation

30.363.2.1 fdt_serial_shakti

```
const struct fdt_driver fdt_serial_shakti
```

Initial value:

```
= {
    .match_table = serial_shakti_match,
    .init = serial_shakti_init
}
```

30.363.2.2 serial_shakti_match

```
const struct fdt_match serial_shakti_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "shakti,uart0" },
    { },
}
```

30.364 lib/utils/serial/fdt_serial_sifive.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/sifive-uart.h>
```

Include dependency graph for fdt_serial_sifive.c:

Functions

- static int serial_sifive_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match serial_sifive_match [ ]
- const struct fdt_driver fdt_serial_sifive

30.364.1 Function Documentation

30.364.1.1 serial_sifive_init()

```
static int serial_sifive_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.364.2 Variable Documentation

30.364.2.1 fdt_serial_sifive

```
const struct fdt_driver fdt_serial_sifive
```

Initial value:

```
= {
    .match_table = serial_sifive_match,
    .init = serial_sifive_init
}
```

30.364.2.2 serial_sifive_match

```
const struct fdt_match serial_sifive_match[] [static]
```

Initial value:

```
= {
    { .compatible = "sifive,fu540-c000-uart" },
    { .compatible = "sifive,uart0" },
    { },
}
```

30.365 lib/utils/serial/fdt_serial_uart8250.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/uart8250.h>
```

Include dependency graph for fdt_serial_uart8250.c:

Functions

- static int serial_uart8250_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_uart8250_match [ ]
- const struct fdt_driver fdt_serial_uart8250

30.365.1 Function Documentation

30.365.1.1 serial_uart8250_init()

```
static int serial_uart8250_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.365.2 Variable Documentation

30.365.2.1 fdt_serial_uart8250

```
const struct fdt_driver fdt_serial_uart8250
```

Initial value:

```
= {
    .match_table = serial_uart8250_match,
    .init = serial_uart8250_init,
}
```

30.365.2.2 serial_uart8250_match

```
const struct fdt_match serial_uart8250_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "ns16550" },
    { .compatible = "ns16550a" },
    { .compatible = "snps,dw-apb-uart" },
    { },
}
```

30.366 lib/utils/serial/fdt_serial_xlnx_uartlite.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/xlnx_uartlite.h>
```

Include dependency graph for fdt_serial_xlnx_uartlite.c:

Functions

- static int serial_xlnx_uartlite_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_xlnx_uartlite_match [ ]
- const struct fdt_driver fdt_serial_xlnx_uartlite

30.366.1 Function Documentation

30.366.1.1 serial_xlnx_uartlite_init()

```
static int serial_xlnx_uartlite_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.366.2 Variable Documentation

30.366.2.1 fdt_serial_xlnx_uartlite

```
const struct fdt_driver fdt_serial_xlnx_uartlite
```

Initial value:

```
= {
    .match_table = serial_xlnx_uartlite_match,
    .init = serial_xlnx_uartlite_init,
}
```

30.366.2.2 serial_xlnx_uartlite_match

```
const struct fdt_match serial_xlnx_uartlite_match[] [static]
```

Initial value:

```
= {
    { .compatible = "xlnx,xps-uartlite-1.00.a" },
    { },
}
```

30.367 lib/utils/serial/gaisler-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/gaisler-uart.h>
```

Include dependency graph for gaisler-uart.c:

Macros

- #define UART_REG_DATA 0
- #define UART_REG_STATUS 1
- #define UART_REG_CTRL 2
- #define UART_REG_SCALER 3
- #define UART_DATA_DATA 0x000000ff
- #define UART_STATUS_FIFOFULL 0x00000200
- #define UART_STATUS_DATAREADY 0x00000001
- #define UART_CTRL_DB (1<<11)
- #define UART_CTRL_FL (1<<6)
- #define UART_CTRL_TE (1<<1)
- #define UART_CTRL_RE (1<<0)

Functions

- static u32 get_reg (u32 num)
- static void set_reg (u32 num, u32 val)
- static void gaisler_uart_putc (char ch)
- static int gaisler_uart_getc (void)
- int gaisler_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile char * uart_base
- static struct sbi_console_device gaisler_console

30.367.1 Macro Definition Documentation

30.367.1.1 UART_CTRL_DB

```
#define UART_CTRL_DB (1<<11)
```

30.367.1.2 UART_CTRL_FL

```
#define UART_CTRL_FL (1<<6)
```

30.367.1.3 UART_CTRL_RE

```
#define UART_CTRL_RE (1<<0)
```

30.367.1.4 UART_CTRL_TE

```
#define UART_CTRL_TE (1<<1)
```

30.367.1.5 UART_DATA_DATA

```
#define UART_DATA_DATA 0x000000ff
```

30.367.1.6 UART_REG_CTRL

```
#define UART_REG_CTRL 2
```

30.367.1.7 UART_REG_DATA

```
#define UART_REG_DATA 0
```

30.367.1.8 UART_REG_SCALER

```
#define UART_REG_SCALER 3
```

30.367.1.9 UART_REG_STATUS

```
#define UART_REG_STATUS 1
```

30.367.1.10 UART_STATUS_DATAREADY

```
#define UART_STATUS_DATAREADY 0x00000001
```

30.367.1.11 UART_STATUS_FIFOFULL

```
#define UART_STATUS_FIFOFULL 0x00000200
```

30.367.2 Function Documentation

30.367.2.1 gaisler_uart_getc()

```
static int gaisler_uart_getc (
            void ) [static]
```

Here is the call graph for this function:

30.367.2.2 gaisler_uart_init()

```
int gaisler_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.367.2.3 gaisler_uart_putc()

```
static void gaisler_uart_putc (
            char ch ) [static]
```

Here is the call graph for this function:

30.367.2.4 get_reg()

```
static u32 get_reg (
            u32 num )  [static]
```

Here is the caller graph for this function:

30.367.2.5 set_reg()

```
static void set_reg (
            u32 num,
            u32 val )  [static]
```

Here is the caller graph for this function:

30.367.3 Variable Documentation

30.367.3.1 gaisler_console

```
struct sbi_console_device gaisler_console  [static]
```

Initial value:

```
= {
    .name         = "gaisler_uart",
    .console_putc = gaisler_uart_putc,
    .console_getc = gaisler_uart_getc
}
```

30.367.3.2 uart_base

```
volatile char* uart_base  [static]
```

30.368 lib/utils/serial/litex-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/litex-uart.h>
```

Include dependency graph for litex-uart.c:

Macros

- #define UART_REG_RXTX 0
- #define UART_REG_TXFULL 1
- #define UART_REG_RXEMPTY 2
- #define UART_REG_EV_STATUS 3
- #define UART_REG_EV_PENDING 4
- #define UART_REG_EV_ENABLE 5
- #define UART_EV_RX 0x2

Functions

- static u8 get_reg (u8 reg)
- static void set_reg (u8 reg, u8 val)
- static void litex_uart_putc (char ch)
- static int litex_uart_getc (void)
- int litex_uart_init (unsigned long base)

Variables

- static volatile u32 * uart_base
- static struct sbi_console_device litex_console

30.368.1 Macro Definition Documentation

30.368.1.1 UART_EV_RX

```
#define UART_EV_RX 0x2
```

30.368.1.2 UART_REG_EV_ENABLE

```
#define UART_REG_EV_ENABLE 5
```

30.368.1.3 UART_REG_EV_PENDING

```
#define UART_REG_EV_PENDING 4
```

30.368.1.4 UART_REG_EV_STATUS

```
#define UART_REG_EV_STATUS 3
```

30.368.1.5 UART_REG_RXEMPTY

```
#define UART_REG_RXEMPTY 2
```

30.368.1.6 UART_REG_RXTX

```
#define UART_REG_RXTX 0
```

30.368.1.7 UART_REG_TXFULL

```
#define UART_REG_TXFULL 1
```

30.368.2 Function Documentation

30.368.2.1 get_reg()

```
static u8 get_reg (
            u8 reg )  [static]
```

Here is the caller graph for this function:

30.368.2.2 litex_uart_getc()

```
static int litex_uart_getc (
            void ) [static]
```

Here is the call graph for this function:

30.368.2.3 litex_uart_init()

```
int litex_uart_init (
            unsigned long base )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.368.2.4 litex_uart_putc()

```
static void litex_uart_putc (
            char ch ) [static]
```

Here is the call graph for this function:

30.368.2.5 set_reg()

```
static void set_reg (
            u8 reg,
            u8 val )  [static]
```

Here is the caller graph for this function:

30.368.3 Variable Documentation

30.368.3.1 litex_console

```
struct sbi_console_device litex_console  [static]
```

Initial value:

```
= {
    .name = "litex_uart",
    .console_putc = litex_uart_putc,
    .console_getc = litex_uart_getc
}
```

30.368.3.2 uart_base

```
volatile u32* uart_base  [static]
```

30.369 lib/utils/serial/renesas_scif.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_timer.h>
```

```
#include <sbi_utils/serial/renesas-scif.h>
```

Include dependency graph for renesas_scif.c:

Macros

- #define SCIF_REG_SMR 0x0
- #define SCIF_REG_BRR 0x2
- #define SCIF_REG_SCR 0x4
- #define SCIF_REG_FTDR 0x6
- #define SCIF_REG_FSR 0x8
- #define SCIF_REG_FCR 0xc
- #define SCIF_REG_LSR 0x12
- #define SCIF_REG_SEMR 0x14
- #define SCIF_FCR_RFRST 0x2 /∗ Reset assert receive-FIFO (bit[1]) ∗/
- #define SCIF_FCR_TFRST 0x4 /∗ Reset assert transmit-FIFO(bit[2]) ∗/
- #define SCIF_FCR_RST_ASSRT_RFTF (SCIF_FCR_RFRST | SCIF_FCR_TFRST) /∗ Reset assert tx-FIFO & rx-FIFO ∗/
- #define SCIF_FCR_RST_NGATE_RFTF 0x0 /∗ Reset negate tx-FIFO & rx-FIFO ∗/
- #define SCIF_SCR_RE 0x10 /∗ Enable receive (bit[4]) ∗/
- #define SCIF_SCR_TE 0x20 /∗ Enable transmit(bit[5]) ∗/
- #define SCIF_SCR_RCV_TRN_EN (SCIF_SCR_RE | SCIF_SCR_TE) /∗ Enable receive & transmit ∗/
- #define SCIF_SCR_RCV_TRN_DIS 0x0 /∗ Disable receive & transmit ∗/
- #define SCIF_FSR_ER 0x80 /∗ Receive error flag ∗/
- #define SCIF_FSR_TEND 0x40 /∗ Transmit End Flag ∗/
- #define SCIF_FSR_TDFE 0x20 /∗ Transmit FIFO Data Empty Flag ∗/
- #define SCIF_FSR_BRK 0x10 /∗ Detect break flag ∗/
- #define SCIF_FSR_DR 0x1 /∗ Receive data ready flag ∗/
- #define SCIF_FSR_TXD_CHK (SCIF_FSR_TEND | SCIF_FSR_TDFE)
- #define SCIF_SEMR_MDDRS 0x10 /∗ MDDR access enable ∗/
- #define SCIF_REG_8BIT(reg)
- #define SCBRR_VALUE(clk, baudrate) ((clk) / (32 ∗ (baudrate)) - 1)

Functions

- static u32 get_reg (u32 offset)
- static void set_reg (u32 offset, u32 val)
- static void renesas_scif_putc (char ch)
- int renesas_scif_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile char ∗ scif_base
- static struct sbi_console_device renesas_scif_console

30.369.1 Macro Definition Documentation

30.369.1.1 SCBRR_VALUE

```
#define SCBRR_VALUE(
            clk,
            baudrate ) ((clk) / (32 * (baudrate)) - 1)
```

30.369.1.2 SCIF_FCR_RFRST

```
#define SCIF_FCR_RFRST 0x2 /* Reset assert receive-FIFO (bit[1]) */
```

30.369.1.3 SCIF_FCR_RST_ASSRT_RFTF

```
#define SCIF_FCR_RST_ASSRT_RFTF (SCIF_FCR_RFRST | SCIF_FCR_TFRST) /* Reset assert tx-FIFO &
rx-FIFO */
```

30.369.1.4 SCIF_FCR_RST_NGATE_RFTF

```
#define SCIF_FCR_RST_NGATE_RFTF 0x0 /* Reset negate tx-FIFO & rx-FIFO */
```

30.369.1.5 SCIF_FCR_TFRST

```
#define SCIF_FCR_TFRST 0x4 /* Reset assert transmit-FIFO(bit[2]) */
```

30.369.1.6 SCIF_FSR_BRK

```
#define SCIF_FSR_BRK 0x10 /* Detect break flag */
```

30.369.1.7 SCIF_FSR_DR

```
#define SCIF_FSR_DR 0x1 /* Receive data ready flag */
```

30.369.1.8 SCIF_FSR_ER

```
#define SCIF_FSR_ER 0x80 /* Receive error flag */
```

30.369.1.9 SCIF_FSR_TDFE

```
#define SCIF_FSR_TDFE 0x20 /* Transmit FIFO Data Empty Flag */
```

30.369.1.10 SCIF_FSR_TEND

```
#define SCIF_FSR_TEND 0x40 /* Transmit End Flag */
```

30.369.1.11 SCIF_FSR_TXD_CHK

```
#define SCIF_FSR_TXD_CHK (SCIF_FSR_TEND | SCIF_FSR_TDFE)
```

30.369.1.12 SCIF_REG_8BIT

```
#define SCIF_REG_8BIT(
            reg )
```

Value:

```
 ((reg == SCIF_REG_BRR) || \
 (reg == SCIF_REG_FTDR) || \
 (reg == SCIF_REG_SEMR))
```

30.369.1.13 SCIF_REG_BRR

```
#define SCIF_REG_BRR 0x2
```

30.369.1.14 SCIF_REG_FCR

```
#define SCIF_REG_FCR 0xc
```

30.369.1.15 SCIF_REG_FSR

```
#define SCIF_REG_FSR 0x8
```

30.369.1.16 SCIF_REG_FTDR

```
#define SCIF_REG_FTDR 0x6
```

30.369.1.17 SCIF_REG_LSR

```
#define SCIF_REG_LSR 0x12
```

30.369.1.18 SCIF_REG_SCR

```
#define SCIF_REG_SCR 0x4
```

30.369.1.19 SCIF_REG_SEMR

```
#define SCIF_REG_SEMR 0x14
```

30.369.1.20 SCIF_REG_SMR

```
#define SCIF_REG_SMR 0x0
```

30.369.1.21 SCIF_SCR_RCV_TRN_DIS

```
#define SCIF_SCR_RCV_TRN_DIS 0x0 /* Disable receive & transmit */
```

30.369.1.22 SCIF_SCR_RCV_TRN_EN

```
#define SCIF_SCR_RCV_TRN_EN (SCIF_SCR_RE | SCIF_SCR_TE) /* Enable receive & transmit */
```

30.369.1.23 SCIF_SCR_RE

```
#define SCIF_SCR_RE 0x10 /* Enable receive (bit[4]) */
```

30.369.1.24 SCIF_SCR_TE

```
#define SCIF_SCR_TE 0x20 /* Enable transmit(bit[5]) */
```

30.369.1.25 SCIF_SEMR_MDDRS

```
#define SCIF_SEMR_MDDRS 0x10 /* MDDR access enable */
```

30.369.2 Function Documentation

30.369.2.1 get_reg()

```
static u32 get_reg (
            u32 offset )  [static]
```

Here is the caller graph for this function:

30.369.2.2 renesas_scif_init()

```
int renesas_scif_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.369.2.3 renesas_scif_putc()

```
static void renesas_scif_putc (
            char ch )  [static]
```

Here is the call graph for this function:

30.369.2.4 set_reg()

```
static void set_reg (
            u32 offset,
            u32 val )  [static]
```

Here is the caller graph for this function:

30.369.3 Variable Documentation

30.369.3.1 renesas_scif_console

```
struct sbi_console_device renesas_scif_console  [static]
```

Initial value:

```
= {
    .name       = "renesas_scif",
    .console_putc   = renesas_scif_putc,
}
```

30.369.3.2 scif_base

```
volatile char* scif_base  [static]
```

30.370 lib/utils/serial/semihosting.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/serial/semihosting.h>
```

Include dependency graph for semihosting.c:

Macros

- #define SYSOPEN 0x01
- #define SYSWRITEC 0x03
- #define SYSWRITE 0x05
- #define SYSREAD 0x06
- #define SYSREADC 0x07
- #define SYSERRNO 0x13

Functions

- static long semihosting_trap (int sysnum, void ∗addr)
- bool semihosting_enabled (void)
- static int semihosting_errno (void)
- static long semihosting_open (const char ∗fname, enum semihosting_open_mode mode)
- static long semihosting_read (long fd, void ∗memp, size_t len)
- static long semihosting_write (long fd, const void ∗memp, size_t len)
- static unsigned long semihosting_puts (const char ∗str, unsigned long len)
- static int semihosting_getc (void)
- int semihosting_init (void)

Variables

- static bool _semihosting_enabled = true
- static bool try_semihosting = true
- static int semihosting_infd = SBI_ENODEV
- static int semihosting_outfd = SBI_ENODEV
- static struct sbi_console_device semihosting_console

30.370.1 Macro Definition Documentation

30.370.1.1 SYSERRNO

```
#define SYSERRNO 0x13
```

30.370.1.2 SYSOPEN

```
#define SYSOPEN 0x01
```

30.370.1.3 SYSREAD

```
#define SYSREAD 0x06
```

30.370.1.4 SYSREADC

```
#define SYSREADC 0x07
```

30.370.1.5 SYSWRITE

```
#define SYSWRITE 0x05
```

30.370.1.6 SYSWRITEC

```
#define SYSWRITEC 0x03
```

30.370.2 Function Documentation

30.370.2.1 semihosting_enabled()

```
bool semihosting_enabled (
            void  )
```

30.370.2.2 semihosting_errno()

```
static int semihosting_errno (
            void  )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.370.2.3 semihosting_getc()

```
static int semihosting_getc (
            void  )  [static]
```

Here is the call graph for this function:

30.370.2.4 semihosting_init()

```
int semihosting_init (
            void  )
```

Here is the call graph for this function:

30.370.2.5 semihosting_open()

```
static long semihosting_open (
            const char * fname,
            enum semihosting_open_mode mode )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.370.2.6 semihosting_puts()

```
static unsigned long semihosting_puts (
            const char * str,
            unsigned long len )  [static]
```

Here is the call graph for this function:

30.370.2.7 semihosting_read()

```
static long semihosting_read (
            long fd,
            void * memp,
            size_t len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.370.2.8 semihosting_trap()

```
static long semihosting_trap (
            int sysnum,
            void * addr )  [static]
```

Here is the caller graph for this function:

30.370.2.9 semihosting_write()

```
static long semihosting_write (
            long fd,
            const void * memp,
            size_t len )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.370.3 Variable Documentation

30.370.3.1 _semihosting_enabled

```
bool _semihosting_enabled = true  [static]
```

30.370.3.2 semihosting_console

```
struct sbi_console_device semihosting_console  [static]
```

Initial value:

```
= {
    .name = "semihosting",
    .console_puts = semihosting_puts,
    .console_getc = semihosting_getc
}
```

30.370.3.3 semihosting_infd

```
int semihosting_infd = SBI_ENODEV  [static]
```

30.370.3.4 semihosting_outfd

```
int semihosting_outfd = SBI_ENODEV  [static]
```

30.370.3.5 try_semihosting

```
bool try_semihosting = true  [static]
```

30.371 lib/utils/serial/shakti-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/shakti-uart.h>
```

Include dependency graph for shakti-uart.c:

Macros

- #define REG_BAUD 0x00
- #define REG_TX 0x04
- #define REG_RX 0x08
- #define REG_STATUS 0x0C
- #define REG_DELAY 0x10
- #define REG_CONTROL 0x14
- #define REG_INT_EN 0x18
- #define REG_IQ_CYCLES 0x1C
- #define REG_RX_THRES 0x20
- #define UART_TX_FULL 0x2
- #define UART_RX_NOT_EMPTY 0x4
- #define UART_RX_FULL 0x8

Functions

- static void shakti_uart_putc (char ch)
- static int shakti_uart_getc (void)
- int shakti_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile char * uart_base
- static struct sbi_console_device shakti_console

30.371.1 Macro Definition Documentation

30.371.1.1 REG_BAUD

```
#define REG_BAUD 0x00
```

30.371.1.2 REG_CONTROL

```
#define REG_CONTROL 0x14
```

30.371.1.3 REG_DELAY

```
#define REG_DELAY 0x10
```

30.371.1.4 REG_INT_EN

```
#define REG_INT_EN 0x18
```

30.371.1.5 REG_IQ_CYCLES

```
#define REG_IQ_CYCLES 0x1C
```

30.371.1.6 REG_RX

```
#define REG_RX 0x08
```

30.371.1.7 REG_RX_THRES

```
#define REG_RX_THRES 0x20
```

30.371.1.8 REG_STATUS

```
#define REG_STATUS 0x0C
```

30.371.1.9 REG_TX

```
#define REG_TX 0x04
```

30.371.1.10 UART_RX_FULL

```
#define UART_RX_FULL 0x8
```

30.371.1.11 UART_RX_NOT_EMPTY

```
#define UART_RX_NOT_EMPTY 0x4
```

30.371.1.12 UART_TX_FULL

```
#define UART_TX_FULL 0x2
```

30.371.2 Function Documentation

30.371.2.1 shakti_uart_getc()

```
static int shakti_uart_getc (
            void ) [static]
```

30.371.2.2 shakti_uart_init()

```
int shakti_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.371.2.3 shakti_uart_putc()

```
static void shakti_uart_putc (
            char ch )  [static]
```

30.371.3 Variable Documentation

30.371.3.1 shakti_console

```
struct sbi_console_device shakti_console  [static]
```

Initial value:

```
= {
    .name = "shakti_uart",
    .console_putc = shakti_uart_putc,
    .console_getc = shakti_uart_getc
}
```

30.371.3.2 uart_base

```
volatile char* uart_base  [static]
```

30.372 lib/utils/serial/sifive-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/sifive-uart.h>
```

Include dependency graph for sifive-uart.c:

Macros

- #define UART_REG_TXFIFO 0
- #define UART_REG_RXFIFO 1
- #define UART_REG_TXCTRL 2
- #define UART_REG_RXCTRL 3
- #define UART_REG_IE 4
- #define UART_REG_IP 5
- #define UART_REG_DIV 6
- #define UART_TXFIFO_FULL 0x80000000
- #define UART_RXFIFO_EMPTY 0x80000000
- #define UART_RXFIFO_DATA 0x000000ff
- #define UART_TXCTRL_TXEN 0x1
- #define UART_RXCTRL_RXEN 0x1

Functions

- static unsigned int uart_min_clk_divisor (uint64_t in_freq, uint64_t max_target_hz)
- static u32 get_reg (u32 num)
- static void set_reg (u32 num, u32 val)
- static void sifive_uart_putc (char ch)
- static int sifive_uart_getc (void)
- int sifive_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile char * uart_base
- static u32 uart_in_freq
- static u32 uart_baudrate
- static struct sbi_console_device sifive_console

30.372.1 Macro Definition Documentation

30.372.1.1 UART_REG_DIV

```
#define UART_REG_DIV 6
```

30.372.1.2 UART_REG_IE

```
#define UART_REG_IE 4
```

30.372.1.3 UART_REG_IP

```
#define UART_REG_IP 5
```

30.372.1.4 UART_REG_RXCTRL

```
#define UART_REG_RXCTRL 3
```

30.372.1.5 UART_REG_RXFIFO

```
#define UART_REG_RXFIFO 1
```

30.372.1.6 UART_REG_TXCTRL

```
#define UART_REG_TXCTRL 2
```

30.372.1.7 UART_REG_TXFIFO

```
#define UART_REG_TXFIFO 0
```

30.372.1.8 UART_RXCTRL_RXEN

```
#define UART_RXCTRL_RXEN 0x1
```

30.372.1.9 UART_RXFIFO_DATA

```
#define UART_RXFIFO_DATA 0x000000ff
```

30.372.1.10 UART_RXFIFO_EMPTY

```
#define UART_RXFIFO_EMPTY 0x80000000
```

30.372.1.11 UART_TXCTRL_TXEN

```
#define UART_TXCTRL_TXEN 0x1
```

30.372.1.12 UART_TXFIFO_FULL

```
#define UART_TXFIFO_FULL 0x80000000
```

30.372.2 Function Documentation

30.372.2.1 get_reg()

```
static u32 get_reg (
            u32 num )  [static]
```

Here is the caller graph for this function:

30.372.2.2 set_reg()

```
static void set_reg (
            u32 num,
            u32 val )  [static]
```

Here is the caller graph for this function:

30.372.2.3 sifive_uart_getc()

```
static int sifive_uart_getc (
            void  )  [static]
```

Here is the call graph for this function:

30.372.2.4 sifive_uart_init()

```
int sifive_uart_init (
            unsigned long base,
```

```
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.372.2.5 sifive_uart_putc()

```
static void sifive_uart_putc (
            char ch )  [static]
```

Here is the call graph for this function:

30.372.2.6 uart_min_clk_divisor()

```
static unsigned int uart_min_clk_divisor (
            uint64_t in_freq,
            uint64_t max_target_hz )  [inline], [static]
```

Find minimum divisor divides in_freq to max_target_hz; Based on uart driver n SiFive FSBL.
f_baud = f_in / (div + 1) => div = (f_in / f_baud) - 1 The nearest integer solution requires rounding up as to not exceed max_target_hz. div = ceil(f_in / f_baud) - 1 = floor((f_in - 1 + f_baud) / f_baud) - 1 This should not overflow as long as (f_in - 1 + f_baud) does not exceed 2^32 - 1, which is unlikely since we represent frequencies in kHz.

Here is the caller graph for this function:

30.372.3 Variable Documentation

30.372.3.1 sifive_console

```
struct sbi_console_device sifive_console  [static]
```

Initial value:

```
= {
    .name = "sifive_uart",
    .console_putc = sifive_uart_putc,
    .console_getc = sifive_uart_getc
}
```

30.372.3.2 uart_base

```
volatile char* uart_base  [static]
```

30.372.3.3 uart_baudrate

```
u32 uart_baudrate  [static]
```

30.372.3.4 uart_in_freq

```
u32 uart_in_freq  [static]
```

30.373 lib/utils/serial/uart8250.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi_utils/serial/uart8250.h>
```

Include dependency graph for uart8250.c:

Macros

- #define UART_RBR_OFFSET 0 /* In: Recieve Buffer Register */
- #define UART_THR_OFFSET 0 /* Out: Transmitter Holding Register */
- #define UART_DLL_OFFSET 0 /* Out: Divisor Latch Low */
- #define UART_IER_OFFSET 1 /* I/O: Interrupt Enable Register */
- #define UART_DLM_OFFSET 1 /* Out: Divisor Latch High */
- #define UART_FCR_OFFSET 2 /* Out: FIFO Control Register */
- #define UART_IIR_OFFSET 2 /* I/O: Interrupt Identification Register */
- #define UART_LCR_OFFSET 3 /* Out: Line Control Register */
- #define UART_MCR_OFFSET 4 /* Out: Modem Control Register */
- #define UART_LSR_OFFSET 5 /* In: Line Status Register */
- #define UART_MSR_OFFSET 6 /* In: Modem Status Register */
- #define UART_SCR_OFFSET 7 /* I/O: Scratch Register */
- #define UART_MDR1_OFFSET 8 /* I/O: Mode Register */
- #define UART_LSR_FIFOE 0x80 /* Fifo error */
- #define UART_LSR_TEMT 0x40 /* Transmitter empty */
- #define UART_LSR_THRE 0x20 /* Transmit-hold-register empty */
- #define UART_LSR_BI 0x10 /* Break interrupt indicator */
- #define UART_LSR_FE 0x08 /* Frame error indicator */
- #define UART_LSR_PE 0x04 /* Parity error indicator */
- #define UART_LSR_OE 0x02 /* Overrun error indicator */
- #define UART_LSR_DR 0x01 /* Receiver data ready */
- #define UART_LSR_BRK_ERROR_BITS 0x1E /* BI, FE, PE, OE bits */

Functions

- static u32 get_reg (u32 num)
- static void set_reg (u32 num, u32 val)
- static void uart8250_putc (char ch)
- static int uart8250_getc (void)
- int uart8250_init (unsigned long base, u32 in_freq, u32 baudrate, u32 reg_shift, u32 reg_width, u32 reg_↩ offset)

Variables

- static volatile char * uart8250_base
- static u32 uart8250_in_freq
- static u32 uart8250_baudrate
- static u32 uart8250_reg_width
- static u32 uart8250_reg_shift
- static struct sbi_console_device uart8250_console

30.373.1 Macro Definition Documentation

30.373.1.1 UART_DLL_OFFSET

```
#define UART_DLL_OFFSET 0 /* Out:  Divisor Latch Low */
```

30.373.1.2 UART_DLM_OFFSET

```
#define UART_DLM_OFFSET 1 /* Out:  Divisor Latch High */
```

30.373.1.3 UART_FCR_OFFSET

```
#define UART_FCR_OFFSET 2 /* Out:  FIFO Control Register */
```

30.373.1.4 UART_IER_OFFSET

```
#define UART_IER_OFFSET 1 /* I/O: Interrupt Enable Register */
```

30.373.1.5 UART_IIR_OFFSET

```
#define UART_IIR_OFFSET 2 /* I/O: Interrupt Identification Register */
```

30.373.1.6 UART_LCR_OFFSET

```
#define UART_LCR_OFFSET 3 /* Out:  Line Control Register */
```

30.373.1.7 UART_LSR_BI

```
#define UART_LSR_BI 0x10 /* Break interrupt indicator */
```

30.373.1.8 UART_LSR_BRK_ERROR_BITS

```
#define UART_LSR_BRK_ERROR_BITS 0x1E /* BI, FE, PE, OE bits */
```

30.373.1.9 UART_LSR_DR

```
#define UART_LSR_DR 0x01 /* Receiver data ready */
```

30.373.1.10 UART_LSR_FE

```
#define UART_LSR_FE 0x08 /* Frame error indicator */
```

30.373.1.11 UART_LSR_FIFOE

```
#define UART_LSR_FIFOE 0x80 /* Fifo error */
```

30.373.1.12 UART_LSR_OE

```
#define UART_LSR_OE 0x02 /* Overrun error indicator */
```

30.373.1.13 UART_LSR_OFFSET

```
#define UART_LSR_OFFSET 5 /* In:  Line Status Register */
```

30.373.1.14 UART_LSR_PE

```
#define UART_LSR_PE 0x04 /* Parity error indicator */
```

30.373.1.15 UART_LSR_TEMT

```
#define UART_LSR_TEMT 0x40 /* Transmitter empty */
```

30.373.1.16 UART_LSR_THRE

```
#define UART_LSR_THRE 0x20 /* Transmit-hold-register empty */
```

30.373.1.17 UART_MCR_OFFSET

```
#define UART_MCR_OFFSET 4 /* Out:  Modem Control Register */
```

30.373.1.18 UART_MDR1_OFFSET

```
#define UART_MDR1_OFFSET 8 /* I/O: Mode Register */
```

30.373.1.19 UART_MSR_OFFSET

```
#define UART_MSR_OFFSET 6 /* In:  Modem Status Register */
```

30.373.1.20 UART_RBR_OFFSET

```
#define UART_RBR_OFFSET 0 /* In:  Recieve Buffer Register */
```

30.373.1.21 UART_SCR_OFFSET

```
#define UART_SCR_OFFSET 7 /* I/O: Scratch Register */
```

30.373.1.22 UART_THR_OFFSET

```
#define UART_THR_OFFSET 0 /* Out:  Transmitter Holding Register */
```

30.373.2 Function Documentation

30.373.2.1 get_reg()

```
static u32 get_reg (
            u32 num )  [static]
```

Here is the caller graph for this function:

30.373.2.2 set_reg()

```
static void set_reg (
            u32 num,
            u32 val )  [static]
```

Here is the caller graph for this function:

30.373.2.3 uart8250_getc()

```
static int uart8250_getc (
            void ) [static]
```

Here is the call graph for this function:

30.373.2.4 uart8250_init()

```
int uart8250_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate,
            u32 reg_shift,
            u32 reg_width,
            u32 reg_offset )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.373.2.5 uart8250_putc()

```
static void uart8250_putc (
            char ch ) [static]
```

Here is the call graph for this function:

30.373.3 Variable Documentation

30.373.3.1 uart8250_base

```
volatile char* uart8250_base [static]
```

30.373.3.2 uart8250_baudrate

```
u32 uart8250_baudrate [static]
```

30.373.3.3 uart8250_console

```
struct sbi_console_device uart8250_console [static]
```

Initial value:

```
= {
    .name = "uart8250",
    .console_putc = uart8250_putc,
    .console_getc = uart8250_getc
}
```

30.373.3.4 uart8250_in_freq

```
u32 uart8250_in_freq [static]
```

30.373.3.5 uart8250_reg_shift

```
u32 uart8250_reg_shift [static]
```

30.373.3.6 uart8250_reg_width

```
u32 uart8250_reg_width [static]
```

30.374 lib/utils/serial/xlnx-uartlite.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/xlnx_uartlite.h>
```

Include dependency graph for xlnx-uartlite.c:

Macros

- #define UART_RX_OFFSET 0x00
- #define UART_TX_OFFSET 0x04
- #define UART_STATUS_OFFSET 0x08
- #define UART_STATUS_RXVALID 0x01
- #define UART_STATUS_RXFULL 0x02
- #define UART_STATUS_TXEMPTY 0x04
- #define UART_STATUS_TXFULL 0x08
- #define UART_STATUS_IE 0x10
- #define UART_STATUS_OVERRUN 0x20
- #define UART_STATUS_FRAME 0x40
- #define UART_STATUS_PARITY 0x80
- #define UART_CTRL_OFFSET 0x0C
- #define UART_CTRL_RST_TX 0x01
- #define UART_CTRL_RST_RX 0x02
- #define UART_CTRL_IE 0x10

Functions

- static void xlnx_uartlite_putc (char ch)
- static int xlnx_uartlite_getc (void)
- int xlnx_uartlite_init (unsigned long base)

Variables

- static volatile char * xlnx_uartlite_base
- static struct sbi_console_device xlnx_uartlite_console

30.374.1 Macro Definition Documentation

30.374.1.1 UART_CTRL_IE

```
#define UART_CTRL_IE 0x10
```

30.374.1.2 UART_CTRL_OFFSET

```
#define UART_CTRL_OFFSET 0x0C
```

30.374.1.3 UART_CTRL_RST_RX

```
#define UART_CTRL_RST_RX 0x02
```

30.374.1.4 UART_CTRL_RST_TX

```
#define UART_CTRL_RST_TX 0x01
```

30.374.1.5 UART_RX_OFFSET

```
#define UART_RX_OFFSET 0x00
```

30.374.1.6 UART_STATUS_FRAME

```
#define UART_STATUS_FRAME 0x40
```

30.374.1.7 UART_STATUS_IE

```
#define UART_STATUS_IE 0x10
```

30.374.1.8 UART_STATUS_OFFSET

```
#define UART_STATUS_OFFSET 0x08
```

30.374.1.9 UART_STATUS_OVERRUN

```
#define UART_STATUS_OVERRUN 0x20
```

30.374.1.10 UART_STATUS_PARITY

```
#define UART_STATUS_PARITY 0x80
```

30.374.1.11 UART_STATUS_RXFULL

```
#define UART_STATUS_RXFULL 0x02
```

30.374.1.12 UART_STATUS_RXVALID

```
#define UART_STATUS_RXVALID 0x01
```

30.374.1.13 UART_STATUS_TXEMPTY

```
#define UART_STATUS_TXEMPTY 0x04
```

30.374.1.14 UART_STATUS_TXFULL

```
#define UART_STATUS_TXFULL 0x08
```

30.374.1.15 UART_TX_OFFSET

```
#define UART_TX_OFFSET 0x04
```

30.374.2 Function Documentation

30.374.2.1 xlnx_uartlite_getc()

```
static int xlnx_uartlite_getc (
            void  )  [static]
```

30.374.2.2 xlnx_uartlite_init()

```
int xlnx_uartlite_init (
            unsigned long base )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.374.2.3 xlnx_uartlite_putc()

```
static void xlnx_uartlite_putc (
            char ch )  [static]
```

30.374.3 Variable Documentation

30.374.3.1 xlnx_uartlite_base

```
volatile char* xlnx_uartlite_base  [static]
```

30.374.3.2 xlnx_uartlite_console

```
struct sbi_console_device xlnx_uartlite_console  [static]
```

Initial value:

```
= {
    .name = "xlnx-uartlite",
    .console_putc = xlnx_uartlite_putc,
    .console_getc = xlnx_uartlite_getc
}
```

30.375 lib/utils/suspend/fdt_suspend.c File Reference

```
#include <sbi_utils/suspend/fdt_suspend.h>
```

Include dependency graph for fdt_suspend.c:

Functions

- void fdt_suspend_init (const void *fdt)

Variables

- const struct fdt_driver *const fdt_suspend_drivers [ ]

30.375.1 Function Documentation

30.375.1.1 fdt_suspend_init()

```
void fdt_suspend_init (
            const void * fdt )
```

Here is the call graph for this function:

30.375.2 Variable Documentation

30.375.2.1 fdt_suspend_drivers

```
const struct fdt_driver* const fdt_suspend_drivers[] [extern]
```

30.376 lib/utils/suspend/fdt_suspend_rpmi.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_system.h>
#include <sbi/riscv_asm.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/mailbox/fdt_mailbox.h>
#include <sbi_utils/mailbox/mailbox.h>
#include <sbi_utils/mailbox/rpmi_mailbox.h>
#include <sbi_utils/suspend/fdt_suspend.h>
```

Include dependency graph for fdt_suspend_rpmi.c:

Functions

- static int rpmi_syssusp_attrs (uint32_t *attrs)
- static int rpmi_syssusp (uint32_t suspend_type, ulong resume_addr)
- static int rpmi_system_suspend_check (u32 sleep_type)
- static int rpmi_system_suspend (u32 sleep_type, ulong resume_addr)
- static int rpmi_suspend_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static struct rpmi_syssusp syssusp_ctx
- static struct sbi_system_suspend_device rpmi_suspend_dev
- static const struct fdt_match rpmi_suspend_match [ ]
- struct fdt_driver fdt_suspend_rpmi

30.376.1 Function Documentation

30.376.1.1 rpmi_suspend_init()

```
static int rpmi_suspend_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.376.1.2 rpmi_syssusp()

```
static int rpmi_syssusp (
            uint32_t suspend_type,
            ulong resume_addr )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.376.1.3 rpmi_syssusp_attrs()

```
static int rpmi_syssusp_attrs (
            uint32_t * attrs )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.376.1.4 rpmi_system_suspend()

```
static int rpmi_system_suspend (
            u32 sleep_type,
            ulong resume_addr )  [static]
```

Here is the call graph for this function:

30.376.1.5 rpmi_system_suspend_check()

```
static int rpmi_system_suspend_check (
            u32 sleep_type )  [static]
```

30.376.2 Variable Documentation

30.376.2.1 fdt_suspend_rpmi

```
struct fdt_driver fdt_suspend_rpmi
```

Initial value:

```
= {
    .match_table = rpmi_suspend_match,
    .init = rpmi_suspend_init,
    .experimental = true,
}
```

30.376.2.2 rpmi_suspend_dev

```
struct sbi_system_suspend_device rpmi_suspend_dev  [static]
```

Initial value:

```
= {
    .name = "rpmi-system-suspend",
    .system_suspend_check = rpmi_system_suspend_check,
    .system_suspend = rpmi_system_suspend,
}
```

30.376.2.3 rpmi_suspend_match

```
const struct fdt_match rpmi_suspend_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,rpmi-system-suspend" },
    {},
}
```

30.376.2.4 syssusp_ctx

```
struct rpmi_syssusp syssusp_ctx  [static]
```

30.377 lib/utils/sys/atcsmu.c File Reference

```
#include <sbi_utils/sys/atcsmu.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_bitops.h>
```

Include dependency graph for atcsmu.c:

Functions

- int smu_set_wakeup_events (struct smu_data *smu, u32 events, u32 hartid)
- bool smu_support_sleep_mode (struct smu_data *smu, u32 sleep_mode, u32 hartid)
- int smu_set_command (struct smu_data *smu, u32 pcs_ctl, u32 hartid)
- int smu_set_reset_vector (struct smu_data *smu, ulong wakeup_addr, u32 hartid)

30.377.1 Function Documentation

30.377.1.1 smu_set_command()

```
int smu_set_command (
            struct smu_data * smu,
            u32 pcs_ctl,
            u32 hartid )  [inline]
```

30.377.1.2 smu_set_reset_vector()

```
int smu_set_reset_vector (
            struct smu_data * smu,
            ulong wakeup_addr,
            u32 hartid )  [inline]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.377.1.3 smu_set_wakeup_events()

```
int smu_set_wakeup_events (
            struct smu_data * smu,
            u32 events,
            u32 hartid )  [inline]
```

30.377.1.4 smu_support_sleep_mode()

```
bool smu_support_sleep_mode (
            struct smu_data * smu,
            u32 sleep_mode,
            u32 hartid )  [inline]
```

Here is the call graph for this function:

30.378 lib/utils/sys/htif.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_system.h>
#include <sbi_utils/sys/htif.h>
```

Include dependency graph for htif.c:

Macros

- #define HTIF_DATA_BITS 48
- #define HTIF_DATA_MASK ((1ULL << HTIF_DATA_BITS) - 1)
- #define HTIF_DATA_SHIFT 0
- #define HTIF_CMD_BITS 8
- #define HTIF_CMD_MASK ((1ULL << HTIF_CMD_BITS) - 1)
- #define HTIF_CMD_SHIFT 48
- #define HTIF_DEV_BITS 8
- #define HTIF_DEV_MASK ((1ULL << HTIF_DEV_BITS) - 1)
- #define HTIF_DEV_SHIFT 56
- #define HTIF_DEV_SYSTEM 0
- #define HTIF_DEV_CONSOLE 1
- #define HTIF_CONSOLE_CMD_GETC 0
- #define HTIF_CONSOLE_CMD_PUTC 1
- #define TOHOST_CMD(dev, cmd, payload)
- #define FROMHOST_DEV(fromhost_value) ((uint64_t)((fromhost_value) >> HTIF_DEV_SHIFT) & HTIF_DEV_MASK)
- #define FROMHOST_CMD(fromhost_value) ((uint64_t)((fromhost_value) >> HTIF_CMD_SHIFT) & HTIF_CMD_MASK)
- #define FROMHOST_DATA(fromhost_value) ((uint64_t)((fromhost_value) >> HTIF_DATA_SHIFT) & HTIF_DATA_MASK)
- #define PK_SYS_write 64

Functions

- volatile uint64_t tohost __attribute__ ((section(".htif")))
- static uint64_t __read_tohost (void)
- static void __write_tohost (uint64_t val)
- static uint64_t __read_fromhost (void)
- static void __write_fromhost (uint64_t val)
- static void __check_fromhost ()
- static void __set_tohost (uint64_t dev, uint64_t cmd, uint64_t data)
- static int set_custom_addr (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom_tohost_addr)
- static void htif_putc (char ch)
- static int htif_getc (void)
- int htif_serial_init (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom_tohost↩_addr)
- static int htif_system_reset_check (u32 type, u32 reason)
- static void htif_system_reset (u32 type, u32 reason)
- int htif_system_reset_init (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom↩_tohost_addr)

Variables

- static uint64_t * htif_fromhost = NULL
- static uint64_t * htif_tohost = NULL
- static bool htif_custom = false
- static int htif_console_buf
- static spinlock_t htif_lock = SPIN_LOCK_INITIALIZER
- static struct sbi_console_device htif_console
- static struct sbi_system_reset_device htif_reset

30.378.1 Macro Definition Documentation

30.378.1.1 FROMHOST_CMD

```
#define FROMHOST_CMD(
            fromhost_value )  ((uint64_t)((fromhost_value) >> HTIF_CMD_SHIFT) & HTIF_CMD_MASK)
```

30.378.1.2 FROMHOST_DATA

```
#define FROMHOST_DATA(
            fromhost_value )  ((uint64_t)((fromhost_value) >> HTIF_DATA_SHIFT) & HTIF_DATA_MASK)
```

30.378.1.3 FROMHOST_DEV

```
#define FROMHOST_DEV(
            fromhost_value )  ((uint64_t)((fromhost_value) >> HTIF_DEV_SHIFT) & HTIF_DEV_MASK)
```

30.378.1.4 HTIF_CMD_BITS

```
#define HTIF_CMD_BITS 8
```

30.378.1.5 HTIF_CMD_MASK

```
#define HTIF_CMD_MASK ((1ULL << HTIF_CMD_BITS) - 1)
```

30.378.1.6 HTIF_CMD_SHIFT

```
#define HTIF_CMD_SHIFT 48
```

30.378.1.7 HTIF_CONSOLE_CMD_GETC

```
#define HTIF_CONSOLE_CMD_GETC 0
```

30.378.1.8 HTIF_CONSOLE_CMD_PUTC

```
#define HTIF_CONSOLE_CMD_PUTC 1
```

30.378.1.9 HTIF_DATA_BITS

```
#define HTIF_DATA_BITS 48
```

30.378.1.10 HTIF_DATA_MASK

```
#define HTIF_DATA_MASK ((1ULL << HTIF_DATA_BITS) - 1)
```

30.378.1.11 HTIF_DATA_SHIFT

```
#define HTIF_DATA_SHIFT 0
```

30.378.1.12 HTIF_DEV_BITS

```
#define HTIF_DEV_BITS 8
```

30.378.1.13 HTIF_DEV_CONSOLE

```
#define HTIF_DEV_CONSOLE 1
```

30.378.1.14 HTIF_DEV_MASK

```
#define HTIF_DEV_MASK ((1ULL << HTIF_DEV_BITS) - 1)
```

30.378.1.15 HTIF_DEV_SHIFT

```
#define HTIF_DEV_SHIFT 56
```

30.378.1.16 HTIF_DEV_SYSTEM

```
#define HTIF_DEV_SYSTEM 0
```

30.378.1.17 PK_SYS_write

```
#define PK_SYS_write 64
```

30.378.1.18 TOHOST_CMD

```
#define TOHOST_CMD(
            dev,
            cmd,
            payload )
```

Value:

```
({ \
if ((dev) || (cmd)) __builtin_trap(); \
(payload); })
```

30.378.2 Function Documentation

30.378.2.1 __attribute__()

```
volatile uint64_t fromhost __attribute__ (
            (section(".htif"))  )
```

30.378.2.2 __check_fromhost()

```
static void __check_fromhost ( )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.378.2.3 __read_fromhost()

```
static uint64_t __read_fromhost (
            void ) [inline], [static]
```

Here is the caller graph for this function:

30.378.2.4 __read_tohost()

```
static uint64_t __read_tohost (
            void ) [inline], [static]
```

Here is the caller graph for this function:

30.378.2.5 __set_tohost()

```
static void __set_tohost (
            uint64_t dev,
            uint64_t cmd,
            uint64_t data ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.378.2.6 __write_fromhost()

```
static void __write_fromhost (
            uint64_t val )  [inline], [static]
```

Here is the caller graph for this function:

30.378.2.7 __write_tohost()

```
static void __write_tohost (
            uint64_t val )  [inline], [static]
```

Here is the caller graph for this function:

30.378.2.8 htif_getc()

```
static int htif_getc (
            void  )  [static]
```

Here is the call graph for this function:

30.378.2.9 htif_putc()

```
static void htif_putc (
            char ch )  [static]
```

Here is the call graph for this function:

30.378.2.10 htif_serial_init()

```
int htif_serial_init (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.378.2.11 htif_system_reset()

```
static void htif_system_reset (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:

30.378.2.12 htif_system_reset_check()

```
static int htif_system_reset_check (
            u32 type,
            u32 reason )  [static]
```

30.378.2.13 htif_system_reset_init()

```
int htif_system_reset_init (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.378.2.14 set_custom_addr()

```
static int set_custom_addr (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )  [static]
```

Here is the caller graph for this function:

30.378.3 Variable Documentation

30.378.3.1 htif_console

```
struct sbi_console_device htif_console  [static]
```

Initial value:

```
= {
    .name = "htif",
    .console_putc = htif_putc,
    .console_getc = htif_getc
}
```

30.378.3.2 htif_console_buf

```
int htif_console_buf  [static]
```

30.378.3.3 htif_custom

```
bool htif_custom = false  [static]
```

30.378.3.4 htif_fromhost

```
uint64_t* htif_fromhost = NULL  [static]
```

30.378.3.5 htif_lock

```
spinlock_t htif_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.378.3.6 htif_reset

```
struct sbi_system_reset_device htif_reset  [static]
```

Initial value:

```
= {
    .name = "htif",
    .system_reset_check = htif_system_reset_check,
    .system_reset = htif_system_reset
}
```

30.378.3.7 htif_tohost

```
uint64_t* htif_tohost = NULL  [static]
```

30.379 lib/utils/timer/aclint_mtimer.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/timer/aclint_mtimer.h>
```

Include dependency graph for aclint_mtimer.c:

Macros

- #define mtimer_get_hart_data_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, mtimer_ptr_offset)
- #define mtimer_set_hart_data_ptr(__scratch, __mtimer) sbi_scratch_write_type((__scratch), void *, mtimer_ptr_offset, (__mtimer))

Functions

- static u64 mtimer_time_rd64 (volatile u64 *addr)
- static void mtimer_time_wr64 (bool timecmp, u64 value, volatile u64 *addr)
- static u64 mtimer_time_rd32 (volatile u64 *addr)
- static void mtimer_time_wr32 (bool timecmp, u64 value, volatile u64 *addr)
- static u64 mtimer_value (void)
- static void mtimer_event_stop (void)
- static void mtimer_event_start (u64 next_event)
- void aclint_mtimer_sync (struct aclint_mtimer_data *mt)
- void aclint_mtimer_set_reference (struct aclint_mtimer_data *mt, struct aclint_mtimer_data *ref)
- static int aclint_mtimer_warm_init (void)
- int aclint_mtimer_cold_init (struct aclint_mtimer_data *mt, struct aclint_mtimer_data *reference)

Variables

- static unsigned long mtimer_ptr_offset
- static struct sbi_timer_device mtimer

30.379.1 Macro Definition Documentation

30.379.1.1 mtimer_get_hart_data_ptr

```
#define mtimer_get_hart_data_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, mtimer_ptr_offset)
```

30.379.1.2 mtimer_set_hart_data_ptr

```
#define mtimer_set_hart_data_ptr(
            __scratch,
            __mtimer )  sbi_scratch_write_type((__scratch), void *, mtimer_ptr_offset, (__↩
mtimer))
```

30.379.2 Function Documentation

30.379.2.1 aclint_mtimer_cold_init()

```
int aclint_mtimer_cold_init (
            struct aclint_mtimer_data * mt,
            struct aclint_mtimer_data * reference )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.379.2.2 aclint_mtimer_set_reference()

```
void aclint_mtimer_set_reference (
            struct aclint_mtimer_data * mt,
            struct aclint_mtimer_data * ref )
```

Here is the caller graph for this function:

30.379.2.3 aclint_mtimer_sync()

```
void aclint_mtimer_sync (
            struct aclint_mtimer_data * mt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.379.2.4 aclint_mtimer_warm_init()

```
static int aclint_mtimer_warm_init (
            void ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.379.2.5 mtimer_event_start()

```
static void mtimer_event_start (
            u64 next_event )  [static]
```

30.379.2.6 mtimer_event_stop()

```
static void mtimer_event_stop (
            void  )  [static]
```

30.379.2.7 mtimer_time_rd32()

```
static u64 mtimer_time_rd32 (
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:

30.379.2.8 mtimer_time_rd64()

```
static u64 mtimer_time_rd64 (
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:

30.379.2.9 mtimer_time_wr32()

```
static void mtimer_time_wr32 (
            bool timecmp,
            u64 value,
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:

30.379.2.10 mtimer_time_wr64()

```
static void mtimer_time_wr64 (
            bool timecmp,
```

```
            u64 value,
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:

30.379.2.11 mtimer_value()

```
static u64 mtimer_value (
            void  )  [static]
```

30.379.3 Variable Documentation

30.379.3.1 mtimer

```
struct sbi_timer_device mtimer  [static]
```

Initial value:

```
= {
    .name = "aclint-mtimer",
    .timer_value = mtimer_value,
    .timer_event_start = mtimer_event_start,
    .timer_event_stop = mtimer_event_stop
}
```

30.379.3.2 mtimer_ptr_offset

```
unsigned long mtimer_ptr_offset  [static]
```

30.380 lib/utils/timer/andes_plmt.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/timer/andes_plmt.h>
```

Include dependency graph for andes_plmt.c:

Functions

- static u64 plmt_timer_value (void)

- static void plmt_timer_event_stop (void)
- static void plmt_timer_event_start (u64 next_event)
- static int plmt_warm_timer_init (void)
- int plmt_cold_timer_init (struct plmt_data *plmt)

Variables

- struct plmt_data plmt
- static struct sbi_timer_device plmt_timer

30.380.1 Function Documentation

30.380.1.1 plmt_cold_timer_init()

```
int plmt_cold_timer_init (
            struct plmt_data * plmt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.380.1.2 plmt_timer_event_start()

```
static void plmt_timer_event_start (
            u64 next_event )  [static]
```

30.380.1.3 plmt_timer_event_stop()

```
static void plmt_timer_event_stop (
            void  )  [static]
```

Here is the caller graph for this function:

30.380.1.4 plmt_timer_value()

```
static u64 plmt_timer_value (
            void  )  [static]
```

30.380.1.5 plmt_warm_timer_init()

```
static int plmt_warm_timer_init (
            void  )  [static]
```

Here is the call graph for this function:

30.380.2 Variable Documentation

30.380.2.1 plmt

```
struct plmt_data plmt
```

30.380.2.2 plmt_timer

```
struct sbi_timer_device plmt_timer  [static]
```

Initial value:

```
= {
    .name           = "andes_plmt",
    .timer_freq     = DEFAULT_AE350_PLMT_FREQ,
    .timer_value        = plmt_timer_value,
    .timer_event_start = plmt_timer_event_start,
    .timer_event_stop  = plmt_timer_event_stop,
    .warm_init      = plmt_warm_timer_init,
}
```

30.381 lib/utils/timer/fdt_timer.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/timer/fdt_timer.h>
```

Include dependency graph for fdt_timer.c:

Functions

- int fdt_timer_init (void)

Variables

- const struct fdt_driver *const fdt_timer_drivers [ ]

30.381.1 Function Documentation

30.381.1.1 fdt_timer_init()

```
int fdt_timer_init (
            void )
```

Here is the call graph for this function:

30.381.2 Variable Documentation

30.381.2.1 fdt_timer_drivers

```
const struct fdt_driver* const fdt_timer_drivers[] [extern]
```

30.382 lib/utils/timer/fdt_timer_mtimer.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_list.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/timer/fdt_timer.h>
#include <sbi_utils/timer/aclint_mtimer.h>
```

Include dependency graph for fdt_timer_mtimer.c:

Functions

- static SBI_LIST_HEAD (mtn_list)
- static int timer_mtimer_cold_init (const void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static struct aclint_mtimer_data ∗ mt_reference = NULL
- static const struct timer_mtimer_quirks sifive_clint_quirks
- static const struct timer_mtimer_quirks thead_clint_quirks
- static const struct timer_mtimer_quirks thead_aclint_quirks
- static const struct fdt_match timer_mtimer_match [ ]
- const struct fdt_driver fdt_timer_mtimer

30.382.1 Function Documentation

30.382.1.1 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            mtn_list  )  [static]
```

30.382.1.2 timer_mtimer_cold_init()

```
static int timer_mtimer_cold_init (
            const void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.382.2 Variable Documentation

30.382.2.1 fdt_timer_mtimer

```
const struct fdt_driver fdt_timer_mtimer
```

Initial value:

```
= {
    .match_table = timer_mtimer_match,
    .init = timer_mtimer_cold_init,
}
```

30.382.2.2 mt_reference

```
struct aclint_mtimer_data* mt_reference = NULL  [static]
```

30.382.2.3 sifive_clint_quirks

```
const struct timer_mtimer_quirks sifive_clint_quirks  [static]
```

Initial value:

```
= {
    .is_clint        = true,
    .clint_mtime_offset = CLINT_MTIMER_OFFSET,
    .has_64bit_mmio      = true,
}
```

30.382.2.4 thead_aclint_quirks

```
const struct timer_mtimer_quirks thead_aclint_quirks  [static]
```

Initial value:

```
= {
    .has_64bit_mmio      = false,
}
```

30.382.2.5 thead_clint_quirks

```
const struct timer_mtimer_quirks thead_clint_quirks  [static]
```

Initial value:

```
= {
    .is_clint        = true,
    .clint_mtime_offset = CLINT_MTIMER_OFFSET,
    .clint_without_mtime     = true,
}
```

30.382.2.6 timer_mtimer_match

```
const struct fdt_match timer_mtimer_match[] [static]
```

Initial value:

```
= {
    { .compatible = "riscv,clint0", .data = &sifive_clint_quirks },
    { .compatible = "sifive,clint0", .data = &sifive_clint_quirks },
    { .compatible = "thead,c900-clint", .data = &thead_clint_quirks },
    { .compatible = "thead,c900-aclint-mtimer",
      .data = &thead_aclint_quirks },
    { .compatible = "riscv,aclint-mtimer" },
    { },
}
```

30.383 lib/utils/timer/fdt_timer_plmt.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/timer/fdt_timer.h>
#include <sbi_utils/timer/andes_plmt.h>
```

Include dependency graph for fdt_timer_plmt.c:

Functions

- static int fdt_plmt_cold_timer_init (const void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- struct plmt_data plmt
- static const struct fdt_match timer_plmt_match [ ]
- const struct fdt_driver fdt_timer_plmt

30.383.1 Function Documentation

30.383.1.1 fdt_plmt_cold_timer_init()

```
static int fdt_plmt_cold_timer_init (
            const void * fdt,
```

```
int nodeoff,
const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

30.383.2 Variable Documentation

30.383.2.1 fdt_timer_plmt

```
const struct fdt_driver fdt_timer_plmt
```

Initial value:

```
= {
    .match_table = timer_plmt_match,
    .init = fdt_plmt_cold_timer_init,
}
```

30.383.2.2 plmt

```
struct plmt_data plmt  [extern]
```

30.383.2.3 timer_plmt_match

```
const struct fdt_match timer_plmt_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "andestech,plmt0" },
    {},
}
```

30.384 README.md File Reference

Index

_AC
sbi_const.h, 334
_AT
sbi_const.h, 334
_BITUL
sbi_const.h, 334
_BITULL
sbi_const.h, 334
_QUAD_HIGHWORD
types.h, 1449
_QUAD_LOWWORD
types.h, 1449
_UL
sbi_const.h, 334
_ULL
sbi_const.h, 335
__AC
sbi_const.h, 334
__ASM_STR
riscv_asm.h, 167
__FBSDID
cdefs.h, 1447
divdi3.c, 1446
moddi3.c, 1450
qdivrem.c, 1452
udivdi3.c, 1459
umoddi3.c, 1460
__SBI_HART_EXT_DATA
sbi_hart.c, 929
__SPIN_LOCK_UNLOCKED
riscv_locks.h, 300
__STR
sbi_const.h, 334
__aligned, 77
next, 77
owner, 77
sbi_types.h, 619
__always_inline
sbi_types.h, 619
__attribute__
htif.c, 1591
sbi_hart.c, 929
sbi_hart.h, 430
__axchg
riscv_atomic.c, 785
__bitmap_and
sbi_bitmap.c, 796
sbi_bitmap.h, 306
__bitmap_or
sbi_bitmap.c, 796
sbi_bitmap.h, 306
__bitmap_xor
sbi_bitmap.c, 797
sbi_bitmap.h, 306
__change_bit
sbi_bitops.h, 315
__check_csr
sbi_hart.c, 926
__check_csr_16
sbi_hart.c, 926
__check_csr_2
sbi_hart.c, 927
__check_csr_32
sbi_hart.c, 927
__check_csr_4
sbi_hart.c, 927
__check_csr_64
sbi_hart.c, 927
__check_csr_8
sbi_hart.c, 927
__check_ext_csr
sbi_hart.c, 928
__check_fromhost
htif.c, 1591
__check_hpm_csr
sbi_hart.c, 928
__check_hpm_csr_16
sbi_hart.c, 928
__check_hpm_csr_2
sbi_hart.c, 928
__check_hpm_csr_4
sbi_hart.c, 928
__check_hpm_csr_8
sbi_hart.c, 928
__check_priv
sbi_hart.c, 929
__clear_bit
sbi_bitops.h, 315
__cmpdi2
quad.h, 1455
__divdi3
divdi3.c, 1446
quad.h, 1455
__fdt_parse_domain
fdt_domain.c, 1134
__fdt_parse_region
fdt_domain.c, 1135
__fixup_count_disable_devices

fdt_domain.c, 1135
__fixup_disable_devices
fdt_domain.c, 1136
__fixup_find_domain_offset
fdt_domain.c, 1136
__has_builtin
sbi_types.h, 619
__io_ar
riscv_io.h, 295
__io_aw
riscv_io.h, 295
__io_br
riscv_io.h, 295
__io_bw
riscv_io.h, 295
__io_rar
riscv_io.h, 295
__io_raw
riscv_io.h, 295
__io_rbr
riscv_io.h, 295
__io_rbw
riscv_io.h, 295
__moddi3
moddi3.c, 1450
quad.h, 1455
__mpxy_mbox_send_message
fdt_mpxy_rpmi_mbox.c, 1485
__noreturn
sbi_types.h, 619
__packed
fdt_mailbox_rpmi_shmem.c, 1474
fw_dynamic.h, 164
rpmi_msgprot.h, 725
sbi_types.h, 619
__printf
sbi_console.h, 327
__qdivrem
qdivrem.c, 1452
quad.h, 1455
__raw_readb
riscv_io.h, 297
__raw_readl
riscv_io.h, 297
__raw_readq
riscv_io.h, 297
__raw_readw
riscv_io.h, 297
__raw_writeb
riscv_io.h, 297
__raw_writel
riscv_io.h, 297
__raw_writeq
riscv_io.h, 297
__raw_writew
riscv_io.h, 297
__read_fromhost
htif.c, 1592
__read_tohost
htif.c, 1592
__sbi_expected_trap
sbi_hart.c, 929
__sbi_expected_trap_hext
sbi_hart.c, 929
__sbi_fifo_dequeue
sbi_fifo.c, 906
__sbi_fifo_enqueue
sbi_fifo.c, 906
__sbi_fifo_is_empty
sbi_fifo.c, 907
__sbi_fifo_is_full
sbi_fifo.c, 908
__sbi_fifo_reset
sbi_fifo.c, 908
__sbi_hart_update_extension
sbi_hart.c, 930
__sbi_hfence_gvma_all
sbi_hfence.h, 466
__sbi_hfence_gvma_gpa
sbi_hfence.h, 466
__sbi_hfence_gvma_vmid
sbi_hfence.h, 466
__sbi_hfence_gvma_vmid_gpa
sbi_hfence.h, 467
__sbi_hfence_vvma_all
sbi_hfence.h, 467
__sbi_hfence_vvma_asid
sbi_hfence.h, 467
__sbi_hfence_vvma_asid_va
sbi_hfence.h, 467
__sbi_hfence_vvma_va
sbi_hfence.h, 468
__sbi_hsm_hart_change_state
sbi_hsm.c, 955
__sbi_hsm_hart_get_state
sbi_hsm.c, 955
sbi_hsm.h, 470
__sbi_hsm_suspend_default
sbi_hsm.c, 955
__sbi_hsm_suspend_non_ret_restore
sbi_hsm.c, 956
__sbi_hsm_suspend_non_ret_save
sbi_hsm.c, 956
sbi_hsm.h, 470
__sbi_list_add
sbi_list.h, 499
__sbi_list_del
sbi_list.h, 500
__sbi_list_del_entry
sbi_list.h, 501
__set_bit
sbi_bitops.h, 316
__set_menvcfg_ext
sbi_hart.c, 929
__set_tohost
htif.c, 1592

__smp_load_acquire
riscv_barrier.h, 185
__smp_store_release
riscv_barrier.h, 185
__smq_queue_empty
fdt_mailbox_rpmi_shmem.c, 1466
__smq_queue_full
fdt_mailbox_rpmi_shmem.c, 1467
__smq_rx
fdt_mailbox_rpmi_shmem.c, 1467
__smq_tx
fdt_mailbox_rpmi_shmem.c, 1467
__test_and_clear_bit
sbi_bitops.h, 316
__test_and_set_bit
sbi_bitops.h, 316
__test_bit
sbi_bitops.h, 316
__ucmpdi2
quad.h, 1456
__udivdi3
quad.h, 1456
udivdi3.c, 1459
__umoddi3
quad.h, 1457
umoddi3.c, 1460
__write_fromhost
htif.c, 1593
__write_tohost
htif.c, 1593
_conv_cast
sbi_byteorder.h, 323
_semihosting_enabled
semihosting.c, 1566

a0
sbi_trap_regs, 156
a1
sbi_trap_regs, 156
a2
sbi_trap_regs, 156
a3
sbi_trap_regs, 156
a4
sbi_trap_regs, 156
a5
sbi_trap_regs, 156
a6
sbi_trap_regs, 156
a7
sbi_trap_regs, 156
ACLINT_DEFAULT_MTIME_OFFSET
aclint_mtimer.h, 769
ACLINT_DEFAULT_MTIME_SIZE
aclint_mtimer.h, 769
ACLINT_DEFAULT_MTIMECMP_OFFSET
aclint_mtimer.h, 769
ACLINT_DEFAULT_MTIMECMP_SIZE
aclint_mtimer.h, 769
aclint_mswi
aclint_mswi.c, 1222
aclint_mswi.c
aclint_mswi, 1222
aclint_mswi_cold_init, 1221
mswi_get_hart_data_ptr, 1221
mswi_ipi_clear, 1222
mswi_ipi_send, 1222
mswi_ptr_offset, 1222
mswi_set_hart_data_ptr, 1221
aclint_mswi.h
ACLINT_MSWI_ALIGN, 684
aclint_mswi_cold_init, 684
ACLINT_MSWI_MAX_HARTS, 684
ACLINT_MSWI_SIZE, 684
CLINT_MSWI_OFFSET, 684
ACLINT_MSWI_ALIGN
aclint_mswi.h, 684
aclint_mswi_cold_init
aclint_mswi.c, 1221
aclint_mswi.h, 684
aclint_mswi_data, 77
addr, 77
first_hartid, 77
hart_count, 77
size, 78
ACLINT_MSWI_MAX_HARTS
aclint_mswi.h, 684
ACLINT_MSWI_SIZE
aclint_mswi.h, 684
aclint_mtimer.c
aclint_mtimer_cold_init, 1598
aclint_mtimer_set_reference, 1598
aclint_mtimer_sync, 1599
aclint_mtimer_warm_init, 1599
mtimer, 1601
mtimer_event_start, 1599
mtimer_event_stop, 1600
mtimer_get_hart_data_ptr, 1598
mtimer_ptr_offset, 1601
mtimer_set_hart_data_ptr, 1598
mtimer_time_rd32, 1600
mtimer_time_rd64, 1600
mtimer_time_wr32, 1600
mtimer_time_wr64, 1600
mtimer_value, 1601
aclint_mtimer.h
ACLINT_DEFAULT_MTIME_OFFSET, 769
ACLINT_DEFAULT_MTIME_SIZE, 769
ACLINT_DEFAULT_MTIMECMP_OFFSET, 769
ACLINT_DEFAULT_MTIMECMP_SIZE, 769
ACLINT_MTIMER_ALIGN, 769
aclint_mtimer_cold_init, 770
ACLINT_MTIMER_MAX_HARTS, 770
aclint_mtimer_set_reference, 770
aclint_mtimer_sync, 771
CLINT_MTIMER_OFFSET, 770
MTIMER_REGION_ALIGN, 770

ACLINT_MTIMER_ALIGN
aclint_mtimer.h, 769
aclint_mtimer_cold_init
aclint_mtimer.c, 1598
aclint_mtimer.h, 770
aclint_mtimer_data, 78
first_hartid, 78
hart_count, 78
has_64bit_mmio, 78
has_shared_mtime, 78
mtime_addr, 78
mtime_freq, 78
mtime_size, 78
mtimecmp_addr, 78
mtimecmp_size, 79
time_delta_computed, 79
time_delta_reference, 79
time_rd, 79
time_wr, 79
ACLINT_MTIMER_MAX_HARTS
aclint_mtimer.h, 770
aclint_mtimer_set_reference
aclint_mtimer.c, 1598
aclint_mtimer.h, 770
aclint_mtimer_sync
aclint_mtimer.c, 1599
aclint_mtimer.h, 771
aclint_mtimer_warm_init
aclint_mtimer.c, 1599
adapter
dw_i2c_adapter, 82
add_return_test
riscv_atomic_test.c, 1111
addr
aclint_mswi_data, 77
aplic_data, 79
dw_i2c_adapter, 82
imsic_regs, 93
platform_uart_data, 97
plic_data, 98
plicsw_data, 98
smu_data, 159
address
fdt_reserve_entry, 88
ae350_system_reset
fdt_reset_atcwdt200.c, 1509
ae350_system_reset_check
fdt_reset_atcwdt200.c, 1509
alloc_with_align
sbi_heap.c, 946
Andes AE350 SoC Platform, 37
andes_plicsw.c
plicsw, 1224
plicsw_cold_ipi_init, 1223
plicsw_ipi, 1224
plicsw_ipi_clear, 1223
plicsw_ipi_send, 1224
andes_plicsw.h
plicsw_cold_ipi_init, 686
PLICSW_CONTEXT_BASE, 686
PLICSW_CONTEXT_CLAIM, 686
PLICSW_CONTEXT_STRIDE, 686
PLICSW_ENABLE_BASE, 686
PLICSW_ENABLE_STRIDE, 686
PLICSW_PENDING_BASE, 686
PLICSW_PRIORITY_BASE, 686
PLICSW_REGION_ALIGN, 686
andes_plmt.c
plmt, 1603
plmt_cold_timer_init, 1602
plmt_timer, 1603
plmt_timer_event_start, 1602
plmt_timer_event_stop, 1602
plmt_timer_value, 1603
plmt_warm_timer_init, 1603
andes_plmt.h
DEFAULT_AE350_PLMT_FREQ, 772
plmt_cold_timer_init, 773
PLMT_REGION_ALIGN, 772
ANSI_COLOR_GREEN
sbi_unit_test.c, 1126
ANSI_COLOR_RED
sbi_unit_test.c, 1126
ANSI_COLOR_RESET
sbi_unit_test.c, 1127
aplic.c
aplic_check_msicfg, 1235
APLIC_CLRIE_BASE, 1230
APLIC_CLRIENUM, 1230
APLIC_CLRIP_BASE, 1230
APLIC_CLRIPNUM, 1230
aplic_cold_irqchip_init, 1235
APLIC_DEFAULT_PRIORITY, 1230
APLIC_DISABLE_IDELIVERY, 1230
APLIC_DISABLE_ITHRESHOLD, 1230
APLIC_DOMAINCFG, 1230
APLIC_DOMAINCFG_BE, 1230
APLIC_DOMAINCFG_DM, 1231
APLIC_DOMAINCFG_IE, 1231
APLIC_ENABLE_IDELIVERY, 1231
APLIC_ENABLE_ITHRESHOLD, 1231
APLIC_IDC_BASE, 1231
APLIC_IDC_CLAIMI, 1231
APLIC_IDC_IDELIVERY, 1231
APLIC_IDC_IFORCE, 1231
APLIC_IDC_ITHRESHOLD, 1231
APLIC_IDC_SIZE, 1231
APLIC_IDC_TOPI, 1231
APLIC_IDC_TOPI_ID_MASK, 1231
APLIC_IDC_TOPI_ID_SHIFT, 1231
APLIC_IDC_TOPI_PRIO_MASK, 1231
APLIC_MAX_IDC, 1231
APLIC_MAX_SOURCE, 1232
APLIC_MMSICFGADDR, 1232
APLIC_MMSICFGADDRH, 1232
APLIC_SETIE_BASE, 1232

APLIC_SETIENUM, 1232
APLIC_SETIP_BASE, 1232
APLIC_SETIPNUM, 1232
APLIC_SETIPNUM_BE, 1232
APLIC_SETIPNUM_LE, 1232
APLIC_SMSICFGADDR, 1232
APLIC_SMSICFGADDRH, 1232
APLIC_SOURCECFG_BASE, 1232
APLIC_SOURCECFG_CHILDIDX_MASK, 1232
APLIC_SOURCECFG_D, 1232
APLIC_SOURCECFG_SM_DETACH, 1232
APLIC_SOURCECFG_SM_EDGE_FALL, 1233
APLIC_SOURCECFG_SM_EDGE_RISE, 1233
APLIC_SOURCECFG_SM_INACTIVE, 1233
APLIC_SOURCECFG_SM_LEVEL_HIGH, 1233
APLIC_SOURCECFG_SM_LEVEL_LOW, 1233
APLIC_SOURCECFG_SM_MASK, 1233
APLIC_TARGET_BASE, 1233
APLIC_TARGET_EIID_MASK, 1233
APLIC_TARGET_GUEST_IDX_MASK, 1233
APLIC_TARGET_GUEST_IDX_SHIFT, 1233
APLIC_TARGET_HART_IDX_MASK, 1233
APLIC_TARGET_HART_IDX_SHIFT, 1233
APLIC_TARGET_IPRIO_MASK, 1233
aplic_writel_msicfg, 1236
APLIC_xMSICFGADDR_PPN_HART, 1233
APLIC_xMSICFGADDR_PPN_HHX, 1233
APLIC_xMSICFGADDR_PPN_HHX_MASK, 1234
APLIC_xMSICFGADDR_PPN_HHX_SHIFT, 1234
APLIC_xMSICFGADDR_PPN_LHX, 1234
APLIC_xMSICFGADDR_PPN_LHX_MASK, 1234
APLIC_xMSICFGADDR_PPN_LHX_SHIFT, 1234
APLIC_xMSICFGADDR_PPN_SHIFT, 1234
APLIC_xMSICFGADDRH_BAPPN_MASK, 1234
APLIC_xMSICFGADDRH_HHXS_MASK, 1234
APLIC_xMSICFGADDRH_HHXS_SHIFT, 1234
APLIC_xMSICFGADDRH_HHXW_MASK, 1234
APLIC_xMSICFGADDRH_HHXW_SHIFT, 1234
APLIC_xMSICFGADDRH_L, 1234
APLIC_xMSICFGADDRH_LHXS_MASK, 1235
APLIC_xMSICFGADDRH_LHXS_SHIFT, 1235
APLIC_xMSICFGADDRH_LHXW_MASK, 1235
APLIC_xMSICFGADDRH_LHXW_SHIFT, 1235
aplic.h
aplic_cold_irqchip_init, 690
APLIC_MAX_DELEGATE, 690
aplic_check_msicfg
aplic.c, 1235
APLIC_CLRIE_BASE
aplic.c, 1230
APLIC_CLRIENUM
aplic.c, 1230
APLIC_CLRIP_BASE
aplic.c, 1230
APLIC_CLRIPNUM
aplic.c, 1230
aplic_cold_irqchip_init
aplic.c, 1235
aplic.h, 690
aplic_data, 79
addr, 79
delegate, 79
has_msicfg_mmode, 79
has_msicfg_smode, 79
irqchip, 80
msicfg_mmode, 80
msicfg_smode, 80
num_idc, 80
num_source, 80
size, 80
targets_mmode, 80
APLIC_DEFAULT_PRIORITY
aplic.c, 1230
aplic_delegate_data, 80
child_index, 80
first_irq, 80
last_irq, 80
APLIC_DISABLE_IDELIVERY
aplic.c, 1230
APLIC_DISABLE_ITHRESHOLD
aplic.c, 1230
APLIC_DOMAINCFG
aplic.c, 1230
APLIC_DOMAINCFG_BE
aplic.c, 1230
APLIC_DOMAINCFG_DM
aplic.c, 1231
APLIC_DOMAINCFG_IE
aplic.c, 1231
APLIC_ENABLE_IDELIVERY
aplic.c, 1231
APLIC_ENABLE_ITHRESHOLD
aplic.c, 1231
APLIC_IDC_BASE
aplic.c, 1231
APLIC_IDC_CLAIMI
aplic.c, 1231
APLIC_IDC_IDELIVERY
aplic.c, 1231
APLIC_IDC_IFORCE
aplic.c, 1231
APLIC_IDC_ITHRESHOLD
aplic.c, 1231
APLIC_IDC_SIZE
aplic.c, 1231
APLIC_IDC_TOPI
aplic.c, 1231
APLIC_IDC_TOPI_ID_MASK
aplic.c, 1231
APLIC_IDC_TOPI_ID_SHIFT
aplic.c, 1231
APLIC_IDC_TOPI_PRIO_MASK
aplic.c, 1231
APLIC_MAX_DELEGATE
aplic.h, 690
APLIC_MAX_IDC

aplic.c, 1231
APLIC_MAX_SOURCE
aplic.c, 1232
APLIC_MMSICFGADDR
aplic.c, 1232
APLIC_MMSICFGADDRH
aplic.c, 1232
aplic_msicfg_data, 81
base_addr, 81
hhxs, 81
hhxw, 81
lhxs, 81
lhxw, 81
APLIC_SETIE_BASE
aplic.c, 1232
APLIC_SETIENUM
aplic.c, 1232
APLIC_SETIP_BASE
aplic.c, 1232
APLIC_SETIPNUM
aplic.c, 1232
APLIC_SETIPNUM_BE
aplic.c, 1232
APLIC_SETIPNUM_LE
aplic.c, 1232
APLIC_SMSICFGADDR
aplic.c, 1232
APLIC_SMSICFGADDRH
aplic.c, 1232
APLIC_SOURCECFG_BASE
aplic.c, 1232
APLIC_SOURCECFG_CHILDIDX_MASK
aplic.c, 1232
APLIC_SOURCECFG_D
aplic.c, 1232
APLIC_SOURCECFG_SM_DETACH
aplic.c, 1232
APLIC_SOURCECFG_SM_EDGE_FALL
aplic.c, 1233
APLIC_SOURCECFG_SM_EDGE_RISE
aplic.c, 1233
APLIC_SOURCECFG_SM_INACTIVE
aplic.c, 1233
APLIC_SOURCECFG_SM_LEVEL_HIGH
aplic.c, 1233
APLIC_SOURCECFG_SM_LEVEL_LOW
aplic.c, 1233
APLIC_SOURCECFG_SM_MASK
aplic.c, 1233
APLIC_TARGET_BASE
aplic.c, 1233
APLIC_TARGET_EIID_MASK
aplic.c, 1233
APLIC_TARGET_GUEST_IDX_MASK
aplic.c, 1233
APLIC_TARGET_GUEST_IDX_SHIFT
aplic.c, 1233
APLIC_TARGET_HART_IDX_MASK
aplic.c, 1233
APLIC_TARGET_HART_IDX_SHIFT
aplic.c, 1233
APLIC_TARGET_IPRIO_MASK
aplic.c, 1233
aplic_writel_msicfg
aplic.c, 1236
APLIC_xMSICFGADDR_PPN_HART
aplic.c, 1233
APLIC_xMSICFGADDR_PPN_HHX
aplic.c, 1233
APLIC_xMSICFGADDR_PPN_HHX_MASK
aplic.c, 1234
APLIC_xMSICFGADDR_PPN_HHX_SHIFT
aplic.c, 1234
APLIC_xMSICFGADDR_PPN_LHX
aplic.c, 1234
APLIC_xMSICFGADDR_PPN_LHX_MASK
aplic.c, 1234
APLIC_xMSICFGADDR_PPN_LHX_SHIFT
aplic.c, 1234
APLIC_xMSICFGADDR_PPN_SHIFT
aplic.c, 1234
APLIC_xMSICFGADDRH_BAPPN_MASK
aplic.c, 1234
APLIC_xMSICFGADDRH_HHXS_MASK
aplic.c, 1234
APLIC_xMSICFGADDRH_HHXS_SHIFT
aplic.c, 1234
APLIC_xMSICFGADDRH_HHXW_MASK
aplic.c, 1234
APLIC_xMSICFGADDRH_HHXW_SHIFT
aplic.c, 1234
APLIC_xMSICFGADDRH_L
aplic.c, 1234
APLIC_xMSICFGADDRH_LHXS_MASK
aplic.c, 1235
APLIC_xMSICFGADDRH_LHXS_SHIFT
aplic.c, 1235
APLIC_xMSICFGADDRH_LHXW_MASK
aplic.c, 1235
APLIC_xMSICFGADDRH_LHXW_SHIFT
aplic.c, 1235
arg1
sbi_hart.h, 439
args
fdt_phandle_args, 86
mbox_xfer, 96
args_count
fdt_phandle_args, 86
Ariane FPGA SoC Platform, 41
array_size
sbi_types.h, 619
asid
sbi_tlb_info, 153
assert_field_offset
fdt_mpxy_rpmi_mbox.c, 1485, 1486
sbi_mpxy.c, 996, 998, 999

sbi_sse.c, 1034–1036
assigned_harts
sbi_domain, 128
assigned_harts_lock
sbi_domain, 128
ASSUME_LATEST
libfdt_internal.h, 1436
ASSUME_LIBFDT_FLAWLESS
libfdt_internal.h, 1436
ASSUME_LIBFDT_ORDER
libfdt_internal.h, 1436
ASSUME_NO_ROLLBACK
libfdt_internal.h, 1436
ASSUME_PERFECT
libfdt_internal.h, 1435
ASSUME_VALID_DTB
libfdt_internal.h, 1436
ASSUME_VALID_INPUT
libfdt_internal.h, 1436
atcsmu.c
smu_set_command, 1587
smu_set_reset_vector, 1587
smu_set_wakeup_events, 1588
smu_support_sleep_mode, 1588
atcsmu.h
DEEP_SLEEP_CMD, 762
DEEPSLEEP_MODE, 762
FLASH_BASE, 762
HARTn_RESET_VEC_HI, 762
HARTn_RESET_VEC_LO, 762
LIGHT_SLEEP_CMD, 763
LIGHTSLEEP_MODE, 763
PCS0_CFG_OFFSET, 763
PCS0_CTL_OFFSET, 763
PCS0_WE_OFFSET, 763
PCS_CFG_DEEP_SLEEP, 763
PCS_CFG_DEEP_SLEEP_SHIFT, 763
PCS_CFG_LIGHT_SLEEP, 763
PCS_CFG_LIGHT_SLEEP_SHIFT, 763
PCS_CTL_CMD_SHIFT, 763
PCS_CTL_PARAM_SHIFT, 763
PCS_MAX_NR, 763
PCSm_CFG_OFFSET, 763
PCSm_CTL_OFFSET, 763
PCSm_WE_OFFSET, 764
RESET_VEC_8CORE_OFFSET, 764
RESET_VEC_HI_OFFSET, 764
RESET_VEC_LO_OFFSET, 764
SLEEP_CMD, 764
smu_set_command, 764
smu_set_reset_vector, 764
smu_set_wakeup_events, 765
smu_support_sleep_mode, 765
WAKEUP_CMD, 764
atcwdt200_reset
fdt_reset_atcwdt200.c, 1509
atcwdt200_reset_init
fdt_reset_atcwdt200.c, 1509
atcwdt200_reset_match
fdt_reset_atcwdt200.c, 1509
ATCWDT200_WP_NUM
fdt_reset_atcwdt200.c, 1507
atomic_add_return
riscv_atomic.c, 786
riscv_atomic.h, 179
atomic_clear_bit
riscv_atomic.c, 786
riscv_atomic.h, 179
atomic_clear_bit_test
riscv_atomic_test.c, 1112
atomic_cmpxchg
riscv_atomic.c, 787
riscv_atomic.h, 180
ATOMIC_INIT
riscv_atomic.h, 179
ATOMIC_INITIALIZER
riscv_atomic.h, 179
atomic_raw_clear_bit
riscv_atomic.c, 787
riscv_atomic.h, 180
atomic_raw_clear_bit_test
riscv_atomic_test.c, 1112
atomic_raw_set_bit
riscv_atomic.c, 788
riscv_atomic.h, 181
atomic_raw_set_bit_test
riscv_atomic_test.c, 1112
atomic_raw_xchg_uint
riscv_atomic.c, 788
riscv_atomic.h, 181
atomic_raw_xchg_ulong
riscv_atomic.c, 788
riscv_atomic.h, 181
atomic_read
riscv_atomic.c, 788
riscv_atomic.h, 181
atomic_rw_test
riscv_atomic_test.c, 1113
atomic_set_bit
riscv_atomic.c, 789
riscv_atomic.h, 182
atomic_set_bit_test
riscv_atomic_test.c, 1113
atomic_sub_return
riscv_atomic.c, 789
riscv_atomic.h, 182
atomic_t, 81
counter, 81
ATOMIC_TEST_BIT_NUM
riscv_atomic_test.c, 1111
atomic_test_cases
riscv_atomic_test.c, 1115
ATOMIC_TEST_RAW_BIT_CELL
riscv_atomic_test.c, 1111
ATOMIC_TEST_RAW_BIT_NUM
riscv_atomic_test.c, 1111

atomic_test_suite
riscv_atomic_test.c, 1115
atomic_test_suite_init
riscv_atomic_test.c, 1113
ATOMIC_TEST_VAL1
riscv_atomic_test.c, 1111
ATOMIC_TEST_VAL2
riscv_atomic_test.c, 1111
ATOMIC_TEST_VAL3
riscv_atomic_test.c, 1111
atomic_write
riscv_atomic.c, 790
riscv_atomic.h, 183
atomic_xchg
riscv_atomic.c, 790
riscv_atomic.h, 183
atomic_xchg_test
riscv_atomic_test.c, 1113
attr_id2index
fdt_mpxy_rpmi_mbox.c, 1485
ATTR_SIZE
sbi_mpxy.c, 996
attrs
sbi_mpxy_channel, 139
avail
sbi_fifo, 133
available_trigs
sbi_dbtr_hart_triggers_state, 125
axchg
riscv_atomic.c, 786

B
qdivrem.c, 1451
BANNER
sbi_init.c, 971
base
sbi_domain_memregion, 131
base_addr
aplic_msicfg_data, 81
baud
platform_uart_data, 97
be16_t
sbi_types.h, 621
be16_to_cpu
sbi_byteorder.h, 323
be32_t
sbi_types.h, 621
be32_to_cpu
sbi_byteorder.h, 323
be64_t
sbi_types.h, 621
be64_to_cpu
sbi_byteorder.h, 324
BIT
sbi_bitops.h, 313
BIT_ALIGN
sbi_bitops.h, 313
BIT_MASK
sbi_bitops.h, 313
BIT_ULL
sbi_bitops.h, 314
BIT_WORD
sbi_bitops.h, 314
BIT_WORD_OFFSET
sbi_bitops.h, 314
bitmap_and
sbi_bitmap.h, 306
bitmap_and_test
sbi_bitmap_test.c, 1119
bitmap_clear
sbi_bitmap.h, 307
bitmap_copy
sbi_bitmap.h, 307
bitmap_estimate_size
sbi_bitmap.h, 308
bitmap_fill
sbi_bitmap.h, 308
BITMAP_FIRST_WORD_MASK
sbi_bitmap.h, 305
BITMAP_LAST_WORD_MASK
sbi_bitmap.h, 305
bitmap_or
sbi_bitmap.h, 308
bitmap_or_test
sbi_bitmap_test.c, 1119
bitmap_set
sbi_bitmap.h, 309
bitmap_test
sbi_bitmap.h, 309
bitmap_test_cases
sbi_bitmap_test.c, 1120
bitmap_xor
sbi_bitmap.h, 309
bitmap_xor_test
sbi_bitmap_test.c, 1119
bitmap_zero
sbi_bitmap.h, 310
bitmap_zero_except
sbi_bitmap.h, 310
BITOP_WORD
sbi_bitops.c, 798
BITS_PER_LONG
sbi_bitops.h, 314
BITS_PER_LONG_LONG
sbi_bitops.h, 314
BITS_TO_LONGS
sbi_bitops.h, 314
bool
sbi_types.h, 621
boot_cpuid_phys
fdt_header, 84
boot_hart
fw_dynamic_info, 89
boot_hartid
sbi_domain, 128
BSWAP16
sbi_byteorder.h, 324

BSWAP32
sbi_byteorder.h, 324
BSWAP64
sbi_byteorder.h, 324

cadence-uart.c
cadence_console, 1534
cadence_uart_getc, 1532
cadence_uart_init, 1532
cadence_uart_putc, 1532
get_reg, 1533
set_reg, 1533
uart_base, 1534
uart_baudrate, 1534
UART_BRGR_CD_CLKDIVISOR, 1530
UART_CSR_REMPTY, 1530
UART_CSR_TFUL, 1531
UART_CTRL_RXDIS, 1531
UART_CTRL_RXEN, 1531
UART_CTRL_RXRES, 1531
UART_CTRL_TXDIS, 1531
UART_CTRL_TXEN, 1531
UART_CTRL_TXRES, 1531
uart_in_freq, 1534
uart_min_clk_divisor, 1533
UART_MODE_PAR_NONE, 1531
UART_REG_BDIVR, 1531
UART_REG_BRGR, 1531
UART_REG_CSR, 1531
UART_REG_CTRL, 1531
UART_REG_IDR, 1531
UART_REG_MODE, 1531
UART_REG_RFIFO_TFIFO, 1531
cadence-uart.h
cadence_uart_init, 743
cadence_console
cadence-uart.c, 1534
cadence_uart_getc
cadence-uart.c, 1532
cadence_uart_init
cadence-uart.c, 1532
cadence-uart.h, 743
cadence_uart_putc
cadence-uart.c, 1532
can_assume
libfdt_internal.h, 1435
can_assume_
libfdt_internal.h, 1436
CAP_DISABLE
sbi_mpxy.c, 996
CAP_ENABLE
sbi_mpxy.c, 996
CAP_EVENTSSTATE_MASK
sbi_mpxy.c, 996
CAP_EVENTSSTATE_POS
sbi_mpxy.c, 996
CAP_GET
sbi_mpxy.c, 997
CAP_GET_NOTIFICATIONS_MASK
sbi_mpxy.c, 997
CAP_GET_NOTIFICATIONS_POS
sbi_mpxy.c, 997
CAP_MSI_MASK
sbi_mpxy.c, 997
CAP_MSI_POS
sbi_mpxy.c, 997
CAP_SEND_MSG_WITH_RESP_MASK
sbi_mpxy.c, 997
CAP_SEND_MSG_WITH_RESP_POS
sbi_mpxy.c, 997
CAP_SEND_MSG_WITHOUT_RESP_MASK
sbi_mpxy.c, 997
CAP_SEND_MSG_WITHOUT_RESP_POS
sbi_mpxy.c, 997
CAP_SSE_MASK
sbi_mpxy.c, 997
CAP_SSE_POS
sbi_mpxy.c, 997
capability
sbi_mpxy_channel_attrs, 140
cause
sbi_trap_info, 154
CAUSE_BREAKPOINT
riscv_encoding.h, 213
CAUSE_DOUBLE_TRAP
riscv_encoding.h, 213
CAUSE_FETCH_ACCESS
riscv_encoding.h, 214
CAUSE_FETCH_GUEST_PAGE_FAULT
riscv_encoding.h, 214
CAUSE_FETCH_PAGE_FAULT
riscv_encoding.h, 214
CAUSE_ILLEGAL_INSTRUCTION
riscv_encoding.h, 214
CAUSE_LOAD_ACCESS
riscv_encoding.h, 214
CAUSE_LOAD_GUEST_PAGE_FAULT
riscv_encoding.h, 214
CAUSE_LOAD_PAGE_FAULT
riscv_encoding.h, 214
CAUSE_MACHINE_ECALL
riscv_encoding.h, 214
CAUSE_MISALIGNED_FETCH
riscv_encoding.h, 214
CAUSE_MISALIGNED_LOAD
riscv_encoding.h, 214
CAUSE_MISALIGNED_STORE
riscv_encoding.h, 214
CAUSE_STORE_ACCESS
riscv_encoding.h, 214
CAUSE_STORE_GUEST_PAGE_FAULT
riscv_encoding.h, 214
CAUSE_STORE_PAGE_FAULT
riscv_encoding.h, 214
CAUSE_SUPERVISOR_ECALL
riscv_encoding.h, 214
CAUSE_SW_CHECK_EXCP

riscv_encoding.h, 215
CAUSE_USER_ECALL
riscv_encoding.h, 215
CAUSE_VIRTUAL_INST_FAULT
riscv_encoding.h, 215
CAUSE_VIRTUAL_SUPERVISOR_ECALL
riscv_encoding.h, 215
cdefs.h
__FBSDID, 1447
chan_args
mbox_chan, 94
chan_list
mbox_controller, 94
channel_id
sbi_mpxy_channel, 139
CHAR_BIT
limits.h, 1447
check_block_
fdt.c, 1257
check_off_
fdt.c, 1258
child_index
aplic_delegate_data, 80
chip
gpio_pin, 91
CLAMP
sbi_types.h, 619
CLEAR_DBTR_BIT
riscv_dbtr.h, 189
clear_region
sbi_domain.c, 839
clear_test_console_buf
sbi_console_test.c, 1122
CLINT_MSWI_OFFSET
aclint_mswi.h, 684
CLINT_MTIMER_OFFSET
aclint_mtimer.h, 770
clint_offset
fdt_ipi_mswi.c, 1226
CLK_PCLK
fdt_reset_atcwdt200.c, 1507
clock_data
fdt_mpxy_rpmi_mbox.c, 1490
clock_id
rpmi_clock_get_attributes_req, 102
rpmi_clock_get_config_req, 102
rpmi_clock_get_rate_req, 103
rpmi_clock_get_supported_rates_req, 104
rpmi_clock_set_config_req, 105
rpmi_clock_set_rate_req, 106
clock_rate
rpmi_clock_get_supported_rates_resp, 105
clock_rate_high
rpmi_clock_get_rate_resp, 104
rpmi_clock_set_rate_req, 106
clock_rate_index
rpmi_clock_get_supported_rates_req, 104
clock_rate_low
rpmi_clock_get_rate_resp, 104
rpmi_clock_set_rate_req, 106
clock_services
fdt_mpxy_rpmi_mbox.c, 1490
CMD_POWEROFF
fdt_reset_sg2042_hwmon_mcu.c, 1520
CMD_REBOOT
fdt_reset_sg2042_hwmon_mcu.c, 1520
CMD_RESET
fdt_reset_sg2042_hwmon_mcu.c, 1520
cmpxchg_test
riscv_atomic_test.c, 1114
cold_boot_allowed
sbi_platform_operations, 144
coldboot_done
sbi_init.c, 981
coldboot_lottery
sbi_init.c, 981
COMBINE
qdivrem.c, 1451
compatible
fdt_match, 85
complete_cb
sbi_sse_cb_ops, 150
config
rpmi_clock_get_config_resp, 103
rpmi_clock_set_config_req, 105
console_dev
sbi_console.c, 821
console_early_buffer
sbi_console.c, 821
CONSOLE_EARLY_BUFFER_SIZE
sbi_console.c, 801
console_getc
sbi_console_device, 123
console_out_lock
sbi_console.c, 821
console_putc
sbi_console_device, 123
console_puts
sbi_console_device, 123
console_tbuf
sbi_console.c, 821
console_tbuf_len
sbi_console.c, 821
CONSOLE_TBUF_MAX
sbi_console.c, 801
console_test_cases
sbi_console_test.c, 1124
container_of
sbi_types.h, 619
context_map
plic_data, 98
copy_attrs
sbi_sse.c, 1036
counter
atomic_t, 81
cppc_dev

sbi_cppc.c, 825
cppc_probe
sbi_cppc_device, 123
cppc_read
sbi_cppc_device, 123
cppc_write
sbi_cppc_device, 123
cpu_relax
riscv_barrier.h, 185
cpu_to_be16
sbi_byteorder.h, 324
cpu_to_be32
sbi_byteorder.h, 324
cpu_to_be64
sbi_byteorder.h, 324
cpu_to_fdt16
libfdt_env.h, 1430
cpu_to_fdt32
libfdt_env.h, 1430
cpu_to_fdt64
libfdt_env.h, 1431
cpu_to_le16
sbi_byteorder.h, 324
cpu_to_le32
sbi_byteorder.h, 324
cpu_to_le64
sbi_byteorder.h, 325
csr_clear
riscv_asm.h, 167
CSR_CYCLE
riscv_encoding.h, 215
CSR_CYCLEH
riscv_encoding.h, 215
CSR_DCSR
riscv_encoding.h, 215
CSR_DPC
riscv_encoding.h, 215
CSR_DSCRATCH0
riscv_encoding.h, 215
CSR_DSCRATCH1
riscv_encoding.h, 215
CSR_FCSR
riscv_encoding.h, 215
CSR_FFLAGS
riscv_encoding.h, 215
CSR_FRM
riscv_encoding.h, 215
CSR_HCOUNTEREN
riscv_encoding.h, 215
CSR_HEDELEG
riscv_encoding.h, 215
CSR_HENVCFG
riscv_encoding.h, 216
CSR_HENVCFGH
riscv_encoding.h, 216
CSR_HGATP
riscv_encoding.h, 216
CSR_HGEIE
riscv_encoding.h, 216
CSR_HGEIP
riscv_encoding.h, 216
CSR_HIDELEG
riscv_encoding.h, 216
CSR_HIDELEGH
riscv_encoding.h, 216
CSR_HIE
riscv_encoding.h, 216
CSR_HIP
riscv_encoding.h, 216
CSR_HPMCOUNTER10
riscv_encoding.h, 216
CSR_HPMCOUNTER10H
riscv_encoding.h, 216
CSR_HPMCOUNTER11
riscv_encoding.h, 216
CSR_HPMCOUNTER11H
riscv_encoding.h, 216
CSR_HPMCOUNTER12
riscv_encoding.h, 216
CSR_HPMCOUNTER12H
riscv_encoding.h, 216
CSR_HPMCOUNTER13
riscv_encoding.h, 217
CSR_HPMCOUNTER13H
riscv_encoding.h, 217
CSR_HPMCOUNTER14
riscv_encoding.h, 217
CSR_HPMCOUNTER14H
riscv_encoding.h, 217
CSR_HPMCOUNTER15
riscv_encoding.h, 217
CSR_HPMCOUNTER15H
riscv_encoding.h, 217
CSR_HPMCOUNTER16
riscv_encoding.h, 217
CSR_HPMCOUNTER16H
riscv_encoding.h, 217
CSR_HPMCOUNTER17
riscv_encoding.h, 217
CSR_HPMCOUNTER17H
riscv_encoding.h, 217
CSR_HPMCOUNTER18
riscv_encoding.h, 217
CSR_HPMCOUNTER18H
riscv_encoding.h, 217
CSR_HPMCOUNTER19
riscv_encoding.h, 217
CSR_HPMCOUNTER19H
riscv_encoding.h, 217
CSR_HPMCOUNTER20
riscv_encoding.h, 217
CSR_HPMCOUNTER20H
riscv_encoding.h, 218
CSR_HPMCOUNTER21
riscv_encoding.h, 218
CSR_HPMCOUNTER21H

riscv_encoding.h, 218
CSR_HPMCOUNTER22
riscv_encoding.h, 218
CSR_HPMCOUNTER22H
riscv_encoding.h, 218
CSR_HPMCOUNTER23
riscv_encoding.h, 218
CSR_HPMCOUNTER23H
riscv_encoding.h, 218
CSR_HPMCOUNTER24
riscv_encoding.h, 218
CSR_HPMCOUNTER24H
riscv_encoding.h, 218
CSR_HPMCOUNTER25
riscv_encoding.h, 218
CSR_HPMCOUNTER25H
riscv_encoding.h, 218
CSR_HPMCOUNTER26
riscv_encoding.h, 218
CSR_HPMCOUNTER26H
riscv_encoding.h, 218
CSR_HPMCOUNTER27
riscv_encoding.h, 218
CSR_HPMCOUNTER27H
riscv_encoding.h, 218
CSR_HPMCOUNTER28
riscv_encoding.h, 219
CSR_HPMCOUNTER28H
riscv_encoding.h, 219
CSR_HPMCOUNTER29
riscv_encoding.h, 219
CSR_HPMCOUNTER29H
riscv_encoding.h, 219
CSR_HPMCOUNTER3
riscv_encoding.h, 219
CSR_HPMCOUNTER30
riscv_encoding.h, 219
CSR_HPMCOUNTER30H
riscv_encoding.h, 219
CSR_HPMCOUNTER31
riscv_encoding.h, 219
CSR_HPMCOUNTER31H
riscv_encoding.h, 219
CSR_HPMCOUNTER3H
riscv_encoding.h, 219
CSR_HPMCOUNTER4
riscv_encoding.h, 219
CSR_HPMCOUNTER4H
riscv_encoding.h, 219
CSR_HPMCOUNTER5
riscv_encoding.h, 219
CSR_HPMCOUNTER5H
riscv_encoding.h, 219
CSR_HPMCOUNTER6
riscv_encoding.h, 219
CSR_HPMCOUNTER6H
riscv_encoding.h, 220
CSR_HPMCOUNTER7
riscv_encoding.h, 220
CSR_HPMCOUNTER7H
riscv_encoding.h, 220
CSR_HPMCOUNTER8
riscv_encoding.h, 220
CSR_HPMCOUNTER8H
riscv_encoding.h, 220
CSR_HPMCOUNTER9
riscv_encoding.h, 220
CSR_HPMCOUNTER9H
riscv_encoding.h, 220
CSR_HSTATEEN0
riscv_encoding.h, 220
CSR_HSTATEEN0H
riscv_encoding.h, 220
CSR_HSTATEEN1
riscv_encoding.h, 220
CSR_HSTATEEN1H
riscv_encoding.h, 220
CSR_HSTATEEN2
riscv_encoding.h, 220
CSR_HSTATEEN2H
riscv_encoding.h, 220
CSR_HSTATEEN3
riscv_encoding.h, 220
CSR_HSTATEEN3H
riscv_encoding.h, 220
CSR_HSTATUS
riscv_encoding.h, 221
CSR_HTIMEDELTA
riscv_encoding.h, 221
CSR_HTIMEDELTAH
riscv_encoding.h, 221
CSR_HTINST
riscv_encoding.h, 221
CSR_HTVAL
riscv_encoding.h, 221
CSR_HVICTL
riscv_encoding.h, 221
CSR_HVIEN
riscv_encoding.h, 221
CSR_HVIENH
riscv_encoding.h, 221
CSR_HVIP
riscv_encoding.h, 221
CSR_HVIPH
riscv_encoding.h, 221
CSR_HVIPRIO1
riscv_encoding.h, 221
CSR_HVIPRIO1H
riscv_encoding.h, 221
CSR_HVIPRIO2
riscv_encoding.h, 221
CSR_HVIPRIO2H
riscv_encoding.h, 221
CSR_INSTRET
riscv_encoding.h, 221
CSR_INSTRETH

riscv_encoding.h, 222
CSR_MARCHID
riscv_encoding.h, 222
CSR_MCAUSE
riscv_encoding.h, 222
CSR_MCONFIGPTR
riscv_encoding.h, 222
CSR_MCOUNTEREN
riscv_encoding.h, 222
CSR_MCOUNTINHIBIT
riscv_encoding.h, 222
CSR_MCYCLE
riscv_encoding.h, 222
CSR_MCYCLECFG
riscv_encoding.h, 222
CSR_MCYCLECFGH
riscv_encoding.h, 222
CSR_MCYCLEH
riscv_encoding.h, 222
CSR_MEDELEG
riscv_encoding.h, 222
CSR_MENVCFG
riscv_encoding.h, 222
CSR_MENVCFGH
riscv_encoding.h, 222
CSR_MEPC
riscv_encoding.h, 222
CSR_MHARTID
riscv_encoding.h, 222
CSR_MHPMCOUNTER10
riscv_encoding.h, 223
CSR_MHPMCOUNTER10H
riscv_encoding.h, 223
CSR_MHPMCOUNTER11
riscv_encoding.h, 223
CSR_MHPMCOUNTER11H
riscv_encoding.h, 223
CSR_MHPMCOUNTER12
riscv_encoding.h, 223
CSR_MHPMCOUNTER12H
riscv_encoding.h, 223
CSR_MHPMCOUNTER13
riscv_encoding.h, 223
CSR_MHPMCOUNTER13H
riscv_encoding.h, 223
CSR_MHPMCOUNTER14
riscv_encoding.h, 223
CSR_MHPMCOUNTER14H
riscv_encoding.h, 223
CSR_MHPMCOUNTER15
riscv_encoding.h, 223
CSR_MHPMCOUNTER15H
riscv_encoding.h, 223
CSR_MHPMCOUNTER16
riscv_encoding.h, 223
CSR_MHPMCOUNTER16H
riscv_encoding.h, 223
CSR_MHPMCOUNTER17
riscv_encoding.h, 223
CSR_MHPMCOUNTER17H
riscv_encoding.h, 224
CSR_MHPMCOUNTER18
riscv_encoding.h, 224
CSR_MHPMCOUNTER18H
riscv_encoding.h, 224
CSR_MHPMCOUNTER19
riscv_encoding.h, 224
CSR_MHPMCOUNTER19H
riscv_encoding.h, 224
CSR_MHPMCOUNTER20
riscv_encoding.h, 224
CSR_MHPMCOUNTER20H
riscv_encoding.h, 224
CSR_MHPMCOUNTER21
riscv_encoding.h, 224
CSR_MHPMCOUNTER21H
riscv_encoding.h, 224
CSR_MHPMCOUNTER22
riscv_encoding.h, 224
CSR_MHPMCOUNTER22H
riscv_encoding.h, 224
CSR_MHPMCOUNTER23
riscv_encoding.h, 224
CSR_MHPMCOUNTER23H
riscv_encoding.h, 224
CSR_MHPMCOUNTER24
riscv_encoding.h, 224
CSR_MHPMCOUNTER24H
riscv_encoding.h, 224
CSR_MHPMCOUNTER25
riscv_encoding.h, 225
CSR_MHPMCOUNTER25H
riscv_encoding.h, 225
CSR_MHPMCOUNTER26
riscv_encoding.h, 225
CSR_MHPMCOUNTER26H
riscv_encoding.h, 225
CSR_MHPMCOUNTER27
riscv_encoding.h, 225
CSR_MHPMCOUNTER27H
riscv_encoding.h, 225
CSR_MHPMCOUNTER28
riscv_encoding.h, 225
CSR_MHPMCOUNTER28H
riscv_encoding.h, 225
CSR_MHPMCOUNTER29
riscv_encoding.h, 225
CSR_MHPMCOUNTER29H
riscv_encoding.h, 225
CSR_MHPMCOUNTER3
riscv_encoding.h, 225
CSR_MHPMCOUNTER30
riscv_encoding.h, 225
CSR_MHPMCOUNTER30H
riscv_encoding.h, 225
CSR_MHPMCOUNTER31

riscv_encoding.h, 225
CSR_MHPMCOUNTER31H
riscv_encoding.h, 225
CSR_MHPMCOUNTER3H
riscv_encoding.h, 226
CSR_MHPMCOUNTER4
riscv_encoding.h, 226
CSR_MHPMCOUNTER4H
riscv_encoding.h, 226
CSR_MHPMCOUNTER5
riscv_encoding.h, 226
CSR_MHPMCOUNTER5H
riscv_encoding.h, 226
CSR_MHPMCOUNTER6
riscv_encoding.h, 226
CSR_MHPMCOUNTER6H
riscv_encoding.h, 226
CSR_MHPMCOUNTER7
riscv_encoding.h, 226
CSR_MHPMCOUNTER7H
riscv_encoding.h, 226
CSR_MHPMCOUNTER8
riscv_encoding.h, 226
CSR_MHPMCOUNTER8H
riscv_encoding.h, 226
CSR_MHPMCOUNTER9
riscv_encoding.h, 226
CSR_MHPMCOUNTER9H
riscv_encoding.h, 226
CSR_MHPMEVENT10
riscv_encoding.h, 226
CSR_MHPMEVENT10H
riscv_encoding.h, 226
CSR_MHPMEVENT11
riscv_encoding.h, 227
CSR_MHPMEVENT11H
riscv_encoding.h, 227
CSR_MHPMEVENT12
riscv_encoding.h, 227
CSR_MHPMEVENT12H
riscv_encoding.h, 227
CSR_MHPMEVENT13
riscv_encoding.h, 227
CSR_MHPMEVENT13H
riscv_encoding.h, 227
CSR_MHPMEVENT14
riscv_encoding.h, 227
CSR_MHPMEVENT14H
riscv_encoding.h, 227
CSR_MHPMEVENT15
riscv_encoding.h, 227
CSR_MHPMEVENT15H
riscv_encoding.h, 227
CSR_MHPMEVENT16
riscv_encoding.h, 227
CSR_MHPMEVENT16H
riscv_encoding.h, 227
CSR_MHPMEVENT17
riscv_encoding.h, 227
CSR_MHPMEVENT17H
riscv_encoding.h, 227
CSR_MHPMEVENT18
riscv_encoding.h, 227
CSR_MHPMEVENT18H
riscv_encoding.h, 228
CSR_MHPMEVENT19
riscv_encoding.h, 228
CSR_MHPMEVENT19H
riscv_encoding.h, 228
CSR_MHPMEVENT20
riscv_encoding.h, 228
CSR_MHPMEVENT20H
riscv_encoding.h, 228
CSR_MHPMEVENT21
riscv_encoding.h, 228
CSR_MHPMEVENT21H
riscv_encoding.h, 228
CSR_MHPMEVENT22
riscv_encoding.h, 228
CSR_MHPMEVENT22H
riscv_encoding.h, 228
CSR_MHPMEVENT23
riscv_encoding.h, 228
CSR_MHPMEVENT23H
riscv_encoding.h, 228
CSR_MHPMEVENT24
riscv_encoding.h, 228
CSR_MHPMEVENT24H
riscv_encoding.h, 228
CSR_MHPMEVENT25
riscv_encoding.h, 228
CSR_MHPMEVENT25H
riscv_encoding.h, 228
CSR_MHPMEVENT26
riscv_encoding.h, 229
CSR_MHPMEVENT26H
riscv_encoding.h, 229
CSR_MHPMEVENT27
riscv_encoding.h, 229
CSR_MHPMEVENT27H
riscv_encoding.h, 229
CSR_MHPMEVENT28
riscv_encoding.h, 229
CSR_MHPMEVENT28H
riscv_encoding.h, 229
CSR_MHPMEVENT29
riscv_encoding.h, 229
CSR_MHPMEVENT29H
riscv_encoding.h, 229
CSR_MHPMEVENT3
riscv_encoding.h, 229
CSR_MHPMEVENT30
riscv_encoding.h, 229
CSR_MHPMEVENT30H
riscv_encoding.h, 229
CSR_MHPMEVENT31

riscv_encoding.h, 229
CSR_MHPMEVENT31H
riscv_encoding.h, 229
CSR_MHPMEVENT3H
riscv_encoding.h, 229
CSR_MHPMEVENT4
riscv_encoding.h, 229
CSR_MHPMEVENT4H
riscv_encoding.h, 230
CSR_MHPMEVENT5
riscv_encoding.h, 230
CSR_MHPMEVENT5H
riscv_encoding.h, 230
CSR_MHPMEVENT6
riscv_encoding.h, 230
CSR_MHPMEVENT6H
riscv_encoding.h, 230
CSR_MHPMEVENT7
riscv_encoding.h, 230
CSR_MHPMEVENT7H
riscv_encoding.h, 230
CSR_MHPMEVENT8
riscv_encoding.h, 230
CSR_MHPMEVENT8H
riscv_encoding.h, 230
CSR_MHPMEVENT9
riscv_encoding.h, 230
CSR_MHPMEVENT9H
riscv_encoding.h, 230
CSR_MIDELEG
riscv_encoding.h, 230
CSR_MIDELEGH
riscv_encoding.h, 230
CSR_MIE
riscv_encoding.h, 230
CSR_MIEH
riscv_encoding.h, 230
CSR_MIMPID
riscv_encoding.h, 231
CSR_MINSTRET
riscv_encoding.h, 231
CSR_MINSTRETCFG
riscv_encoding.h, 231
CSR_MINSTRETCFGH
riscv_encoding.h, 231
CSR_MINSTRETH
riscv_encoding.h, 231
CSR_MIP
riscv_encoding.h, 231
CSR_MIPH
riscv_encoding.h, 231
CSR_MIREG
riscv_encoding.h, 231
CSR_MIREG2
riscv_encoding.h, 231
CSR_MIREG3
riscv_encoding.h, 231
CSR_MIREG4
riscv_encoding.h, 231
CSR_MIREG5
riscv_encoding.h, 231
CSR_MIREG6
riscv_encoding.h, 231
CSR_MISA
riscv_encoding.h, 231
CSR_MISELECT
riscv_encoding.h, 231
CSR_MSCRATCH
riscv_encoding.h, 232
CSR_MSECCFG
riscv_encoding.h, 232
CSR_MSECCFGH
riscv_encoding.h, 232
CSR_MSTATEEN0
riscv_encoding.h, 232
CSR_MSTATEEN0H
riscv_encoding.h, 232
CSR_MSTATEEN1
riscv_encoding.h, 232
CSR_MSTATEEN1H
riscv_encoding.h, 232
CSR_MSTATEEN2
riscv_encoding.h, 232
CSR_MSTATEEN2H
riscv_encoding.h, 232
CSR_MSTATEEN3
riscv_encoding.h, 232
CSR_MSTATEEN3H
riscv_encoding.h, 232
CSR_MSTATUS
riscv_encoding.h, 232
CSR_MSTATUSH
riscv_encoding.h, 232
CSR_MTINST
riscv_encoding.h, 232
CSR_MTOPEI
riscv_encoding.h, 232
CSR_MTOPI
riscv_encoding.h, 233
CSR_MTVAL
riscv_encoding.h, 233
CSR_MTVAL2
riscv_encoding.h, 233
CSR_MTVEC
riscv_encoding.h, 233
CSR_MVENDORID
riscv_encoding.h, 233
CSR_MVIEN
riscv_encoding.h, 233
CSR_MVIENH
riscv_encoding.h, 233
CSR_MVIP
riscv_encoding.h, 233
CSR_MVIPH
riscv_encoding.h, 233
CSR_PMPADDR0

riscv_encoding.h, 233
CSR_PMPADDR1
riscv_encoding.h, 233
CSR_PMPADDR10
riscv_encoding.h, 233
CSR_PMPADDR11
riscv_encoding.h, 233
CSR_PMPADDR12
riscv_encoding.h, 233
CSR_PMPADDR13
riscv_encoding.h, 233
CSR_PMPADDR14
riscv_encoding.h, 234
CSR_PMPADDR15
riscv_encoding.h, 234
CSR_PMPADDR16
riscv_encoding.h, 234
CSR_PMPADDR17
riscv_encoding.h, 234
CSR_PMPADDR18
riscv_encoding.h, 234
CSR_PMPADDR19
riscv_encoding.h, 234
CSR_PMPADDR2
riscv_encoding.h, 234
CSR_PMPADDR20
riscv_encoding.h, 234
CSR_PMPADDR21
riscv_encoding.h, 234
CSR_PMPADDR22
riscv_encoding.h, 234
CSR_PMPADDR23
riscv_encoding.h, 234
CSR_PMPADDR24
riscv_encoding.h, 234
CSR_PMPADDR25
riscv_encoding.h, 234
CSR_PMPADDR26
riscv_encoding.h, 234
CSR_PMPADDR27
riscv_encoding.h, 234
CSR_PMPADDR28
riscv_encoding.h, 235
CSR_PMPADDR29
riscv_encoding.h, 235
CSR_PMPADDR3
riscv_encoding.h, 235
CSR_PMPADDR30
riscv_encoding.h, 235
CSR_PMPADDR31
riscv_encoding.h, 235
CSR_PMPADDR32
riscv_encoding.h, 235
CSR_PMPADDR33
riscv_encoding.h, 235
CSR_PMPADDR34
riscv_encoding.h, 235
CSR_PMPADDR35
riscv_encoding.h, 235
CSR_PMPADDR36
riscv_encoding.h, 235
CSR_PMPADDR37
riscv_encoding.h, 235
CSR_PMPADDR38
riscv_encoding.h, 235
CSR_PMPADDR39
riscv_encoding.h, 235
CSR_PMPADDR4
riscv_encoding.h, 235
CSR_PMPADDR40
riscv_encoding.h, 235
CSR_PMPADDR41
riscv_encoding.h, 236
CSR_PMPADDR42
riscv_encoding.h, 236
CSR_PMPADDR43
riscv_encoding.h, 236
CSR_PMPADDR44
riscv_encoding.h, 236
CSR_PMPADDR45
riscv_encoding.h, 236
CSR_PMPADDR46
riscv_encoding.h, 236
CSR_PMPADDR47
riscv_encoding.h, 236
CSR_PMPADDR48
riscv_encoding.h, 236
CSR_PMPADDR49
riscv_encoding.h, 236
CSR_PMPADDR5
riscv_encoding.h, 236
CSR_PMPADDR50
riscv_encoding.h, 236
CSR_PMPADDR51
riscv_encoding.h, 236
CSR_PMPADDR52
riscv_encoding.h, 236
CSR_PMPADDR53
riscv_encoding.h, 236
CSR_PMPADDR54
riscv_encoding.h, 236
CSR_PMPADDR55
riscv_encoding.h, 237
CSR_PMPADDR56
riscv_encoding.h, 237
CSR_PMPADDR57
riscv_encoding.h, 237
CSR_PMPADDR58
riscv_encoding.h, 237
CSR_PMPADDR59
riscv_encoding.h, 237
CSR_PMPADDR6
riscv_encoding.h, 237
CSR_PMPADDR60
riscv_encoding.h, 237
CSR_PMPADDR61

riscv_encoding.h, 237
CSR_PMPADDR62
riscv_encoding.h, 237
CSR_PMPADDR63
riscv_encoding.h, 237
CSR_PMPADDR7
riscv_encoding.h, 237
CSR_PMPADDR8
riscv_encoding.h, 237
CSR_PMPADDR9
riscv_encoding.h, 237
CSR_PMPCFG0
riscv_encoding.h, 237
CSR_PMPCFG1
riscv_encoding.h, 237
CSR_PMPCFG10
riscv_encoding.h, 238
CSR_PMPCFG11
riscv_encoding.h, 238
CSR_PMPCFG12
riscv_encoding.h, 238
CSR_PMPCFG13
riscv_encoding.h, 238
CSR_PMPCFG14
riscv_encoding.h, 238
CSR_PMPCFG15
riscv_encoding.h, 238
CSR_PMPCFG2
riscv_encoding.h, 238
CSR_PMPCFG3
riscv_encoding.h, 238
CSR_PMPCFG4
riscv_encoding.h, 238
CSR_PMPCFG5
riscv_encoding.h, 238
CSR_PMPCFG6
riscv_encoding.h, 238
CSR_PMPCFG7
riscv_encoding.h, 238
CSR_PMPCFG8
riscv_encoding.h, 238
CSR_PMPCFG9
riscv_encoding.h, 238
csr_read
riscv_asm.h, 167
csr_read_allowed
sbi_csr_detect.h, 340
csr_read_clear
riscv_asm.h, 167
csr_read_num
riscv_asm.c, 778
riscv_asm.h, 169
csr_read_relaxed
riscv_asm.h, 167
csr_read_set
riscv_asm.h, 168
CSR_SATP
riscv_encoding.h, 238
CSR_SCAUSE
riscv_encoding.h, 239
CSR_SCOUNTEREN
riscv_encoding.h, 239
CSR_SCOUNTINHIBIT
riscv_encoding.h, 239
CSR_SCOUNTOVF
riscv_encoding.h, 239
CSR_SENVCFG
riscv_encoding.h, 239
CSR_SEPC
riscv_encoding.h, 239
csr_set
riscv_asm.h, 168
CSR_SIE
riscv_encoding.h, 239
CSR_SIEH
riscv_encoding.h, 239
CSR_SIP
riscv_encoding.h, 239
CSR_SIPH
riscv_encoding.h, 239
CSR_SIREG
riscv_encoding.h, 239
CSR_SIREG2
riscv_encoding.h, 239
CSR_SIREG3
riscv_encoding.h, 239
CSR_SIREG4
riscv_encoding.h, 239
CSR_SIREG5
riscv_encoding.h, 239
CSR_SIREG6
riscv_encoding.h, 240
CSR_SISELECT
riscv_encoding.h, 240
CSR_SSCRATCH
riscv_encoding.h, 240
CSR_SSP
riscv_encoding.h, 240
CSR_SSTATEEN0
riscv_encoding.h, 240
CSR_SSTATEEN1
riscv_encoding.h, 240
CSR_SSTATEEN2
riscv_encoding.h, 240
CSR_SSTATEEN3
riscv_encoding.h, 240
CSR_SSTATUS
riscv_encoding.h, 240
CSR_STIMECMP
riscv_encoding.h, 240
CSR_STIMECMPH
riscv_encoding.h, 240
CSR_STOPEI
riscv_encoding.h, 240
CSR_STOPI
riscv_encoding.h, 240

CSR_STVAL
riscv_encoding.h, 240
CSR_STVEC
riscv_encoding.h, 240
csr_swap
riscv_asm.h, 168
CSR_TDATA1
riscv_encoding.h, 241
CSR_TDATA2
riscv_encoding.h, 241
CSR_TDATA3
riscv_encoding.h, 241
CSR_TIME
riscv_encoding.h, 241
CSR_TIMEH
riscv_encoding.h, 241
CSR_TINFO
riscv_encoding.h, 241
CSR_TSELECT
riscv_encoding.h, 241
CSR_UCAUSE
riscv_encoding.h, 241
CSR_UEPC
riscv_encoding.h, 241
CSR_UIE
riscv_encoding.h, 241
CSR_UIP
riscv_encoding.h, 241
CSR_USCRATCH
riscv_encoding.h, 241
CSR_USTATUS
riscv_encoding.h, 241
CSR_UTVAL
riscv_encoding.h, 241
CSR_UTVEC
riscv_encoding.h, 241
CSR_VL
riscv_encoding.h, 242
CSR_VLENB
riscv_encoding.h, 242
CSR_VSATP
riscv_encoding.h, 242
CSR_VSCAUSE
riscv_encoding.h, 242
CSR_VSEPC
riscv_encoding.h, 242
CSR_VSIE
riscv_encoding.h, 242
CSR_VSIEH
riscv_encoding.h, 242
CSR_VSIP
riscv_encoding.h, 242
CSR_VSIPH
riscv_encoding.h, 242
CSR_VSIREG
riscv_encoding.h, 242
CSR_VSIREG2
riscv_encoding.h, 242
CSR_VSIREG3
riscv_encoding.h, 242
CSR_VSIREG4
riscv_encoding.h, 242
CSR_VSIREG5
riscv_encoding.h, 242
CSR_VSIREG6
riscv_encoding.h, 242
CSR_VSISELECT
riscv_encoding.h, 243
CSR_VSSCRATCH
riscv_encoding.h, 243
CSR_VSSTATUS
riscv_encoding.h, 243
CSR_VSTART
riscv_encoding.h, 243
CSR_VSTOPEI
riscv_encoding.h, 243
CSR_VSTOPI
riscv_encoding.h, 243
CSR_VSTVAL
riscv_encoding.h, 243
CSR_VSTVEC
riscv_encoding.h, 243
CSR_VTYPE
riscv_encoding.h, 243
csr_write
riscv_asm.h, 168
csr_write_allowed
sbi_csr_detect.h, 341
csr_write_num
riscv_asm.c, 779
riscv_asm.h, 170
CSRRC
riscv_encoding.h, 243
CSRRCI
riscv_encoding.h, 243
CSRRS
riscv_encoding.h, 243
CSRRSI
riscv_encoding.h, 243
CSRRW
riscv_encoding.h, 243
CSRRWI
riscv_encoding.h, 243
ctr_map
fdt_pmu_hw_event_counter_map, 86
fdt_pmu_raw_event_counter_map, 87
CTRL_REG
fdt_reset_atcwdt200.c, 1507
ctz
riscv_asm.c, 779
current_hartid
riscv_asm.h, 168
current_hartindex
sbi_scratch.h, 556

data
fdt_match, 85

fdt_property, 87
rpmi_message, 118
sbi_dbtr_shmem_entry, 127
data_a
sbi_bitmap_test.c, 1120
data_b
sbi_bitmap_test.c, 1120
DATA_BIT_SIZE
sbi_bitmap_test.c, 1119
data_bytes
sbi_ldst_data, 138
data_cleanup
sbi_domain_data, 130
data_hi
rpmi_cppc_read_reg_resp, 111
rpmi_cppc_write_reg_req, 111
data_idx
sbi_domain_data, 130
data_lo
rpmi_cppc_read_reg_resp, 111
rpmi_cppc_write_reg_req, 111
data_priv
sbi_domain, 129
data_setup
sbi_domain_data, 130
DATA_SIZE
sbi_bitmap_test.c, 1119
data_size
sbi_domain_data, 130
data_u32
sbi_ldst_data, 138
data_u64
sbi_ldst_data, 138
data_ulong
sbi_ldst_data, 138
data_zero
sbi_bitmap_test.c, 1120
datalen
rpmi_message_header, 120
db_addr_hi
rpmi_cppc_get_fastchan_region_resp, 108
db_addr_lo
rpmi_cppc_get_fastchan_region_resp, 108
db_preservemask_hi
rpmi_cppc_get_fastchan_region_resp, 108
db_preservemask_lo
rpmi_cppc_get_fastchan_region_resp, 108
db_setmask_hi
rpmi_cppc_get_fastchan_region_resp, 108
db_setmask_lo
rpmi_cppc_get_fastchan_region_resp, 108
dbtr_get_hart_state_ptr
sbi_dbtr.c, 827
dbtr_set_hart_state_ptr
sbi_dbtr.c, 827
DBTR_SHMEM_MAKE_PHYS
sbi_dbtr.c, 827
dbtr_thishart_state_ptr
sbi_dbtr.c, 827
dbtr_trigger_clear
sbi_dbtr.c, 827
dbtr_trigger_disable
sbi_dbtr.c, 828
dbtr_trigger_enable
sbi_dbtr.c, 828
dbtr_trigger_setup
sbi_dbtr.c, 829
dbtr_trigger_supported
sbi_dbtr.c, 829
dbtr_trigger_valid
sbi_dbtr.c, 829
dcpriv
sbi_domain_context.c, 861
DECLARE_BITMAP
sbi_bitmap.h, 305
sbi_domain_data.c, 862
sbi_hartmask, 135
DECLARE_UNPRIVILEGED_LOAD_FUNCTION
sbi_unpriv.h, 626
DECLARE_UNPRIVILEGED_STORE_FUNCTION
sbi_unpriv.h, 626
DEEP_SLEEP_CMD
atcsmu.h, 762
DEEPSLEEP_MODE
atcsmu.h, 762
DEFAULT_AE350_PLMT_FREQ
andes_plmt.h, 772
default_irqfn
sbi_irqchip.c, 991
DEFAULT_RENESAS_SCIF_BAUD
fdt_helper.c, 1155
DEFAULT_RENESAS_SCIF_FREQ
fdt_helper.c, 1155
DEFAULT_SHAKTI_UART_BAUD
fdt_helper.c, 1156
DEFAULT_SHAKTI_UART_FREQ
fdt_helper.c, 1156
DEFAULT_SIFIVE_UART_BAUD
fdt_helper.c, 1156
DEFAULT_SIFIVE_UART_FREQ
fdt_helper.c, 1156
DEFAULT_UART_BAUD
fdt_helper.c, 1156
DEFAULT_UART_FREQ
fdt_helper.c, 1156
DEFAULT_UART_REG_IO_WIDTH
fdt_helper.c, 1156
DEFAULT_UART_REG_OFFSET
fdt_helper.c, 1156
DEFAULT_UART_REG_SHIFT
fdt_helper.c, 1156
DEFINE_BITMAP
sbi_bitmap.h, 305
DEFINE_SPIN_LOCK
riscv_locks.h, 300
DEFINE_UNPRIVILEGED_LOAD_FUNCTION

sbi_unpriv.c, 1108
DEFINE_UNPRIVILEGED_STORE_FUNCTION
sbi_unpriv.c, 1109
delegate
aplic_data, 79
delegate_traps
sbi_hart.c, 930
detected
sbi_hart_features, 134
digit
qdivrem.c, 1451
direction_input
gpio_chip, 90
direction_output
gpio_chip, 90
disable_cb
sbi_sse_cb_ops, 150
DISABLE_DEVICES_MASK
fdt_domain.c, 1134
divdi3.c
__FBSDID, 1446
__divdi3, 1446
do_fixup
fdt_general_fixup, 83
docs/contributing.md, 162
docs/domain_support.md, 162
docs/firmware/fw.md, 162
docs/firmware/fw_dynamic.md, 162
docs/firmware/fw_jump.md, 162
docs/firmware/fw_payload.md, 162
docs/firmware/payload_linux.md, 162
docs/firmware/payload_uboot.md, 162
docs/library_usage.md, 162
docs/opensbi_config.md, 162
docs/platform/andes-ae350.md, 162
docs/platform/fpga-ariane.md, 162
docs/platform/fpga-openpiton.md, 162
docs/platform/generic.md, 162
docs/platform/nuclei_ux600.md, 162
docs/platform/platform.md, 162
docs/platform/qemu_virt.md, 162
docs/platform/renesas-rzfive.md, 162
docs/platform/shakti_cclass.md, 162
docs/platform/sifive_fu540.md, 162
docs/platform/spike.md, 162
docs/platform/thead-c9xx.md, 162
docs/platform_guide.md, 162
docs/platform_requirements.md, 162
docs/writing_tests.md, 162
domain_cleanup_data_one
sbi_domain_data.c, 862
domain_count
sbi_domain.c, 856
domain_finalized
sbi_domain.c, 856
domain_hart_ptr_offset
sbi_domain.c, 856
domain_list
sbi_domain.h, 367
domain_setup_data_one
sbi_domain_data.c, 863
domains_init
sbi_platform_operations, 144
driver
fdt_gpio, 83
fdt_mailbox, 85
gpio_chip, 90
mbox_controller, 94
DW_GPIO_BIT
fdt_gpio_designware.c, 1180
DW_GPIO_DDR
fdt_gpio_designware.c, 1180
dw_gpio_direction_output
fdt_gpio_designware.c, 1181
DW_GPIO_DR
fdt_gpio_designware.c, 1180
dw_gpio_init_bank
fdt_gpio_designware.c, 1181
dw_gpio_match
fdt_gpio_designware.c, 1182
DW_GPIO_PINS_MAX
fdt_gpio_designware.c, 1180
dw_gpio_set
fdt_gpio_designware.c, 1181
dw_i2c.c
dw_i2c_adapter_poll, 1203
dw_i2c_adapter_poll_rxrdy, 1200
dw_i2c_adapter_poll_txfifo_ready, 1200
dw_i2c_adapter_read, 1203
dw_i2c_adapter_write, 1204
dw_i2c_getreg, 1204
dw_i2c_init, 1205
dw_i2c_setreg, 1205
DW_I2C_STATUS_RXFIFO_NOT_EMPTY, 1200
DW_I2C_STATUS_TXFIFO_EMPTY, 1200
dw_i2c_write_addr, 1206
DW_IC_CLR_ACTIVITY, 1200
DW_IC_CLR_GEN_CALL, 1200
DW_IC_CLR_INTR, 1200
DW_IC_CLR_RD_REQ, 1200
DW_IC_CLR_RESTART_DET, 1200
DW_IC_CLR_RX_DONE, 1200
DW_IC_CLR_RX_OVER, 1201
DW_IC_CLR_RX_UNDER, 1201
DW_IC_CLR_START_DET, 1201
DW_IC_CLR_STOP_DET, 1201
DW_IC_CLR_TX_ABRT, 1201
DW_IC_CLR_TX_OVER, 1201
DW_IC_COMP_PARAM_1, 1201
DW_IC_COMP_VERSION, 1201
DW_IC_CON, 1201
DW_IC_DATA_CMD, 1201
DW_IC_ENABLE, 1201
DW_IC_ENABLE_STATUS, 1201
DW_IC_FS_SCL_HCNT, 1201
DW_IC_FS_SCL_LCNT, 1201

DW_IC_HS_SCL_HCNT, 1201
DW_IC_HS_SCL_LCNT, 1202
DW_IC_INTR_MASK, 1202
DW_IC_INTR_STAT, 1202
DW_IC_RAW_INTR_STAT, 1202
DW_IC_RX_TL, 1202
DW_IC_RXFLR, 1202
DW_IC_SAR, 1202
DW_IC_SDA_HOLD, 1202
DW_IC_SS_SCL_HCNT, 1202
DW_IC_SS_SCL_LCNT, 1202
DW_IC_STATUS, 1202
DW_IC_TAR, 1202
DW_IC_TX_ABRT_SOURCE, 1202
DW_IC_TX_TL, 1202
DW_IC_TXFLR, 1202
IC_DATA_CMD_READ, 1203
IC_DATA_CMD_RESTART, 1203
IC_DATA_CMD_STOP, 1203
IC_INT_STATUS_STOPDET, 1203
dw_i2c.h
dw_i2c_init, 675
dw_i2c_adapter, 81
adapter, 82
addr, 82
dw_i2c_adapter_poll
dw_i2c.c, 1203
dw_i2c_adapter_poll_rxrdy
dw_i2c.c, 1200
dw_i2c_adapter_poll_txfifo_ready
dw_i2c.c, 1200
dw_i2c_adapter_read
dw_i2c.c, 1203
dw_i2c_adapter_write
dw_i2c.c, 1204
dw_i2c_getreg
dw_i2c.c, 1204
dw_i2c_init
dw_i2c.c, 1205
dw_i2c.h, 675
dw_i2c_setreg
dw_i2c.c, 1205
DW_I2C_STATUS_RXFIFO_NOT_EMPTY
dw_i2c.c, 1200
DW_I2C_STATUS_TXFIFO_EMPTY
dw_i2c.c, 1200
dw_i2c_write_addr
dw_i2c.c, 1206
DW_IC_CLR_ACTIVITY
dw_i2c.c, 1200
DW_IC_CLR_GEN_CALL
dw_i2c.c, 1200
DW_IC_CLR_INTR
dw_i2c.c, 1200
DW_IC_CLR_RD_REQ
dw_i2c.c, 1200
DW_IC_CLR_RESTART_DET
dw_i2c.c, 1200
DW_IC_CLR_RX_DONE
dw_i2c.c, 1200
DW_IC_CLR_RX_OVER
dw_i2c.c, 1201
DW_IC_CLR_RX_UNDER
dw_i2c.c, 1201
DW_IC_CLR_START_DET
dw_i2c.c, 1201
DW_IC_CLR_STOP_DET
dw_i2c.c, 1201
DW_IC_CLR_TX_ABRT
dw_i2c.c, 1201
DW_IC_CLR_TX_OVER
dw_i2c.c, 1201
DW_IC_COMP_PARAM_1
dw_i2c.c, 1201
DW_IC_COMP_VERSION
dw_i2c.c, 1201
DW_IC_CON
dw_i2c.c, 1201
DW_IC_DATA_CMD
dw_i2c.c, 1201
DW_IC_ENABLE
dw_i2c.c, 1201
DW_IC_ENABLE_STATUS
dw_i2c.c, 1201
DW_IC_FS_SCL_HCNT
dw_i2c.c, 1201
DW_IC_FS_SCL_LCNT
dw_i2c.c, 1201
DW_IC_HS_SCL_HCNT
dw_i2c.c, 1201
DW_IC_HS_SCL_LCNT
dw_i2c.c, 1202
DW_IC_INTR_MASK
dw_i2c.c, 1202
DW_IC_INTR_STAT
dw_i2c.c, 1202
DW_IC_RAW_INTR_STAT
dw_i2c.c, 1202
DW_IC_RX_TL
dw_i2c.c, 1202
DW_IC_RXFLR
dw_i2c.c, 1202
DW_IC_SAR
dw_i2c.c, 1202
DW_IC_SDA_HOLD
dw_i2c.c, 1202
DW_IC_SS_SCL_HCNT
dw_i2c.c, 1202
DW_IC_SS_SCL_LCNT
dw_i2c.c, 1202
DW_IC_STATUS
dw_i2c.c, 1202
DW_IC_TAR
dw_i2c.c, 1202
DW_IC_TX_ABRT_SOURCE
dw_i2c.c, 1202

DW_IC_TX_TL
dw_i2c.c, 1202
DW_IC_TXFLR
dw_i2c.c, 1202

early_exit
sbi_platform_operations, 144
early_init
sbi_platform_operations, 144
ebreak
riscv_asm.h, 168
ecall_base
sbi_ecall_base.c, 874
ecall_cppc
sbi_ecall_cppc.c, 876
ecall_dbcn
sbi_ecall_dbcn.c, 877
ecall_dbtr
sbi_ecall_dbtr.c, 879
ecall_fwft
sbi_ecall_fwft.c, 880
ecall_hsm
sbi_ecall_hsm.c, 882
ecall_impid
sbi_ecall.c, 872
ecall_ipi
sbi_ecall_ipi.c, 884
ecall_legacy
sbi_ecall_legacy.c, 886
ecall_mpxy
sbi_ecall_mpxy.c, 888
ecall_pmu
sbi_ecall_pmu.c, 890
ecall_rfence
sbi_ecall_rfence.c, 891
ecall_srst
sbi_ecall_srst.c, 894
ecall_sse
sbi_ecall_sse.c, 896
ecall_susp
sbi_ecall_susp.c, 898
ecall_time
sbi_ecall_time.c, 899
ecall_vendor
sbi_ecall_vendor.c, 901
eidx
fdt_pmu_hw_event_select_map, 87
eidx_end
fdt_pmu_hw_event_counter_map, 86
eidx_start
fdt_pmu_hw_event_counter_map, 86
emulate_load
sbi_platform_operations, 144
emulate_store
sbi_platform_operations, 144
enable_cb
sbi_sse_cb_ops, 150
entry_count_offset
sbi_init.c, 981
entry_latency_us
rpmi_hsm_get_susp_info_resp, 115
sbi_cpu_idle_state, 124
entry_size
sbi_fifo, 133
ENVCFG_ADUE
riscv_encoding.h, 244
ENVCFG_ADUE_SHIFT
riscv_encoding.h, 244
ENVCFG_CBCFE
riscv_encoding.h, 244
ENVCFG_CBIE
riscv_encoding.h, 244
ENVCFG_CBIE_FLUSH
riscv_encoding.h, 244
ENVCFG_CBIE_ILL
riscv_encoding.h, 244
ENVCFG_CBIE_INV
riscv_encoding.h, 244
ENVCFG_CBIE_SHIFT
riscv_encoding.h, 244
ENVCFG_CBZE
riscv_encoding.h, 244
ENVCFG_CDE
riscv_encoding.h, 244
ENVCFG_DTE
riscv_encoding.h, 244
ENVCFG_DTE_SHIFT
riscv_encoding.h, 244
ENVCFG_FIOM
riscv_encoding.h, 244
ENVCFG_LPE
riscv_encoding.h, 244
ENVCFG_LPE_SHIFT
riscv_encoding.h, 244
ENVCFG_PBMTE
riscv_encoding.h, 245
ENVCFG_PMM
riscv_encoding.h, 245
ENVCFG_PMM_PMLEN_0
riscv_encoding.h, 245
ENVCFG_PMM_PMLEN_16
riscv_encoding.h, 245
ENVCFG_PMM_PMLEN_7
riscv_encoding.h, 245
ENVCFG_SSE
riscv_encoding.h, 245
ENVCFG_SSE_SHIFT
riscv_encoding.h, 245
ENVCFG_STCE
riscv_encoding.h, 245
EVENT_COUNT
sbi_sse.c, 1034
event_data
sbi_pmu_event_info, 147
event_idx
sbi_pmu_event_info, 147
EVENT_IS_GLOBAL

sbi_sse.c, 1034
eventid
rpmi_enable_notification_req, 112
eventsstate_ctrl
sbi_mpxy_channel_attrs, 140
EXC_MODE_PP
sbi_sse.h, 566
EXC_MODE_PP_SHIFT
sbi_sse.h, 566
EXC_MODE_PV
sbi_sse.h, 566
EXC_MODE_PV_SHIFT
sbi_sse.h, 566
EXC_MODE_SSTATUS_SPIE
sbi_sse.h, 566
EXC_MODE_SSTATUS_SPIE_SHIFT
sbi_sse.h, 566
exit_latency_us
rpmi_hsm_get_susp_info_resp, 115
sbi_cpu_idle_state, 124
experimental
fdt_driver, 82
sbi_ecall_extension, 132
ext_irqfn
sbi_irqchip.c, 993
extensions
sbi_hart_features, 134
extensions_init
sbi_platform_operations, 144
extid_end
sbi_ecall_extension, 132
extid_start
sbi_ecall_extension, 132
extra_lock
sbi_scratch.c, 1031
extra_offset
sbi_scratch.c, 1031
EXTRACT_FIELD
sbi_bitops.h, 314

f0
rpmi_base_get_attributes_resp, 101
f1
rpmi_base_get_attributes_resp, 101
f2
rpmi_base_get_attributes_resp, 101
f3
rpmi_base_get_attributes_resp, 101
false
sbi_types.h, 619
fc_perf_feedback_offset_hi
rpmi_cppc_get_fastchan_offset_resp, 107
fc_perf_feedback_offset_lo
rpmi_cppc_get_fastchan_offset_resp, 107
fc_perf_request_offset_hi
rpmi_cppc_get_fastchan_offset_resp, 107
fc_perf_request_offset_lo
rpmi_cppc_get_fastchan_offset_resp, 107
fdt.c
check_block_, 1257
check_off_, 1258
fdt_check_header, 1258
fdt_check_node_offset_, 1259
fdt_check_prop_offset_, 1260
fdt_find_string_, 1261
fdt_first_subnode, 1261
fdt_header_size, 1262
fdt_header_size_, 1262
fdt_move, 1262
fdt_next_node, 1263
fdt_next_subnode, 1264
fdt_next_tag, 1265
fdt_offset_ptr, 1265
fdt_ro_probe_, 1266
fdt.h
FDT_BEGIN_NODE, 1267
FDT_END, 1267
FDT_END_NODE, 1268
FDT_MAGIC, 1268
FDT_NOP, 1268
FDT_PROP, 1268
FDT_TAGSIZE, 1268
FDT_V16_SIZE, 1268
FDT_V17_SIZE, 1268
FDT_V1_SIZE, 1268
FDT_V2_SIZE, 1268
FDT_V3_SIZE, 1268
fdt16_t
libfdt_env.h, 1430
fdt16_to_cpu
libfdt_env.h, 1431
fdt32_ld
libfdt.h, 1362
fdt32_ld_
libfdt_internal.h, 1436
fdt32_st
libfdt.h, 1362
fdt32_t
libfdt_env.h, 1430
fdt32_to_cpu
libfdt_env.h, 1431
fdt64_ld
libfdt.h, 1362
fdt64_ld_
libfdt_internal.h, 1437
fdt64_st
libfdt.h, 1362
fdt64_t
libfdt_env.h, 1430
fdt64_to_cpu
libfdt_env.h, 1432
fdt_add_cpu_idle_states
fdt_fixup.c, 1146
fdt_fixup.h, 637
fdt_add_mem_rsv
fdt_rw.c, 1324
libfdt.h, 1362

fdt_add_property_
fdt_rw.c, 1324
fdt_add_reservemap_entry
fdt_sw.c, 1340
libfdt.h, 1363
fdt_add_string_
fdt_sw.c, 1341
fdt_add_subnode
fdt_rw.c, 1325
libfdt.h, 1363
fdt_add_subnode_namelen
fdt_rw.c, 1326
libfdt.h, 1364
fdt_address_cells
fdt_addresses.c, 1270
libfdt.h, 1364
fdt_addresses.c
fdt_address_cells, 1270
fdt_appendprop_addrrange, 1271
fdt_cells, 1271
fdt_size_cells, 1272
FDT_ALIGN
libfdt_internal.h, 1435
fdt_aplic_fixup
fdt_fixup.c, 1147
fdt_fixup.h, 637
fdt_appendprop
fdt_rw.c, 1326
libfdt.h, 1365
fdt_appendprop_addrrange
fdt_addresses.c, 1271
libfdt.h, 1366
fdt_appendprop_cell
libfdt.h, 1366
fdt_appendprop_string
libfdt.h, 1358
fdt_appendprop_u32
libfdt.h, 1367
fdt_appendprop_u64
libfdt.h, 1367
FDT_ASSUME_MASK
libfdt_internal.h, 1435
FDT_BEGIN_NODE
fdt.h, 1267
fdt_begin_node
fdt_sw.c, 1341
libfdt.h, 1368
FDT_BITWISE
libfdt_env.h, 1429
fdt_blocks_misordered_
fdt_rw.c, 1327
fdt_boot_cpuid_phys
libfdt.h, 1358
fdt_cells
fdt_addresses.c, 1271
fdt_check.c
fdt_check_full, 1274
fdt_check_full
fdt_check.c, 1274
libfdt.h, 1368
fdt_check_header
fdt.c, 1258
libfdt.h, 1369
fdt_check_imsic_mlevel
fdt_helper.c, 1156
fdt_helper.h, 645
fdt_check_node_offset_
fdt.c, 1259
libfdt_internal.h, 1437
fdt_check_prop_offset_
fdt.c, 1260
libfdt_internal.h, 1438
fdt_config_fixup
fdt_fixup.c, 1147
fdt_cppc.c
fdt_cppc_drivers, 1129
fdt_cppc_init, 1128
fdt_cppc.h
fdt_cppc_init, 629
fdt_cppc_drivers
fdt_cppc.c, 1129
fdt_cppc_init
fdt_cppc.c, 1128
fdt_cppc.h, 629
fdt_cppc_rpmi
fdt_cppc_rpmi.c, 1132
fdt_cppc_rpmi.c
fdt_cppc_rpmi, 1132
rpmi_cppc_cold_init, 1130
rpmi_cppc_fc_db_trigger, 1130
rpmi_cppc_get_pointer, 1130
RPMI_CPPC_HART_FASTCHAN_SIZE, 1130
rpmi_cppc_match, 1132
rpmi_cppc_offset, 1132
rpmi_cppc_probe, 1131
rpmi_cppc_read, 1131
rpmi_cppc_update_hart_scratch, 1131
rpmi_cppc_write, 1132
sbi_rpmi_cppc, 1133
fdt_cpu_fixup
fdt_fixup.c, 1148
fdt_fixup.h, 638
fdt_create
fdt_sw.c, 1341
libfdt.h, 1369
fdt_create_empty_tree
fdt_empty_tree.c, 1275
libfdt.h, 1370
FDT_CREATE_FLAG_NO_NAME_DEDUP
libfdt.h, 1358
FDT_CREATE_FLAGS_ALL
libfdt.h, 1358
fdt_create_with_flags
fdt_sw.c, 1342
libfdt.h, 1371
fdt_data_size_

fdt_rw.c, 1327
fdt_del_last_string_
fdt_rw.c, 1328
fdt_sw.c, 1342
fdt_del_mem_rsv
fdt_rw.c, 1328
libfdt.h, 1371
fdt_del_node
fdt_rw.c, 1329
libfdt.h, 1372
fdt_delprop
fdt_rw.c, 1329
libfdt.h, 1372
fdt_domain.c
__fdt_parse_domain, 1134
__fdt_parse_region, 1135
__fixup_count_disable_devices, 1135
__fixup_disable_devices, 1136
__fixup_find_domain_offset, 1136
DISABLE_DEVICES_MASK, 1134
fdt_domain_fixup, 1137
FDT_DOMAIN_REGION_MAX_COUNT, 1134
fdt_domains_populate, 1137
fdt_iterate_each_domain, 1138
fdt_iterate_each_domain_ro, 1138
fdt_iterate_each_memregion, 1139
fdt_iterate_each_memregion_ro, 1139
fdt_domain.h
fdt_domain_fixup, 631
fdt_domains_populate, 631
fdt_domain_based_fixup_one
fdt_fixup.c, 1148
fdt_domain_fixup
fdt_domain.c, 1137
fdt_domain.h, 631
FDT_DOMAIN_REGION_MAX_COUNT
fdt_domain.c, 1134
fdt_domains_populate
fdt_domain.c, 1137
fdt_domain.h, 631
fdt_driver, 82
experimental, 82
init, 82
match_table, 82
fdt_driver.c
fdt_driver_init_all, 1140
fdt_driver_init_by_offset, 1142
fdt_driver_init_one, 1143
fdt_driver_init_scan, 1144
fdt_driver.h
fdt_driver_init_all, 632
fdt_driver_init_by_offset, 633
fdt_driver_init_one, 634
fdt_driver_init_all
fdt_driver.c, 1140
fdt_driver.h, 632
fdt_driver_init_by_offset
fdt_driver.c, 1142
fdt_driver.h, 633
fdt_driver_init_one
fdt_driver.c, 1143
fdt_driver.h, 634
fdt_driver_init_scan
fdt_driver.c, 1144
fdt_dw_i2c_init
fdt_i2c_dw.c, 1209
fdt_dw_i2c_match
fdt_i2c_dw.c, 1210
fdt_empty_tree.c
fdt_create_empty_tree, 1275
FDT_END
fdt.h, 1267
FDT_END_NODE
fdt.h, 1268
fdt_end_node
fdt_sw.c, 1343
libfdt.h, 1373
FDT_ERR_ALIGNMENT
libfdt.h, 1358
FDT_ERR_BADFLAGS
libfdt.h, 1358
FDT_ERR_BADLAYOUT
libfdt.h, 1358
FDT_ERR_BADMAGIC
libfdt.h, 1358
FDT_ERR_BADNCELLS
libfdt.h, 1358
FDT_ERR_BADOFFSET
libfdt.h, 1358
FDT_ERR_BADOVERLAY
libfdt.h, 1358
FDT_ERR_BADPATH
libfdt.h, 1358
FDT_ERR_BADPHANDLE
libfdt.h, 1358
FDT_ERR_BADSTATE
libfdt.h, 1358
FDT_ERR_BADSTRUCTURE
libfdt.h, 1359
FDT_ERR_BADVALUE
libfdt.h, 1359
FDT_ERR_BADVERSION
libfdt.h, 1359
FDT_ERR_EXISTS
libfdt.h, 1359
FDT_ERR_INTERNAL
libfdt.h, 1359
FDT_ERR_MAX
libfdt.h, 1359
FDT_ERR_NOPHANDLES
libfdt.h, 1359
FDT_ERR_NOSPACE
libfdt.h, 1359
FDT_ERR_NOTFOUND
libfdt.h, 1359
FDT_ERR_TRUNCATED

libfdt.h, 1359
FDT_ERRTABENT
fdt_strerror.c, 1338
fdt_errtable
fdt_strerror.c, 1338
FDT_ERRTABSIZE
fdt_strerror.c, 1338
fdt_find_add_string_
fdt_rw.c, 1330
fdt_sw.c, 1343
fdt_find_match
fdt_helper.c, 1157
fdt_helper.h, 645
fdt_find_max_phandle
fdt_ro.c, 1288
libfdt.h, 1373
fdt_find_string_
fdt.c, 1261
libfdt_internal.h, 1439
fdt_finish
fdt_sw.c, 1344
libfdt.h, 1374
fdt_finish_reservemap
fdt_sw.c, 1344
libfdt.h, 1375
fdt_first_property_offset
fdt_ro.c, 1288
libfdt.h, 1375
fdt_first_subnode
fdt.c, 1261
libfdt.h, 1376
FDT_FIRST_SUPPORTED_VERSION
libfdt.h, 1359
fdt_fixup.c
fdt_add_cpu_idle_states, 1146
fdt_aplic_fixup, 1147
fdt_config_fixup, 1147
fdt_cpu_fixup, 1148
fdt_domain_based_fixup_one, 1148
fdt_fixup_node, 1149
fdt_fixups, 1149
fdt_imsic_fixup, 1150
fdt_plic_fixup, 1151
fdt_register_general_fixup, 1151
fdt_reserved_memory_fixup, 1152
fdt_resv_memory_update_node, 1153
fdt_unregister_general_fixup, 1153
SBI_LIST_HEAD, 1154
fdt_fixup.h
fdt_add_cpu_idle_states, 637
fdt_aplic_fixup, 637
fdt_cpu_fixup, 638
fdt_fixups, 639
fdt_imsic_fixup, 639
fdt_plic_fixup, 640
fdt_register_general_fixup, 641
fdt_reserved_memory_fixup, 641
fdt_unregister_general_fixup, 642
fdt_fixup_node
fdt_fixup.c, 1149
fdt_fixups
fdt_fixup.c, 1149
fdt_fixup.h, 639
fdt_for_each_property_offset
libfdt.h, 1359
fdt_for_each_subnode
libfdt.h, 1359
FDT_FORCE
libfdt_env.h, 1429
fdt_general_fixup, 82
do_fixup, 83
head, 83
name, 83
fdt_generate_phandle
fdt_ro.c, 1290
libfdt.h, 1377
fdt_get_aclint_addr_size
fdt_helper.c, 1157
fdt_get_aclint_addr_size_by_name
fdt_helper.c, 1158
fdt_get_address
fdt_helper.h, 646
fdt_get_address_rw
fdt_helper.h, 646
fdt_get_alias
fdt_ro.c, 1291
libfdt.h, 1377
fdt_get_alias_namelen
fdt_ro.c, 1291
libfdt.h, 1377
fdt_get_header
libfdt.h, 1360
fdt_get_max_phandle
libfdt.h, 1378
fdt_get_mem_rsv
fdt_ro.c, 1292
libfdt.h, 1378
fdt_get_name
fdt_ro.c, 1292
libfdt.h, 1379
fdt_get_node_addr_size
fdt_helper.c, 1158
fdt_helper.h, 646
fdt_get_node_addr_size_by_name
fdt_helper.c, 1159
fdt_helper.h, 647
fdt_get_path
fdt_ro.c, 1293
libfdt.h, 1380
fdt_get_phandle
fdt_ro.c, 1294
libfdt.h, 1381
fdt_get_property
fdt_ro.c, 1295
libfdt.h, 1381
fdt_get_property_by_offset

fdt_ro.c, 1295
libfdt.h, 1382
fdt_get_property_by_offset_
fdt_ro.c, 1296
fdt_get_property_namelen
fdt_ro.c, 1297
libfdt.h, 1382
fdt_get_property_namelen_
fdt_ro.c, 1298
fdt_get_property_w
libfdt.h, 1383
fdt_get_string
fdt_ro.c, 1299
libfdt.h, 1383
fdt_getprop
fdt_ro.c, 1300
libfdt.h, 1385
fdt_getprop_by_offset
fdt_ro.c, 1301
libfdt.h, 1385
fdt_getprop_namelen
fdt_ro.c, 1301
libfdt.h, 1386
fdt_getprop_namelen_w
libfdt.h, 1387
fdt_getprop_w
libfdt.h, 1388
fdt_gpio, 83
driver, 83
xlate, 83
fdt_gpio.c
fdt_gpio_chip_find, 1178
fdt_gpio_drivers, 1179
fdt_gpio_init, 1178
fdt_gpio_pin_get, 1179
fdt_gpio_simple_xlate, 1179
fdt_gpio.h
fdt_gpio_pin_get, 665
fdt_gpio_simple_xlate, 666
fdt_gpio_chip_find
fdt_gpio.c, 1178
fdt_gpio_designware
fdt_gpio_designware.c, 1182
fdt_gpio_designware.c
DW_GPIO_BIT, 1180
DW_GPIO_DDR, 1180
dw_gpio_direction_output, 1181
DW_GPIO_DR, 1180
dw_gpio_init_bank, 1181
dw_gpio_match, 1182
DW_GPIO_PINS_MAX, 1180
dw_gpio_set, 1181
fdt_gpio_designware, 1182
pin_to_chip, 1181
fdt_gpio_drivers
fdt_gpio.c, 1179
fdt_gpio_init
fdt_gpio.c, 1178
fdt_gpio_pin_get
fdt_gpio.c, 1179
fdt_gpio.h, 665
fdt_gpio_sifive
fdt_gpio_sifive.c, 1184
fdt_gpio_sifive.c
fdt_gpio_sifive, 1184
SIFIVE_GPIO_BIT, 1183
sifive_gpio_direction_output, 1183
sifive_gpio_init, 1184
sifive_gpio_match, 1185
SIFIVE_GPIO_OUTEN, 1183
SIFIVE_GPIO_OUTVAL, 1183
SIFIVE_GPIO_PINS_DEF, 1183
SIFIVE_GPIO_PINS_MAX, 1183
SIFIVE_GPIO_PINS_MIN, 1183
sifive_gpio_set, 1184
fdt_gpio_simple_xlate
fdt_gpio.c, 1179
fdt_gpio.h, 666
fdt_gpio_starfive
fdt_gpio_starfive.c, 1187
fdt_gpio_starfive.c
fdt_gpio_starfive, 1187
starfive_gpio_direction_output, 1186
starfive_gpio_init, 1186
STARFIVE_GPIO_MASK, 1186
starfive_gpio_match, 1187
STARFIVE_GPIO_OUTVAL, 1186
STARFIVE_GPIO_PINS_DEF, 1186
STARFIVE_GPIO_REG_SHIFT_MASK, 1186
starfive_gpio_set, 1187
STARFIVE_GPIO_SHIFT_BITS, 1186
fdt_grab_space_
fdt_sw.c, 1345
fdt_header, 83
boot_cpuid_phys, 84
last_comp_version, 84
magic, 84
off_dt_strings, 84
off_dt_struct, 84
off_mem_rsvmap, 84
size_dt_strings, 84
size_dt_struct, 84
totalsize, 84
version, 84
fdt_header_size
fdt.c, 1262
libfdt.h, 1388
fdt_header_size_
fdt.c, 1262
libfdt.h, 1389
fdt_helper.c
DEFAULT_RENESAS_SCIF_BAUD, 1155
DEFAULT_RENESAS_SCIF_FREQ, 1155
DEFAULT_SHAKTI_UART_BAUD, 1156
DEFAULT_SHAKTI_UART_FREQ, 1156
DEFAULT_SIFIVE_UART_BAUD, 1156

DEFAULT_SIFIVE_UART_FREQ, 1156
DEFAULT_UART_BAUD, 1156
DEFAULT_UART_FREQ, 1156
DEFAULT_UART_REG_IO_WIDTH, 1156
DEFAULT_UART_REG_OFFSET, 1156
DEFAULT_UART_REG_SHIFT, 1156
fdt_check_imsic_mlevel, 1156
fdt_find_match, 1157
fdt_get_aclint_addr_size, 1157
fdt_get_aclint_addr_size_by_name, 1158
fdt_get_node_addr_size, 1158
fdt_get_node_addr_size_by_name, 1159
fdt_isa_bitmap_offset, 1175
fdt_match_node, 1160
fdt_node_is_enabled, 1160
fdt_parse_aclint_node, 1161
fdt_parse_aplic_node, 1162
fdt_parse_compat_addr, 1162
fdt_parse_gaisler_uart_node, 1163
fdt_parse_hart_id, 1163
fdt_parse_imsic_node, 1164
fdt_parse_isa_all_harts, 1165
fdt_parse_isa_extensions, 1165
fdt_parse_isa_extensions_one_hart, 1166
fdt_parse_isa_one_hart, 1166
fdt_parse_max_enabled_hart_id, 1167
fdt_parse_phandle_with_args, 1167
fdt_parse_plic, 1167
fdt_parse_plic_node, 1168
fdt_parse_plicsw_node, 1168
fdt_parse_plmt_node, 1169
fdt_parse_renesas_scif_node, 1170
fdt_parse_shakti_uart_node, 1170
fdt_parse_sifive_uart_node, 1171
fdt_parse_timebase_frequency, 1171
fdt_parse_uart8250, 1172
fdt_parse_uart_node, 1172
fdt_parse_uart_node_common, 1173
fdt_parse_xlnx_uartlite_node, 1174
fdt_translate_address, 1174
RISCV_ISA_EXT_NAME_LEN_MAX, 1156
set_multi_letter_ext, 1156
fdt_helper.h
fdt_check_imsic_mlevel, 645
fdt_find_match, 645
fdt_get_address, 646
fdt_get_address_rw, 646
fdt_get_node_addr_size, 646
fdt_get_node_addr_size_by_name, 647
fdt_match_node, 648
FDT_MAX_PHANDLE_ARGS, 645
fdt_node_is_enabled, 648
fdt_parse_aclint_node, 649
fdt_parse_aplic_node, 650
fdt_parse_compat_addr, 650
fdt_parse_gaisler_uart_node, 651
fdt_parse_hart_id, 651
fdt_parse_imsic_node, 652
fdt_parse_isa_extensions, 653
fdt_parse_max_enabled_hart_id, 653
fdt_parse_phandle_with_args, 654
fdt_parse_plic, 654
fdt_parse_plic_node, 655
fdt_parse_plicsw_node, 655
fdt_parse_plmt_node, 656
fdt_parse_renesas_scif_node, 657
fdt_parse_shakti_uart_node, 657
fdt_parse_sifive_uart_node, 658
fdt_parse_timebase_frequency, 658
fdt_parse_uart8250, 659
fdt_parse_uart_node, 659
fdt_parse_xlnx_uartlite_node, 660
fdt_hsm.c
fdt_hsm_drivers, 1192
fdt_hsm_init, 1192
fdt_hsm.h
fdt_hsm_init, 673
fdt_hsm_drivers
fdt_hsm.c, 1192
fdt_hsm_init
fdt_hsm.c, 1192
fdt_hsm.h, 673
fdt_hsm_rpmi
fdt_hsm_rpmi.c, 1198
fdt_hsm_rpmi.c
fdt_hsm_rpmi, 1198
is_rpmi_hsm_susp_supported, 1193
MAX_HSM_SUPSEND_STATE_NAMELEN, 1193
rpmi_hsm_cold_init, 1193
rpmi_hsm_do_fixup, 1194
rpmi_hsm_fixup, 1198
rpmi_hsm_get_num_suspend_states, 1194
rpmi_hsm_get_pointer, 1195
rpmi_hsm_get_suspend_states, 1195
rpmi_hsm_match, 1198
rpmi_hsm_offset, 1198
rpmi_hsm_start, 1196
rpmi_hsm_stop, 1196
rpmi_hsm_suspend, 1196
rpmi_hsm_update_hart_scratch, 1197
sbi_hsm_rpmi, 1198
fdt_i2c.c
fdt_i2c_adapter_drivers, 1208
fdt_i2c_adapter_find, 1207
fdt_i2c_adapter_get, 1208
fdt_i2c.h
fdt_i2c_adapter_get, 677
fdt_i2c_adapter_drivers
fdt_i2c.c, 1208
fdt_i2c_adapter_dw
fdt_i2c_dw.c, 1210
fdt_i2c_adapter_find
fdt_i2c.c, 1207
fdt_i2c_adapter_get
fdt_i2c.c, 1208
fdt_i2c.h, 677

fdt_i2c_adapter_sifive
fdt_i2c_sifive.c, 1217
fdt_i2c_dw.c
fdt_dw_i2c_init, 1209
fdt_dw_i2c_match, 1210
fdt_i2c_adapter_dw, 1210
fdt_i2c_sifive.c
fdt_i2c_adapter_sifive, 1217
sifive_i2c_adapter_poll, 1213
sifive_i2c_adapter_poll_busy, 1211
sifive_i2c_adapter_poll_tip, 1211
sifive_i2c_adapter_read, 1213
sifive_i2c_adapter_rxack, 1214
sifive_i2c_adapter_start, 1214
sifive_i2c_adapter_write, 1215
SIFIVE_I2C_CMD_ACK, 1211
SIFIVE_I2C_CMD_IACK, 1211
SIFIVE_I2C_CMD_RD, 1211
SIFIVE_I2C_CMD_STA, 1211
SIFIVE_I2C_CMD_STO, 1211
SIFIVE_I2C_CMD_WR, 1212
SIFIVE_I2C_CR, 1212
SIFIVE_I2C_CTR, 1212
SIFIVE_I2C_CTR_EN, 1212
SIFIVE_I2C_CTR_IEN, 1212
sifive_i2c_getreg, 1216
sifive_i2c_init, 1216
sifive_i2c_match, 1217
SIFIVE_I2C_PREHI, 1212
SIFIVE_I2C_PRELO, 1212
SIFIVE_I2C_READ_BIT, 1212
SIFIVE_I2C_RXR, 1212
sifive_i2c_setreg, 1216
SIFIVE_I2C_SR, 1212
SIFIVE_I2C_STATUS_AL, 1212
SIFIVE_I2C_STATUS_BUSY, 1212
SIFIVE_I2C_STATUS_IF, 1212
SIFIVE_I2C_STATUS_RXACK, 1212
SIFIVE_I2C_STATUS_TIP, 1212
SIFIVE_I2C_TXR, 1213
SIFIVE_I2C_WRITE_BIT, 1213
fdt_imsic_fixup
fdt_fixup.c, 1150
fdt_fixup.h, 639
fdt_ipi.c
fdt_ipi_drivers, 1225
fdt_ipi_init, 1225
fdt_ipi.h
fdt_ipi_init, 688
fdt_ipi_drivers
fdt_ipi.c, 1225
fdt_ipi_init
fdt_ipi.c, 1225
fdt_ipi.h, 688
fdt_ipi_mswi
fdt_ipi_mswi.c, 1226
fdt_ipi_mswi.c
clint_offset, 1226
fdt_ipi_mswi, 1226
ipi_mswi_cold_init, 1226
ipi_mswi_match, 1226
fdt_ipi_plicsw
fdt_ipi_plicsw.c, 1228
fdt_ipi_plicsw.c
fdt_ipi_plicsw, 1228
fdt_plicsw_cold_ipi_init, 1227
ipi_plicsw_match, 1228
plicsw, 1228
fdt_irqchip.c
fdt_irqchip_drivers, 1237
fdt_irqchip_init, 1237
fdt_irqchip.h
fdt_irqchip_init, 692
fdt_irqchip_aplic
fdt_irqchip_aplic.c, 1238
fdt_irqchip_aplic.c
fdt_irqchip_aplic, 1238
irqchip_aplic_cold_init, 1238
irqchip_aplic_match, 1238
fdt_irqchip_drivers
fdt_irqchip.c, 1237
fdt_irqchip_imsic
fdt_irqchip_imsic.c, 1240
fdt_irqchip_imsic.c
fdt_irqchip_imsic, 1240
irqchip_imsic_cold_init, 1239
irqchip_imsic_match, 1240
irqchip_imsic_update_hartid_table, 1239
fdt_irqchip_init
fdt_irqchip.c, 1237
fdt_irqchip.h, 692
fdt_irqchip_plic
fdt_irqchip_plic.c, 1242
fdt_irqchip_plic.c
fdt_irqchip_plic, 1242
irqchip_plic_cold_init, 1241
irqchip_plic_match, 1242
irqchip_plic_update_context_map, 1241
fdt_isa_bitmap_offset
fdt_helper.c, 1175
fdt_iterate_each_domain
fdt_domain.c, 1138
fdt_iterate_each_domain_ro
fdt_domain.c, 1138
fdt_iterate_each_memregion
fdt_domain.c, 1139
fdt_iterate_each_memregion_ro
fdt_domain.c, 1139
fdt_last_comp_version
libfdt.h, 1360
FDT_LAST_COMPATIBLE_VERSION
libfdt.h, 1360
FDT_LAST_SUPPORTED_VERSION
libfdt.h, 1360
FDT_MAGIC
fdt.h, 1268

fdt_magic
libfdt.h, 1360
fdt_mailbox, 84
driver, 85
xlate, 85
fdt_mailbox.c
fdt_mailbox_drivers, 1463
fdt_mailbox_request_chan, 1461
fdt_mailbox_simple_xlate, 1462
fdt_mbox_controller_find, 1462
fdt_mailbox.h
fdt_mailbox_request_chan, 700
fdt_mailbox_simple_xlate, 700
fdt_mailbox_drivers
fdt_mailbox.c, 1463
fdt_mailbox_request_chan
fdt_mailbox.c, 1461
fdt_mailbox.h, 700
fdt_mailbox_rpmi_shmem
fdt_mailbox_rpmi_shmem.c, 1474
fdt_mailbox_rpmi_shmem.c
__packed, 1474
__smq_queue_empty, 1466
__smq_queue_full, 1467
__smq_rx, 1467
__smq_tx, 1467
fdt_mailbox_rpmi_shmem, 1474
GET_DLEN, 1464
GET_FLAGS, 1464
GET_MESSAGE_ID, 1465
GET_MESSAGE_TYPE, 1465
GET_SERVICE_ID, 1465
GET_SERVICEGROUP_ID, 1465
GET_TOKEN, 1465
MAKE_MESSAGE_ID, 1465
RPMI_BASE_VERSION_MIN, 1466
rpmi_get_platform_info, 1468
rpmi_queue_idx, 1466
RPMI_QUEUE_IDX_A2P_ACK, 1466
RPMI_QUEUE_IDX_A2P_REQ, 1466
RPMI_QUEUE_IDX_MAX_COUNT, 1466
RPMI_QUEUE_IDX_P2A_ACK, 1466
RPMI_QUEUE_IDX_P2A_REQ, 1466
rpmi_queue_type, 1466
RPMI_QUEUE_TYPE_ACK, 1466
RPMI_QUEUE_TYPE_REQ, 1466
rpmi_reg_idx, 1466
RPMI_REG_IDX_DB_REG, 1466
RPMI_REG_IDX_MAX_COUNT, 1466
rpmi_shmem_mbox_free_chan, 1468
rpmi_shmem_mbox_init, 1469
rpmi_shmem_mbox_match, 1474
rpmi_shmem_mbox_request_chan, 1470
rpmi_shmem_mbox_xfer, 1471
rpmi_shmem_transport_init, 1471
smq_base_get_two_u32, 1472
smq_rx, 1472
smq_tx, 1473
to_srvgrp_chan, 1466
fdt_mailbox_simple_xlate
fdt_mailbox.c, 1462
fdt_mailbox.h, 700
fdt_match, 85
compatible, 85
data, 85
fdt_match_node
fdt_helper.c, 1160
fdt_helper.h, 648
FDT_MAX_NCELLS
libfdt.h, 1360
FDT_MAX_PHANDLE
libfdt.h, 1360
FDT_MAX_PHANDLE_ARGS
fdt_helper.h, 645
fdt_mbox_controller_find
fdt_mailbox.c, 1462
fdt_mem_rsv
fdt_ro.c, 1303
fdt_mem_rsv_
libfdt_internal.h, 1440
fdt_mem_rsv_w_
libfdt_internal.h, 1440
fdt_move
fdt.c, 1262
libfdt.h, 1389
fdt_mpxy.c
fdt_mpxy_drivers, 1483
fdt_mpxy_init, 1483
fdt_mpxy.h
fdt_mpxy_init, 733
fdt_mpxy_drivers
fdt_mpxy.c, 1483
fdt_mpxy_init
fdt_mpxy.c, 1483
fdt_mpxy.h, 733
fdt_mpxy_rpmi_mbox
fdt_mpxy_rpmi_mbox.c, 1490
fdt_mpxy_rpmi_mbox.c
__mpxy_mbox_send_message, 1485
assert_field_offset, 1485, 1486
attr_id2index, 1485
clock_data, 1490
clock_services, 1490
fdt_mpxy_rpmi_mbox, 1490
mpxy_check_write_attr, 1486
mpxy_copy_attrs, 1486
mpxy_find_rpmi_srvid, 1487
mpxy_mbox_get_notifications, 1487
mpxy_mbox_init, 1487
mpxy_mbox_match, 1491
mpxy_mbox_read_attributes, 1488
mpxy_mbox_send_message_withoutresp, 1488
mpxy_mbox_send_message_withresp, 1489
mpxy_mbox_write_attributes, 1489
mpxy_msgprot_rpmi_attr_id, 1485
MPXY_MSGPROT_RPMI_ATTR_MAX_ID, 1485

MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_ID, 1485
MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_VERSION, 1485
mpxy_write_attr, 1490
RPMI_MAJOR_VER, 1485
RPMI_MINOR_VER, 1485
fdt_next_node
fdt.c, 1263
libfdt.h, 1389
fdt_next_property_offset
fdt_ro.c, 1304
libfdt.h, 1391
fdt_next_subnode
fdt.c, 1264
libfdt.h, 1392
fdt_next_tag
fdt.c, 1265
libfdt.h, 1393
fdt_node_check_compatible
fdt_ro.c, 1305
libfdt.h, 1394
fdt_node_depth
fdt_ro.c, 1306
libfdt.h, 1395
fdt_node_end_offset_
fdt_wip.c, 1349
libfdt_internal.h, 1441
fdt_node_header, 85
name, 85
tag, 85
fdt_node_is_enabled
fdt_helper.c, 1160
fdt_helper.h, 648
fdt_node_offset_by_compatible
fdt_ro.c, 1307
libfdt.h, 1396
fdt_node_offset_by_phandle
fdt_ro.c, 1308
libfdt.h, 1397
fdt_node_offset_by_prop_value
fdt_ro.c, 1309
libfdt.h, 1398
fdt_nodename_eq_
fdt_ro.c, 1310
FDT_NOP
fdt.h, 1268
fdt_nop_node
fdt_wip.c, 1349
libfdt.h, 1399
fdt_nop_property
fdt_wip.c, 1350
libfdt.h, 1399
fdt_nop_region_
fdt_wip.c, 1350
fdt_num_mem_rsv
fdt_ro.c, 1310
libfdt.h, 1400
fdt_off_dt_strings
libfdt.h, 1360
fdt_off_dt_struct
libfdt.h, 1360
fdt_off_mem_rsvmap
libfdt.h, 1361
fdt_offset_ptr
fdt.c, 1265
libfdt.h, 1400
fdt_offset_ptr_
libfdt_internal.h, 1441
fdt_offset_ptr_w
libfdt.h, 1401
fdt_offset_ptr_w_
libfdt_internal.h, 1442
fdt_open_into
fdt_rw.c, 1330
libfdt.h, 1402
fdt_overlay.c
fdt_overlay_apply, 1277
get_path_len, 1278
overlay_adjust_local_phandles, 1278
overlay_adjust_node_phandles, 1279
overlay_apply_node, 1279
overlay_fixup_one_phandle, 1280
overlay_fixup_phandle, 1281
overlay_fixup_phandles, 1281
overlay_get_target, 1282
overlay_get_target_phandle, 1282
overlay_merge, 1283
overlay_phandle_add_offset, 1284
overlay_symbol_update, 1284
overlay_update_local_node_references, 1285
overlay_update_local_references, 1286
fdt_overlay_apply
fdt_overlay.c, 1277
libfdt.h, 1402
fdt_pack
fdt_rw.c, 1331
libfdt.h, 1403
fdt_packblocks_
fdt_rw.c, 1331
fdt_parent_offset
fdt_ro.c, 1311
libfdt.h, 1403
fdt_parse_aclint_node
fdt_helper.c, 1161
fdt_helper.h, 649
fdt_parse_aplic_node
fdt_helper.c, 1162
fdt_helper.h, 650
fdt_parse_compat_addr
fdt_helper.c, 1162
fdt_helper.h, 650
fdt_parse_gaisler_uart_node
fdt_helper.c, 1163
fdt_helper.h, 651
fdt_parse_hart_id

fdt_helper.c, 1163
fdt_helper.h, 651
fdt_parse_imsic_node
fdt_helper.c, 1164
fdt_helper.h, 652
fdt_parse_isa_all_harts
fdt_helper.c, 1165
fdt_parse_isa_extensions
fdt_helper.c, 1165
fdt_helper.h, 653
fdt_parse_isa_extensions_one_hart
fdt_helper.c, 1166
fdt_parse_isa_one_hart
fdt_helper.c, 1166
fdt_parse_max_enabled_hart_id
fdt_helper.c, 1167
fdt_helper.h, 653
fdt_parse_phandle_with_args
fdt_helper.c, 1167
fdt_helper.h, 654
fdt_parse_plic
fdt_helper.c, 1167
fdt_helper.h, 654
fdt_parse_plic_node
fdt_helper.c, 1168
fdt_helper.h, 655
fdt_parse_plicsw_node
fdt_helper.c, 1168
fdt_helper.h, 655
fdt_parse_plmt_node
fdt_helper.c, 1169
fdt_helper.h, 656
fdt_parse_renesas_scif_node
fdt_helper.c, 1170
fdt_helper.h, 657
fdt_parse_shakti_uart_node
fdt_helper.c, 1170
fdt_helper.h, 657
fdt_parse_sifive_uart_node
fdt_helper.c, 1171
fdt_helper.h, 658
fdt_parse_timebase_frequency
fdt_helper.c, 1171
fdt_helper.h, 658
fdt_parse_uart8250
fdt_helper.c, 1172
fdt_helper.h, 659
fdt_parse_uart_node
fdt_helper.c, 1172
fdt_helper.h, 659
fdt_parse_uart_node_common
fdt_helper.c, 1173
fdt_parse_xlnx_uartlite_node
fdt_helper.c, 1174
fdt_helper.h, 660
fdt_path_offset
fdt_ro.c, 1312
libfdt.h, 1404
fdt_path_offset_namelen
fdt_ro.c, 1313
libfdt.h, 1405
fdt_phandle_args, 86
args, 86
args_count, 86
node_offset, 86
fdt_plic_fixup
fdt_fixup.c, 1151
fdt_fixup.h, 640
fdt_plicsw_cold_ipi_init
fdt_ipi_plicsw.c, 1227
fdt_plmt_cold_timer_init
fdt_timer_plmt.c, 1607
fdt_pmu.c
fdt_pmu_evt_select, 1177
fdt_pmu_fixup, 1176
fdt_pmu_get_select_value, 1176
FDT_PMU_HW_EVENT_MAX, 1176
fdt_pmu_setup, 1176
hw_event_count, 1177
fdt_pmu.h
fdt_pmu_fixup, 663
fdt_pmu_get_select_value, 664
fdt_pmu_setup, 664
fdt_pmu_evt_select
fdt_pmu.c, 1177
fdt_pmu_fixup
fdt_pmu.c, 1176
fdt_pmu.h, 663
fdt_pmu_get_select_value
fdt_pmu.c, 1176
fdt_pmu.h, 664
fdt_pmu_hw_event_counter_map, 86
ctr_map, 86
eidx_end, 86
eidx_start, 86
FDT_PMU_HW_EVENT_MAX
fdt_pmu.c, 1176
fdt_pmu_hw_event_select_map, 86
eidx, 87
select, 87
fdt_pmu_raw_event_counter_map, 87
ctr_map, 87
select, 87
select_mask, 87
fdt_pmu_setup
fdt_pmu.c, 1176
fdt_pmu.h, 664
fdt_poweroff_gpio
fdt_reset_gpio.c, 1513
FDT_PROP
fdt.h, 1268
fdt_property, 87
data, 87
fdt_sw.c, 1345
len, 87
libfdt.h, 1406

nameoff, 88
tag, 88
fdt_property_cell
libfdt.h, 1406
fdt_property_placeholder
fdt_sw.c, 1346
libfdt.h, 1407
fdt_property_string
libfdt.h, 1361
fdt_property_u32
libfdt.h, 1407
fdt_property_u64
libfdt.h, 1408
fdt_register_general_fixup
fdt_fixup.c, 1151
fdt_fixup.h, 641
fdt_regmap.c
fdt_regmap_drivers, 1493
fdt_regmap_find, 1491
fdt_regmap_get, 1492
fdt_regmap_get_by_phandle, 1492
fdt_regmap.h
fdt_regmap_get, 734
fdt_regmap_get_by_phandle, 735
fdt_regmap_drivers
fdt_regmap.c, 1493
fdt_regmap_find
fdt_regmap.c, 1491
fdt_regmap_get
fdt_regmap.c, 1492
fdt_regmap.h, 734
fdt_regmap_get_by_phandle
fdt_regmap.c, 1492
fdt_regmap.h, 735
fdt_regmap_syscon
fdt_regmap_syscon.c, 1500
fdt_regmap_syscon.c
fdt_regmap_syscon, 1500
regmap_syscon_init, 1495
regmap_syscon_match, 1500
regmap_syscon_read_16, 1495
regmap_syscon_read_32, 1496
regmap_syscon_read_8, 1496
regmap_syscon_read_be16, 1496
regmap_syscon_read_be32, 1497
regmap_syscon_read_le16, 1497
regmap_syscon_read_le32, 1497
regmap_syscon_write_16, 1498
regmap_syscon_write_32, 1498
regmap_syscon_write_8, 1498
regmap_syscon_write_be16, 1499
regmap_syscon_write_be32, 1499
regmap_syscon_write_le16, 1499
regmap_syscon_write_le32, 1500
SYSCON_ENDIAN_BIG, 1495
SYSCON_ENDIAN_LITTLE, 1495
SYSCON_ENDIAN_MAX, 1495
SYSCON_ENDIAN_NATIVE, 1495
syscon_regmap_endian, 1495
to_syscon_regmap, 1494
fdt_reserve_entry, 88
address, 88
size, 88
fdt_reserved_memory_fixup
fdt_fixup.c, 1152
fdt_fixup.h, 641
fdt_reset.c
fdt_reset_drivers, 1506
fdt_reset_init, 1506
fdt_reset.h
fdt_reset_init, 742
fdt_reset_atcwdt200
fdt_reset_atcwdt200.c, 1510
fdt_reset_atcwdt200.c
ae350_system_reset, 1509
ae350_system_reset_check, 1509
atcwdt200_reset, 1509
atcwdt200_reset_init, 1509
atcwdt200_reset_match, 1509
ATCWDT200_WP_NUM, 1507
CLK_PCLK, 1507
CTRL_REG, 1507
fdt_reset_atcwdt200, 1510
INT_CLK_1024, 1507
INT_CLK_16384, 1507
INT_CLK_2048, 1507
INT_CLK_256, 1507
INT_CLK_32768, 1508
INT_CLK_4096, 1508
INT_CLK_64, 1508
INT_CLK_8192, 1508
INT_EN, 1508
INT_TIME_MSK, 1508
INT_TIME_OFF, 1508
RST_CLK_1024, 1508
RST_CLK_128, 1508
RST_CLK_256, 1508
RST_CLK_512, 1508
RST_EN, 1508
RST_TIME_MSK, 1508
RST_TIME_OFF, 1508
smu, 1510
wdt_addr, 1510
WDT_EN, 1508
WREN_REG, 1509
fdt_reset_drivers
fdt_reset.c, 1506
fdt_reset_gpio
fdt_reset_gpio.c, 1514
fdt_reset_gpio.c
fdt_poweroff_gpio, 1513
fdt_reset_gpio, 1514
gpio_poweroff, 1514
gpio_poweroff_match, 1514
gpio_reset_exec, 1511
gpio_reset_get, 1511

gpio_reset_init, 1512
gpio_reset_match, 1514
gpio_restart, 1514
gpio_system_poweroff, 1512
gpio_system_poweroff_check, 1512
gpio_system_restart, 1513
gpio_system_restart_check, 1513
poweroff, 1514
restart, 1514
fdt_reset_htif
fdt_reset_htif.c, 1516
fdt_reset_htif.c
fdt_reset_htif, 1516
htif_reset_init, 1515
htif_reset_match, 1516
fdt_reset_init
fdt_reset.c, 1506
fdt_reset.h, 742
fdt_reset_rpmi
fdt_reset_rpmi.c, 1519
fdt_reset_rpmi.c
fdt_reset_rpmi, 1519
rpmi_do_system_reset, 1517
rpmi_reset_dev, 1519
rpmi_reset_init, 1517
rpmi_reset_match, 1519
rpmi_system_reset, 1518
rpmi_system_reset_check, 1518
rpmi_system_reset_type_check, 1518
sysreset_ctx, 1519
fdt_reset_sg2042_hwmon_mcu.c
CMD_POWEROFF, 1520
CMD_REBOOT, 1520
CMD_RESET, 1520
fdt_reset_sg2042_mcu, 1522
MANGO_BOARD_TYPE_MASK, 1520
mcu_adapter, 1522
mcu_reg, 1522
REG_BOARD_TYPE, 1520
REG_CMD, 1521
sg2042_mcu_reset, 1521
sg2042_mcu_reset_check, 1521
sg2042_mcu_reset_check_board, 1521
sg2042_mcu_reset_device, 1522
sg2042_mcu_reset_init, 1521
sg2042_mcu_reset_match, 1522
fdt_reset_sg2042_mcu
fdt_reset_sg2042_hwmon_mcu.c, 1522
fdt_reset_sunxi_wdt
fdt_reset_sunxi_wdt.c, 1524
fdt_reset_sunxi_wdt.c
fdt_reset_sunxi_wdt, 1524
sunxi_wdt_base, 1524
sunxi_wdt_reset, 1524
sunxi_wdt_reset_init, 1524
sunxi_wdt_reset_match, 1524
sunxi_wdt_system_reset, 1524
sunxi_wdt_system_reset_check, 1524
WDT_KEY_VAL, 1523
WDT_MODE_REG, 1523
WDT_SOFT_RST_EN, 1523
WDT_SOFT_RST_REG, 1523
fdt_reset_syscon.c
fdt_syscon_poweroff, 1528
fdt_syscon_reboot, 1528
poweroff, 1528
reboot, 1529
syscon_do_poweroff, 1526
syscon_do_reboot, 1526
syscon_poweroff, 1529
syscon_poweroff_check, 1526
syscon_poweroff_match, 1529
syscon_reboot, 1529
syscon_reboot_check, 1526
syscon_reboot_match, 1529
syscon_reset_exec, 1527
syscon_reset_get, 1527
syscon_reset_init, 1528
fdt_resize
fdt_sw.c, 1346
libfdt.h, 1408
fdt_resize_property_
fdt_rw.c, 1332
fdt_resv_memory_update_node
fdt_fixup.c, 1153
fdt_ro.c
fdt_find_max_phandle, 1288
fdt_first_property_offset, 1288
fdt_generate_phandle, 1290
fdt_get_alias, 1291
fdt_get_alias_namelen, 1291
fdt_get_mem_rsv, 1292
fdt_get_name, 1292
fdt_get_path, 1293
fdt_get_phandle, 1294
fdt_get_property, 1295
fdt_get_property_by_offset, 1295
fdt_get_property_by_offset_, 1296
fdt_get_property_namelen, 1297
fdt_get_property_namelen_, 1298
fdt_get_string, 1299
fdt_getprop, 1300
fdt_getprop_by_offset, 1301
fdt_getprop_namelen, 1301
fdt_mem_rsv, 1303
fdt_next_property_offset, 1304
fdt_node_check_compatible, 1305
fdt_node_depth, 1306
fdt_node_offset_by_compatible, 1307
fdt_node_offset_by_phandle, 1308
fdt_node_offset_by_prop_value, 1309
fdt_nodename_eq_, 1310
fdt_num_mem_rsv, 1310
fdt_parent_offset, 1311
fdt_path_offset, 1312
fdt_path_offset_namelen, 1313

fdt_string, 1314
fdt_string_eq_, 1315
fdt_stringlist_contains, 1316
fdt_stringlist_count, 1317
fdt_stringlist_get, 1318
fdt_stringlist_search, 1318
fdt_subnode_offset, 1318
fdt_subnode_offset_namelen, 1319
fdt_supernode_atdepth_offset, 1320
nextprop_, 1321
FDT_RO_PROBE
libfdt_internal.h, 1435
fdt_ro_probe_
fdt.c, 1266
libfdt_internal.h, 1442
fdt_rw.c
fdt_add_mem_rsv, 1324
fdt_add_property_, 1324
fdt_add_subnode, 1325
fdt_add_subnode_namelen, 1326
fdt_appendprop, 1326
fdt_blocks_misordered_, 1327
fdt_data_size_, 1327
fdt_del_last_string_, 1328
fdt_del_mem_rsv, 1328
fdt_del_node, 1329
fdt_delprop, 1329
fdt_find_add_string_, 1330
fdt_open_into, 1330
fdt_pack, 1331
fdt_packblocks_, 1331
fdt_resize_property_, 1332
FDT_RW_PROBE, 1324
fdt_rw_probe_, 1332
fdt_set_name, 1333
fdt_setprop, 1333
fdt_setprop_placeholder, 1334
fdt_splice_, 1334
fdt_splice_mem_rsv_, 1335
fdt_splice_string_, 1335
fdt_splice_struct_, 1336
FDT_RW_PROBE
fdt_rw.c, 1324
fdt_rw_probe_
fdt_rw.c, 1332
fdt_serial.c
fdt_serial_drivers, 1535
fdt_serial_init, 1535
fdt_serial.h
fdt_serial_init, 745
fdt_serial_cadence
fdt_serial_cadence.c, 1537
fdt_serial_cadence.c
fdt_serial_cadence, 1537
serial_cadence_init, 1536
serial_cadence_match, 1537
fdt_serial_drivers
fdt_serial.c, 1535
fdt_serial_gaisler
fdt_serial_gaisler.c, 1538
fdt_serial_gaisler.c
fdt_serial_gaisler, 1538
serial_gaisler_init, 1538
serial_gaisler_match, 1538
fdt_serial_htif
fdt_serial_htif.c, 1540
fdt_serial_htif.c
fdt_serial_htif, 1540
serial_htif_init, 1539
serial_htif_match, 1540
fdt_serial_init
fdt_serial.c, 1535
fdt_serial.h, 745
fdt_serial_litex
fdt_serial_litex.c, 1541
fdt_serial_litex.c
fdt_serial_litex, 1541
serial_litex_init, 1541
serial_litex_match, 1541
fdt_serial_renesas_scif
fdt_serial_renesas_scif.c, 1543
fdt_serial_renesas_scif.c
fdt_serial_renesas_scif, 1543
serial_renesas_scif_init, 1542
serial_renesas_scif_match, 1543
fdt_serial_shakti
fdt_serial_shakti.c, 1544
fdt_serial_shakti.c
fdt_serial_shakti, 1544
serial_shakti_init, 1544
serial_shakti_match, 1544
fdt_serial_sifive
fdt_serial_sifive.c, 1546
fdt_serial_sifive.c
fdt_serial_sifive, 1546
serial_sifive_init, 1545
serial_sifive_match, 1546
fdt_serial_uart8250
fdt_serial_uart8250.c, 1547
fdt_serial_uart8250.c
fdt_serial_uart8250, 1547
serial_uart8250_init, 1547
serial_uart8250_match, 1547
fdt_serial_xlnx_uartlite
fdt_serial_xlnx_uartlite.c, 1549
fdt_serial_xlnx_uartlite.c
fdt_serial_xlnx_uartlite, 1549
serial_xlnx_uartlite_init, 1548
serial_xlnx_uartlite_match, 1549
fdt_set_hdr_
libfdt.h, 1361, 1408, 1409
fdt_set_name
fdt_rw.c, 1333
libfdt.h, 1409
fdt_setprop
fdt_rw.c, 1333

libfdt.h, 1410
fdt_setprop_cell
libfdt.h, 1410
fdt_setprop_empty
libfdt.h, 1361
fdt_setprop_inplace
fdt_wip.c, 1351
libfdt.h, 1411
fdt_setprop_inplace_cell
libfdt.h, 1411
fdt_setprop_inplace_namelen_partial
fdt_wip.c, 1351
libfdt.h, 1411
fdt_setprop_inplace_u32
libfdt.h, 1412
fdt_setprop_inplace_u64
libfdt.h, 1412
fdt_setprop_placeholder
fdt_rw.c, 1334
libfdt.h, 1413
fdt_setprop_string
libfdt.h, 1361
fdt_setprop_u32
libfdt.h, 1413
fdt_setprop_u64
libfdt.h, 1414
fdt_size_cells
fdt_addresses.c, 1272
libfdt.h, 1414
fdt_size_dt_strings
libfdt.h, 1361
fdt_size_dt_struct
libfdt.h, 1361
fdt_splice_
fdt_rw.c, 1334
fdt_splice_mem_rsv_
fdt_rw.c, 1335
fdt_splice_string_
fdt_rw.c, 1335
fdt_splice_struct_
fdt_rw.c, 1336
fdt_strerror
fdt_strerror.c, 1338
libfdt.h, 1415
fdt_strerror.c
FDT_ERRTABENT, 1338
fdt_errtable, 1338
FDT_ERRTABSIZE, 1338
fdt_strerror, 1338
fdt_string
fdt_ro.c, 1314
libfdt.h, 1415
fdt_string_eq_
fdt_ro.c, 1315
fdt_stringlist_contains
fdt_ro.c, 1316
libfdt.h, 1416
fdt_stringlist_count
fdt_ro.c, 1317
libfdt.h, 1417
fdt_stringlist_get
fdt_ro.c, 1318
libfdt.h, 1418
fdt_stringlist_search
fdt_ro.c, 1318
libfdt.h, 1418
fdt_subnode_offset
fdt_ro.c, 1318
libfdt.h, 1418
fdt_subnode_offset_namelen
fdt_ro.c, 1319
libfdt.h, 1419
fdt_supernode_atdepth_offset
fdt_ro.c, 1320
libfdt.h, 1420
fdt_suspend.c
fdt_suspend_drivers, 1583
fdt_suspend_init, 1583
fdt_suspend.h
fdt_suspend_init, 760
fdt_suspend_drivers
fdt_suspend.c, 1583
fdt_suspend_init
fdt_suspend.c, 1583
fdt_suspend.h, 760
fdt_suspend_rpmi
fdt_suspend_rpmi.c, 1586
fdt_suspend_rpmi.c
fdt_suspend_rpmi, 1586
rpmi_suspend_dev, 1586
rpmi_suspend_init, 1584
rpmi_suspend_match, 1586
rpmi_syssusp, 1585
rpmi_syssusp_attrs, 1585
rpmi_system_suspend, 1586
rpmi_system_suspend_check, 1586
syssusp_ctx, 1586
fdt_sw.c
fdt_add_reservemap_entry, 1340
fdt_add_string_, 1341
fdt_begin_node, 1341
fdt_create, 1341
fdt_create_with_flags, 1342
fdt_del_last_string_, 1342
fdt_end_node, 1343
fdt_find_add_string_, 1343
fdt_finish, 1344
fdt_finish_reservemap, 1344
fdt_grab_space_, 1345
fdt_property, 1345
fdt_property_placeholder, 1346
fdt_resize, 1346
FDT_SW_PROBE, 1340
fdt_sw_probe_, 1346
FDT_SW_PROBE_MEMRSV, 1340
fdt_sw_probe_memrsv_, 1347

FDT_SW_PROBE_STRUCT, 1340
fdt_sw_probe_struct_, 1347
sw_flags, 1347
FDT_SW_MAGIC
libfdt_internal.h, 1435
FDT_SW_PROBE
fdt_sw.c, 1340
fdt_sw_probe_
fdt_sw.c, 1346
FDT_SW_PROBE_MEMRSV
fdt_sw.c, 1340
fdt_sw_probe_memrsv_
fdt_sw.c, 1347
FDT_SW_PROBE_STRUCT
fdt_sw.c, 1340
fdt_sw_probe_struct_
fdt_sw.c, 1347
fdt_syscon_poweroff
fdt_reset_syscon.c, 1528
fdt_syscon_reboot
fdt_reset_syscon.c, 1528
FDT_TAGALIGN
libfdt_internal.h, 1435
FDT_TAGSIZE
fdt.h, 1268
fdt_timer.c
fdt_timer_drivers, 1604
fdt_timer_init, 1604
fdt_timer.h
fdt_timer_init, 774
fdt_timer_drivers
fdt_timer.c, 1604
fdt_timer_init
fdt_timer.c, 1604
fdt_timer.h, 774
fdt_timer_mtimer
fdt_timer_mtimer.c, 1606
fdt_timer_mtimer.c
fdt_timer_mtimer, 1606
mt_reference, 1606
SBI_LIST_HEAD, 1605
sifive_clint_quirks, 1606
thead_aclint_quirks, 1606
thead_clint_quirks, 1606
timer_mtimer_cold_init, 1605
timer_mtimer_match, 1606
fdt_timer_plmt
fdt_timer_plmt.c, 1608
fdt_timer_plmt.c
fdt_plmt_cold_timer_init, 1607
fdt_timer_plmt, 1608
plmt, 1608
timer_plmt_match, 1608
fdt_totalsize
libfdt.h, 1361
fdt_translate_address
fdt_helper.c, 1174
fdt_unregister_general_fixup
fdt_fixup.c, 1153
fdt_fixup.h, 642
FDT_V16_SIZE
fdt.h, 1268
FDT_V17_SIZE
fdt.h, 1268
FDT_V1_SIZE
fdt.h, 1268
FDT_V2_SIZE
fdt.h, 1268
FDT_V3_SIZE
fdt.h, 1268
fdt_version
libfdt.h, 1361
fdt_wip.c
fdt_node_end_offset_, 1349
fdt_nop_node, 1349
fdt_nop_property, 1350
fdt_nop_region_, 1350
fdt_setprop_inplace, 1351
fdt_setprop_inplace_namelen_partial, 1351
features
sbi_fwft.c, 924
sbi_platform, 142
final_exit
sbi_platform_operations, 144
final_init
sbi_platform_operations, 144
find_first_bit
sbi_bitops.c, 798
sbi_bitops.h, 317
find_first_zero_bit
sbi_bitops.c, 798
sbi_bitops.h, 318
find_last_bit
sbi_bitops.c, 798
sbi_bitops.h, 318
find_next_bit
sbi_bitops.c, 799
sbi_bitops.h, 318
find_next_subset_region
sbi_domain.c, 840
find_next_zero_bit
sbi_bitops.c, 799
sbi_bitops.h, 319
find_region
sbi_domain.c, 840
firmware_context
sbi_platform, 142
first_hartid
aclint_mswi_data, 77
aclint_mtimer_data, 78
first_irq
aplic_delegate_data, 80
flags
gpio_pin, 91
mbox_xfer, 96
plic_data, 98

rpmi_clock_get_attributes_resp, 102
rpmi_clock_get_supported_rates_resp, 105
rpmi_clock_set_rate_req, 106
rpmi_cppc_get_fastchan_region_resp, 108
rpmi_hsm_get_susp_info_resp, 115
rpmi_message_args, 119
rpmi_message_header, 120
rpmi_sysrst_get_reset_attributes_resp, 120
rpmi_syssusp_get_attr_resp, 121
sbi_domain_memregion, 131
FLASH_BASE
atcsmu.h, 762
for_each_clear_bit
sbi_bitops.h, 314
for_each_clear_bit_from
sbi_bitops.h, 314
for_each_set_bit
sbi_bitops.h, 314
for_each_set_bit_from
sbi_bitops.h, 315
for_each_trig_entry
sbi_dbtr.c, 827
format
sbi_console.h, 333
fp_init
sbi_hart.c, 931
free_chan
mbox_controller, 94
freq
platform_uart_data, 97
FROMHOST_CMD
htif.c, 1590
FROMHOST_DATA
htif.c, 1590
FROMHOST_DEV
htif.c, 1590
fw_counter_match_encoding
sbi_pmu_device, 146
fw_counter_read_value
sbi_pmu_device, 146
fw_counter_start
sbi_pmu_device, 146
fw_counter_stop
sbi_pmu_device, 146
fw_counter_width
sbi_pmu_device, 146
fw_counter_write_value
sbi_pmu_device, 146
fw_dynamic.h
__packed, 164
FW_DYNAMIC_INFO_BOOT_HART_OFFSET, 163
FW_DYNAMIC_INFO_MAGIC_OFFSET, 163
FW_DYNAMIC_INFO_MAGIC_VALUE, 164
FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET, 164
FW_DYNAMIC_INFO_NEXT_MODE_M, 164
FW_DYNAMIC_INFO_NEXT_MODE_OFFSET, 164
FW_DYNAMIC_INFO_NEXT_MODE_S, 164
FW_DYNAMIC_INFO_NEXT_MODE_U, 164
FW_DYNAMIC_INFO_OPTIONS_OFFSET, 164
FW_DYNAMIC_INFO_VERSION_2, 164
FW_DYNAMIC_INFO_VERSION_MAX, 164
FW_DYNAMIC_INFO_VERSION_OFFSET, 164
fw_dynamic_info, 88
boot_hart, 89
magic, 89
next_addr, 89
next_mode, 89
options, 89
version, 89
FW_DYNAMIC_INFO_BOOT_HART_OFFSET
fw_dynamic.h, 163
FW_DYNAMIC_INFO_MAGIC_OFFSET
fw_dynamic.h, 163
FW_DYNAMIC_INFO_MAGIC_VALUE
fw_dynamic.h, 164
FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET
fw_dynamic.h, 164
FW_DYNAMIC_INFO_NEXT_MODE_M
fw_dynamic.h, 164
FW_DYNAMIC_INFO_NEXT_MODE_OFFSET
fw_dynamic.h, 164
FW_DYNAMIC_INFO_NEXT_MODE_S
fw_dynamic.h, 164
FW_DYNAMIC_INFO_NEXT_MODE_U
fw_dynamic.h, 164
FW_DYNAMIC_INFO_OPTIONS_OFFSET
fw_dynamic.h, 164
FW_DYNAMIC_INFO_VERSION_2
fw_dynamic.h, 164
FW_DYNAMIC_INFO_VERSION_MAX
fw_dynamic.h, 164
FW_DYNAMIC_INFO_VERSION_OFFSET
fw_dynamic.h, 164
fw_event_validate_encoding
sbi_pmu_device, 147
fw_heap_offset
sbi_scratch, 148
fw_heap_size
sbi_scratch, 148
fw_region_inited
sbi_domain, 129
fw_rw_offset
sbi_scratch, 148
fw_size
sbi_scratch, 148
fw_start
sbi_scratch, 148
fwft_adue_supported
sbi_fwft.c, 915
fwft_clear_config_lock
sbi_fwft.c, 916
fwft_defined_features

sbi_fwft.c, 924
fwft_double_trap_supported
sbi_fwft.c, 916
fwft_enable_lpad
sbi_fwft.c, 916
fwft_enable_sstack
sbi_fwft.c, 917
fwft_get_adue
sbi_fwft.c, 917
fwft_get_double_trap
sbi_fwft.c, 917
fwft_get_feature
sbi_fwft.c, 918
fwft_get_hart_state_ptr
sbi_fwft.c, 915
fwft_get_lpad
sbi_fwft.c, 918
fwft_get_misaligned_delegation
sbi_fwft.c, 919
fwft_get_sstack
sbi_fwft.c, 919
fwft_is_defined_feature
sbi_fwft.c, 919
fwft_lpad_supported
sbi_fwft.c, 919
fwft_menvcfg_read_bit
sbi_fwft.c, 920
fwft_menvcfg_set_bit
sbi_fwft.c, 920
fwft_misaligned_delegation_supported
sbi_fwft.c, 921
fwft_ptr_offset
sbi_fwft.c, 924
fwft_set_adue
sbi_fwft.c, 921
fwft_set_double_trap
sbi_fwft.c, 921
fwft_set_hart_state_ptr
sbi_fwft.c, 915
fwft_set_misaligned_delegation
sbi_fwft.c, 922
fwft_sstack_supported
sbi_fwft.c, 922
fwft_thishart_state_ptr
sbi_fwft.c, 915

gaisler-uart.c
gaisler_console, 1553
gaisler_uart_getc, 1551
gaisler_uart_init, 1551
gaisler_uart_putc, 1551
get_reg, 1552
set_reg, 1552
uart_base, 1553
UART_CTRL_DB, 1550
UART_CTRL_FL, 1550
UART_CTRL_RE, 1550
UART_CTRL_TE, 1550
UART_DATA_DATA, 1550
UART_REG_CTRL, 1550
UART_REG_DATA, 1550
UART_REG_SCALER, 1550
UART_REG_STATUS, 1550
UART_STATUS_DATAREADY, 1550
UART_STATUS_FIFOFULL, 1550
gaisler-uart.h
gaisler_uart_init, 746
gaisler_console
gaisler-uart.c, 1553
gaisler_uart_getc
gaisler-uart.c, 1551
gaisler_uart_init
gaisler-uart.c, 1551
gaisler-uart.h, 746
gaisler_uart_putc
gaisler-uart.c, 1551
Generic Platform, 45
GENMASK
sbi_bitops.h, 315
GENMASK_ULL
sbi_bitops.h, 315
get
gpio_chip, 90
get_attribute
mbox_controller, 94
get_cidx_code
sbi_pmu.c, 1011
get_cidx_type
sbi_pmu.c, 1011
GET_CSR_NUM
riscv_encoding.h, 245
get_direction
gpio_chip, 90
GET_DLEN
fdt_mailbox_rpmi_shmem.c, 1464
GET_EMUL
riscv_encoding.h, 245
get_feature_config
sbi_fwft.c, 922
GET_FLAGS
fdt_mailbox_rpmi_shmem.c, 1464
GET_LEN
riscv_encoding.h, 245
GET_MESSAGE_ID
fdt_mailbox_rpmi_shmem.c, 1465
GET_MESSAGE_TYPE
fdt_mailbox_rpmi_shmem.c, 1465
GET_MEW
riscv_encoding.h, 245
GET_NF
riscv_encoding.h, 245
get_notification_events
sbi_mpxy_channel, 139
get_path_len
fdt_overlay.c, 1278
GET_PRECISION
riscv_fp.h, 292

get_reg
cadence-uart.c, 1533
gaisler-uart.c, 1552
litex-uart.c, 1554
renesas_scif.c, 1559
sifive-uart.c, 1571
uart8250.c, 1577
GET_RM
riscv_encoding.h, 245
GET_RS1
riscv_encoding.h, 246
GET_RS1_NUM
riscv_encoding.h, 246
GET_RS1S
riscv_encoding.h, 246
GET_RS2
riscv_encoding.h, 246
GET_RS2C
riscv_encoding.h, 246
GET_RS2S
riscv_encoding.h, 246
GET_SERVICE_ID
fdt_mailbox_rpmi_shmem.c, 1465
GET_SERVICEGROUP_ID
fdt_mailbox_rpmi_shmem.c, 1465
GET_SP
riscv_encoding.h, 246
get_ticks
sbi_timer.c, 1076
get_time_val
sbi_timer.c, 1082
get_tlb_num_entries
sbi_platform_operations, 144
get_tlbr_flush_limit
sbi_platform_operations, 145
GET_TOKEN
fdt_mailbox_rpmi_shmem.c, 1465
GET_VD
riscv_encoding.h, 246
GET_VEMUL
riscv_encoding.h, 246
GET_VIEW
riscv_encoding.h, 246
GET_VLMUL
riscv_encoding.h, 246
GET_VS2
riscv_encoding.h, 247
GET_VSEW
riscv_encoding.h, 247
global_event_count
sbi_sse.c, 1063
global_events
sbi_sse.c, 1063
global_hpctrl
sbi_heap.c, 953
sbi_heap.h, 464
gp
sbi_trap_regs, 156
gpio.c
gpio_chip_add, 1188
gpio_chip_find, 1189
gpio_chip_remove, 1189
gpio_direction_input, 1190
gpio_direction_output, 1190
gpio_get, 1190
gpio_get_direction, 1190
gpio_set, 1190
SBI_LIST_HEAD, 1191
gpio.h
gpio_chip_add, 669
gpio_chip_find, 669
gpio_chip_remove, 670
gpio_direction_input, 670
gpio_direction_output, 670
GPIO_FLAG_ACTIVE_LOW, 668
GPIO_FLAG_OPEN_DRAIN, 668
GPIO_FLAG_PULL_DOWN, 668
GPIO_FLAG_PULL_UP, 668
GPIO_FLAG_SINGLE_ENDED, 668
GPIO_FLAG_TRANSITORY, 668
gpio_get, 671
gpio_get_direction, 671
GPIO_LINE_DIRECTION_IN, 668
GPIO_LINE_DIRECTION_OUT, 668
gpio_set, 671
to_gpio_chip, 671
gpio_chip, 89
direction_input, 90
direction_output, 90
driver, 90
get, 90
get_direction, 90
id, 90
ngpio, 90
node, 90
set, 90
gpio_chip_add
gpio.c, 1188
gpio.h, 669
gpio_chip_find
gpio.c, 1189
gpio.h, 669
gpio_chip_remove
gpio.c, 1189
gpio.h, 670
gpio_direction_input
gpio.c, 1190
gpio.h, 670
gpio_direction_output
gpio.c, 1190
gpio.h, 670
GPIO_FLAG_ACTIVE_LOW
gpio.h, 668
GPIO_FLAG_OPEN_DRAIN
gpio.h, 668
GPIO_FLAG_PULL_DOWN

gpio.h, 668
GPIO_FLAG_PULL_UP
gpio.h, 668
GPIO_FLAG_SINGLE_ENDED
gpio.h, 668
GPIO_FLAG_TRANSITORY
gpio.h, 668
gpio_get
gpio.c, 1190
gpio.h, 671
gpio_get_direction
gpio.c, 1190
gpio.h, 671
GPIO_LINE_DIRECTION_IN
gpio.h, 668
GPIO_LINE_DIRECTION_OUT
gpio.h, 668
gpio_pin, 91
chip, 91
flags, 91
offset, 91
gpio_poweroff
fdt_reset_gpio.c, 1514
gpio_poweroff_match
fdt_reset_gpio.c, 1514
gpio_reset_exec
fdt_reset_gpio.c, 1511
gpio_reset_get
fdt_reset_gpio.c, 1511
gpio_reset_init
fdt_reset_gpio.c, 1512
gpio_reset_match
fdt_reset_gpio.c, 1514
gpio_restart
fdt_reset_gpio.c, 1514
gpio_set
gpio.c, 1190
gpio.h, 671
gpio_system_poweroff
fdt_reset_gpio.c, 1512
gpio_system_poweroff_check
fdt_reset_gpio.c, 1512
gpio_system_restart
fdt_reset_gpio.c, 1513
gpio_system_restart_check
fdt_reset_gpio.c, 1513
group_index_bits
imsic_data, 92
group_index_shift
imsic_data, 92
guest_index_bits
imsic_data, 92
gva
sbi_trap_info, 154

H
quad.h, 1454
HALF_BITS
quad.h, 1454
handle
sbi_ecall_extension, 132
hart_context_get
sbi_domain_context.c, 857
hart_context_set
sbi_domain_context.c, 858
hart_context_thishart_get
sbi_domain_context.c, 857
hart_count
aclint_mswi_data, 77
aclint_mtimer_data, 78
plicsw_data, 98
plmt_data, 99
sbi_platform, 142
hart_data_offset
sbi_hsm.c, 966
hart_detect_features
sbi_hart.c, 931
hart_features_offset
sbi_hart.c, 944
hart_id
rpmi_cppc_get_fastchan_offset_req, 107
rpmi_cppc_probe_req, 110
rpmi_cppc_read_reg_req, 110
rpmi_cppc_write_reg_req, 111
hart_index2id
sbi_platform, 142
hart_index_bits
imsic_data, 93
hart_mhpm_get_allowed_bits
sbi_hart.c, 932
hart_pmp_get_allowed_addr
sbi_hart.c, 932
hart_resume
sbi_hsm_device, 135
hart_shmem_base
sbi_dbtr.c, 830
sbi_mpxy.c, 999
hart_stack_size
sbi_platform, 142
hart_start
sbi_hsm_device, 135
hart_state_ptr_offset
sbi_dbtr.c, 838
hart_status
rpmi_hsm_get_hart_status_resp, 114
hart_stop
sbi_hsm_device, 136
hart_suspend
sbi_hsm_device, 136
hartid
rpmi_cppc_hart_list_resp, 109
rpmi_hsm_get_hart_list_resp, 113
rpmi_hsm_get_hart_status_req, 114
rpmi_hsm_hart_start_req, 116
rpmi_hsm_hart_stop_req, 117
rpmi_hsm_hart_susp_req, 117
rpmi_syssusp_suspend_req, 122

sbi_dbtr_hart_triggers_state, 125
hartid2scratch
sbi_scratch.c, 1029
hartid_to_scratch
sbi_scratch, 148
hartindex
sbi_scratch, 149
hartindex_to_hartid_table
sbi_scratch.c, 1031
sbi_scratch.h, 562
hartindex_to_scratch_table
sbi_scratch.c, 1031
sbi_scratch.h, 562
HARTn_RESET_VEC_HI
atcsmu.h, 762
HARTn_RESET_VEC_LO
atcsmu.h, 762
has_64bit_mmio
aclint_mtimer_data, 78
has_msicfg_mmode
aplic_data, 79
has_msicfg_smode
aplic_data, 79
has_shared_mtime
aclint_mtimer_data, 78
head
fdt_general_fixup, 83
sbi_domain_data, 130
sbi_ecall_extension, 132
sbi_mpxy_channel, 139
header
rpmi_message, 118
HEAP_ALLOC_ALIGN
sbi_heap.c, 946
HEAP_BASE_ALIGN
sbi_heap.h, 453
HEAP_HOUSEKEEPING_FACTOR
sbi_heap.c, 946
heap_size
sbi_platform, 142
HGATP32_MODE_SHIFT
riscv_encoding.h, 247
HGATP32_PPN
riscv_encoding.h, 247
HGATP32_VMID_MASK
riscv_encoding.h, 247
HGATP32_VMID_SHIFT
riscv_encoding.h, 247
HGATP64_MODE_SHIFT
riscv_encoding.h, 247
HGATP64_PPN
riscv_encoding.h, 247
HGATP64_VMID_MASK
riscv_encoding.h, 247
HGATP64_VMID_SHIFT
riscv_encoding.h, 247
HGATP_MODE_OFF
riscv_encoding.h, 247
HGATP_MODE_SHIFT
riscv_encoding.h, 247
HGATP_MODE_SV32X4
riscv_encoding.h, 247
HGATP_MODE_SV39X4
riscv_encoding.h, 247
HGATP_MODE_SV48X4
riscv_encoding.h, 248
HGATP_PPN
riscv_encoding.h, 248
HGATP_VMID_MASK
riscv_encoding.h, 248
HGATP_VMID_SHIFT
riscv_encoding.h, 248
HHALF
quad.h, 1454
hhxs
aplic_msicfg_data, 81
hhxw
aplic_msicfg_data, 81
hpm_allowed
sbi_emulate_csr.c, 903
hsm_dev
sbi_hsm.c, 966
hsm_device_hart_resume
sbi_hsm.c, 956
hsm_device_hart_start
sbi_hsm.c, 957
hsm_device_hart_stop
sbi_hsm.c, 957
hsm_device_hart_suspend
sbi_hsm.c, 957
hsm_device_has_hart_hotplug
sbi_hsm.c, 957
hsm_device_has_hart_secondary_boot
sbi_hsm.c, 958
hsm_start_ticket_acquire
sbi_hsm.c, 958
hsm_start_ticket_release
sbi_hsm.c, 958
HSTATUS_GVA
riscv_encoding.h, 248
HSTATUS_HU
riscv_encoding.h, 248
HSTATUS_SPV
riscv_encoding.h, 248
HSTATUS_SPVP
riscv_encoding.h, 248
HSTATUS_VGEIN
riscv_encoding.h, 248
HSTATUS_VGEIN_SHIFT
riscv_encoding.h, 248
HSTATUS_VSBE
riscv_encoding.h, 248
HSTATUS_VTSR
riscv_encoding.h, 248
HSTATUS_VTVM
riscv_encoding.h, 248

HSTATUS_VTW
riscv_encoding.h, 248
htif.c
__attribute__, 1591
__check_fromhost, 1591
__read_fromhost, 1592
__read_tohost, 1592
__set_tohost, 1592
__write_fromhost, 1593
__write_tohost, 1593
FROMHOST_CMD, 1590
FROMHOST_DATA, 1590
FROMHOST_DEV, 1590
HTIF_CMD_BITS, 1590
HTIF_CMD_MASK, 1590
HTIF_CMD_SHIFT, 1590
htif_console, 1596
htif_console_buf, 1596
HTIF_CONSOLE_CMD_GETC, 1590
HTIF_CONSOLE_CMD_PUTC, 1590
htif_custom, 1596
HTIF_DATA_BITS, 1590
HTIF_DATA_MASK, 1590
HTIF_DATA_SHIFT, 1590
HTIF_DEV_BITS, 1590
HTIF_DEV_CONSOLE, 1590
HTIF_DEV_MASK, 1590
HTIF_DEV_SHIFT, 1590
HTIF_DEV_SYSTEM, 1591
htif_fromhost, 1596
htif_getc, 1593
htif_lock, 1596
htif_putc, 1594
htif_reset, 1596
htif_serial_init, 1594
htif_system_reset, 1595
htif_system_reset_check, 1595
htif_system_reset_init, 1595
htif_tohost, 1597
PK_SYS_write, 1591
set_custom_addr, 1596
TOHOST_CMD, 1591
htif.h
htif_serial_init, 767
htif_system_reset_init, 767
HTIF_CMD_BITS
htif.c, 1590
HTIF_CMD_MASK
htif.c, 1590
HTIF_CMD_SHIFT
htif.c, 1590
htif_console
htif.c, 1596
htif_console_buf
htif.c, 1596
HTIF_CONSOLE_CMD_GETC
htif.c, 1590
HTIF_CONSOLE_CMD_PUTC
htif.c, 1590
htif_custom
htif.c, 1596
HTIF_DATA_BITS
htif.c, 1590
HTIF_DATA_MASK
htif.c, 1590
HTIF_DATA_SHIFT
htif.c, 1590
HTIF_DEV_BITS
htif.c, 1590
HTIF_DEV_CONSOLE
htif.c, 1590
HTIF_DEV_MASK
htif.c, 1590
HTIF_DEV_SHIFT
htif.c, 1590
HTIF_DEV_SYSTEM
htif.c, 1591
htif_fromhost
htif.c, 1596
htif_getc
htif.c, 1593
htif_lock
htif.c, 1596
htif_putc
htif.c, 1594
htif_reset
htif.c, 1596
htif_reset_init
fdt_reset_htif.c, 1515
htif_reset_match
fdt_reset_htif.c, 1516
htif_serial_init
htif.c, 1594
htif.h, 767
htif_system_reset
htif.c, 1595
htif_system_reset_check
htif.c, 1595
htif_system_reset_init
htif.c, 1595
htif.h, 767
htif_tohost
htif.c, 1597
hw_counter_disable_irq
sbi_pmu_device, 147
hw_counter_enable_irq
sbi_pmu_device, 147
hw_counter_filter_mode
sbi_pmu_device, 147
hw_counter_irq_bit
sbi_pmu_device, 147
hw_event_count
fdt_pmu.c, 1177
hw_event_map
sbi_pmu.c, 1027

i2c.c

i2c_adapter_add, 1218
i2c_adapter_find, 1219
i2c_adapter_read, 1219
i2c_adapter_remove, 1220
i2c_adapter_write, 1220
SBI_LIST_HEAD, 1220
i2c.h
i2c_adapter_add, 679
i2c_adapter_find, 679
i2c_adapter_read, 680
i2c_adapter_reg_read, 680
i2c_adapter_reg_write, 680
i2c_adapter_remove, 681
i2c_adapter_write, 681
to_i2c_adapter, 682
i2c_adapter, 91
id, 92
node, 92
read, 92
write, 92
i2c_adapter_add
i2c.c, 1218
i2c.h, 679
i2c_adapter_find
i2c.c, 1219
i2c.h, 679
i2c_adapter_read
i2c.c, 1219
i2c.h, 680
i2c_adapter_reg_read
i2c.h, 680
i2c_adapter_reg_write
i2c.h, 680
i2c_adapter_remove
i2c.c, 1220
i2c.h, 681
i2c_adapter_write
i2c.c, 1220
i2c.h, 681
IC_DATA_CMD_READ
dw_i2c.c, 1203
IC_DATA_CMD_RESTART
dw_i2c.c, 1203
IC_DATA_CMD_STOP
dw_i2c.c, 1203
IC_INT_STATUS_STOPDET
dw_i2c.c, 1203
id
gpio_chip, 90
i2c_adapter, 92
mbox_controller, 95
regmap, 100
sbi_dbtr_shmem_entry, 127
sbi_hart_ext_data, 134
idx
sbi_dbtr_id_msg, 126
idx_to_data_ptr
sbi_domain_data_priv, 131
illegal_insn_func
sbi_illegal_insn.c, 967
illegal_insn_table
sbi_illegal_insn.c, 969
IMM_I
riscv_encoding.h, 248
IMM_S
riscv_encoding.h, 249
imsic.c
imsic_cold_irqchip_init, 1246
imsic_csr_clear, 1244
imsic_csr_read, 1244
imsic_csr_set, 1244
imsic_csr_write, 1244
imsic_data_check, 1247
imsic_device, 1249
IMSIC_DISABLE_EIDELIVERY, 1244
IMSIC_DISABLE_EITHRESHOLD, 1244
IMSIC_EIDELIVERY, 1245
IMSIC_EIE0, 1245
IMSIC_EIE63, 1245
IMSIC_EIEx_BITS, 1245
IMSIC_EIP0, 1245
IMSIC_EIP63, 1245
IMSIC_EIPx_BITS, 1245
IMSIC_EITHRESHOLD, 1245
IMSIC_ENABLE_EIDELIVERY, 1245
IMSIC_ENABLE_EITHRESHOLD, 1245
imsic_external_irqfn, 1247
imsic_file_offset, 1249
imsic_get_data, 1247
imsic_get_hart_data_ptr, 1245
imsic_get_hart_file, 1245
imsic_get_target_file, 1247
imsic_ipi_device, 1249
IMSIC_IPI_ID, 1245
imsic_ipi_send, 1248
imsic_local_eix_update, 1248
imsic_local_irqchip_init, 1248
imsic_map_hartid_to_data, 1248
IMSIC_MAX_ID, 1245
IMSIC_MIN_ID, 1245
IMSIC_MMIO_PAGE_BE, 1246
IMSIC_MMIO_PAGE_LE, 1246
imsic_ptr_offset, 1249
imsic_set_hart_data_ptr, 1246
imsic_set_hart_file, 1246
IMSIC_TOPEI, 1246
IMSIC_TOPEI_ID_MASK, 1246
IMSIC_TOPEI_ID_SHIFT, 1246
IMSIC_TOPEI_PRIO_MASK, 1246
imsic_warm_irqchip_init, 1249
imsic.h
imsic_data_check, 694
imsic_local_irqchip_init, 694
IMSIC_MAX_REGS, 694
IMSIC_MMIO_PAGE_SHIFT, 694
IMSIC_MMIO_PAGE_SZ, 694

imsic_cold_irqchip_init
imsic.c, 1246
imsic_csr_clear
imsic.c, 1244
imsic_csr_read
imsic.c, 1244
imsic_csr_set
imsic.c, 1244
imsic_csr_write
imsic.c, 1244
imsic_data, 92
group_index_bits, 92
group_index_shift, 92
guest_index_bits, 92
hart_index_bits, 93
num_ids, 93
regs, 93
targets_mmode, 93
imsic_data_check
imsic.c, 1247
imsic.h, 694
imsic_device
imsic.c, 1249
IMSIC_DISABLE_EIDELIVERY
imsic.c, 1244
IMSIC_DISABLE_EITHRESHOLD
imsic.c, 1244
IMSIC_EIDELIVERY
imsic.c, 1245
IMSIC_EIE0
imsic.c, 1245
IMSIC_EIE63
imsic.c, 1245
IMSIC_EIEx_BITS
imsic.c, 1245
IMSIC_EIP0
imsic.c, 1245
IMSIC_EIP63
imsic.c, 1245
IMSIC_EIPx_BITS
imsic.c, 1245
IMSIC_EITHRESHOLD
imsic.c, 1245
IMSIC_ENABLE_EIDELIVERY
imsic.c, 1245
IMSIC_ENABLE_EITHRESHOLD
imsic.c, 1245
imsic_external_irqfn
imsic.c, 1247
imsic_file_offset
imsic.c, 1249
imsic_get_data
imsic.c, 1247
imsic_get_hart_data_ptr
imsic.c, 1245
imsic_get_hart_file
imsic.c, 1245
imsic_get_target_file
imsic.c, 1247
imsic_ipi_device
imsic.c, 1249
IMSIC_IPI_ID
imsic.c, 1245
imsic_ipi_send
imsic.c, 1248
imsic_local_eix_update
imsic.c, 1248
imsic_local_irqchip_init
imsic.c, 1248
imsic.h, 694
imsic_map_hartid_to_data
imsic.c, 1248
IMSIC_MAX_ID
imsic.c, 1245
IMSIC_MAX_REGS
imsic.h, 694
IMSIC_MIN_ID
imsic.c, 1245
IMSIC_MMIO_PAGE_BE
imsic.c, 1246
IMSIC_MMIO_PAGE_LE
imsic.c, 1246
IMSIC_MMIO_PAGE_SHIFT
imsic.h, 694
IMSIC_MMIO_PAGE_SZ
imsic.h, 694
imsic_ptr_offset
imsic.c, 1249
imsic_regs, 93
addr, 93
size, 93
imsic_set_hart_data_ptr
imsic.c, 1246
imsic_set_hart_file
imsic.c, 1246
IMSIC_TOPEI
imsic.c, 1246
IMSIC_TOPEI_ID_MASK
imsic.c, 1246
IMSIC_TOPEI_ID_SHIFT
imsic.c, 1246
IMSIC_TOPEI_PRIO_MASK
imsic.c, 1246
imsic_warm_irqchip_init
imsic.c, 1249
include/sbi/fw_dynamic.h, 162, 165
include/sbi/riscv_asm.h, 166, 176
include/sbi/riscv_atomic.h, 178, 184
include/sbi/riscv_barrier.h, 185, 186
include/sbi/riscv_dbtr.h, 187, 194
include/sbi/riscv_elf.h, 197
include/sbi/riscv_encoding.h, 197, 275
include/sbi/riscv_fp.h, 291, 292
include/sbi/riscv_io.h, 294, 298
include/sbi/riscv_locks.h, 299, 303
include/sbi/sbi_bitmap.h, 304, 311

include/sbi/sbi_bitops.h, 312, 320
include/sbi/sbi_byteorder.h, 323, 325
include/sbi/sbi_console.h, 326, 333
include/sbi/sbi_const.h, 334, 335
include/sbi/sbi_cppc.h, 336, 339
include/sbi/sbi_csr_detect.h, 339, 341
include/sbi/sbi_dbtr.h, 342, 350
include/sbi/sbi_domain.h, 351, 367
include/sbi/sbi_domain_context.h, 370, 373
include/sbi/sbi_domain_data.h, 373, 377
include/sbi/sbi_ecall.h, 377, 383
include/sbi/sbi_ecall_interface.h, 385, 407
include/sbi/sbi_emulate_csr.h, 412, 414
include/sbi/sbi_error.h, 415, 417
include/sbi/sbi_fifo.h, 418, 424
include/sbi/sbi_fwft.h, 425, 427
include/sbi/sbi_hart.h, 427, 440
include/sbi/sbi_hartmask.h, 441, 451
include/sbi/sbi_heap.h, 452, 464
include/sbi/sbi_hfence.h, 466, 468
include/sbi/sbi_hsm.h, 468, 478
include/sbi/sbi_illegal_insn.h, 479, 480
include/sbi/sbi_init.h, 481, 485
include/sbi/sbi_ipi.h, 485, 492
include/sbi/sbi_irqchip.h, 493, 496
include/sbi/sbi_list.h, 497, 507
include/sbi/sbi_math.h, 509, 510
include/sbi/sbi_mpxy.h, 511, 518
include/sbi/sbi_platform.h, 520, 538
include/sbi/sbi_pmu.h, 543, 553
include/sbi/sbi_scratch.h, 555, 562
include/sbi/sbi_sse.h, 565, 575
include/sbi/sbi_string.h, 576, 581
include/sbi/sbi_system.h, 582, 588
include/sbi/sbi_timer.h, 589, 596
include/sbi/sbi_tlb.h, 596, 600
include/sbi/sbi_trap.h, 601, 610
include/sbi/sbi_trap_ldst.h, 612, 617
include/sbi/sbi_types.h, 618, 622
include/sbi/sbi_unit_test.h, 624
include/sbi/sbi_unpriv.h, 625, 627
include/sbi/sbi_version.h, 627, 628
include/sbi_utils/cppc/fdt_cppc.h, 628, 629
include/sbi_utils/fdt/fdt_domain.h, 630, 631
include/sbi_utils/fdt/fdt_driver.h, 632, 635
include/sbi_utils/fdt/fdt_fixup.h, 636, 643
include/sbi_utils/fdt/fdt_helper.h, 643, 661
include/sbi_utils/fdt/fdt_pmu.h, 663, 664
include/sbi_utils/gpio/fdt_gpio.h, 665, 666
include/sbi_utils/gpio/gpio.h, 667, 672
include/sbi_utils/hsm/fdt_hsm.h, 672, 673
include/sbi_utils/i2c/dw_i2c.h, 674, 676
include/sbi_utils/i2c/fdt_i2c.h, 676, 677
include/sbi_utils/i2c/i2c.h, 678, 682
include/sbi_utils/ipi/aclint_mswi.h, 683, 685
include/sbi_utils/ipi/andes_plicsw.h, 685, 687
include/sbi_utils/ipi/fdt_ipi.h, 687, 688
include/sbi_utils/irqchip/aplic.h, 689, 691
include/sbi_utils/irqchip/fdt_irqchip.h, 692
include/sbi_utils/irqchip/imsic.h, 693, 694
include/sbi_utils/irqchip/plic.h, 695, 698
include/sbi_utils/mailbox/fdt_mailbox.h, 699, 701
include/sbi_utils/mailbox/mailbox.h, 701, 708
include/sbi_utils/mailbox/rpmi_mailbox.h, 710, 713
include/sbi_utils/mailbox/rpmi_msgprot.h, 714, 726
include/sbi_utils/mpxy/fdt_mpxy.h, 732, 733
include/sbi_utils/regmap/fdt_regmap.h, 733, 735
include/sbi_utils/regmap/regmap.h, 736, 740
include/sbi_utils/reset/fdt_reset.h, 741, 742
include/sbi_utils/serial/cadence-uart.h, 743, 744
include/sbi_utils/serial/fdt_serial.h, 744, 745
include/sbi_utils/serial/gaisler-uart.h, 746, 747
include/sbi_utils/serial/litex-uart.h, 748, 749
include/sbi_utils/serial/renesas-scif.h, 750, 751
include/sbi_utils/serial/semihosting.h, 751, 752
include/sbi_utils/serial/shakti-uart.h, 753, 754
include/sbi_utils/serial/sifive-uart.h, 754, 755
include/sbi_utils/serial/uart8250.h, 756, 757
include/sbi_utils/serial/xlnx_uartlite.h, 758, 759
include/sbi_utils/suspend/fdt_suspend.h, 759, 760
include/sbi_utils/sys/atcsmu.h, 761, 765
include/sbi_utils/sys/htif.h, 766, 768
include/sbi_utils/timer/aclint_mtimer.h, 768, 771
include/sbi_utils/timer/andes_plmt.h, 772, 773
include/sbi_utils/timer/fdt_timer.h, 774
index
    sbi_dbtr_trigger, 127
    sbi_domain, 129
INDEX_TO_TRIGGER
    sbi_dbtr.c, 827
init
    fdt_driver, 82
init_coldboot
    sbi_init.c, 971
init_count_offset
    sbi_init.c, 981
init_warm_resume
    sbi_init.c, 973
init_warm_startup
    sbi_init.c, 973
init_warmboot
    sbi_init.c, 975
INSERT_FIELD
    sbi_bitops.h, 315
INSN_16BIT_MASK
    riscv_encoding.h, 249
INSN_32BIT_MASK
    riscv_encoding.h, 249
INSN_IS_16BIT
    riscv_encoding.h, 249
INSN_IS_32BIT
    riscv_encoding.h, 249
INSN_LEN
    riscv_encoding.h, 249
INSN_MASK_C_FLD
    riscv_encoding.h, 249

INSN_MASK_C_FLDSP
riscv_encoding.h, 249
INSN_MASK_C_FLW
riscv_encoding.h, 249
INSN_MASK_C_FLWSP
riscv_encoding.h, 249
INSN_MASK_C_FSD
riscv_encoding.h, 249
INSN_MASK_C_FSDSP
riscv_encoding.h, 249
INSN_MASK_C_FSW
riscv_encoding.h, 249
INSN_MASK_C_FSWSP
riscv_encoding.h, 250
INSN_MASK_C_LD
riscv_encoding.h, 250
INSN_MASK_C_LDSP
riscv_encoding.h, 250
INSN_MASK_C_LH
riscv_encoding.h, 250
INSN_MASK_C_LHU
riscv_encoding.h, 250
INSN_MASK_C_LW
riscv_encoding.h, 250
INSN_MASK_C_LWSP
riscv_encoding.h, 250
INSN_MASK_C_SD
riscv_encoding.h, 250
INSN_MASK_C_SDSP
riscv_encoding.h, 250
INSN_MASK_C_SH
riscv_encoding.h, 250
INSN_MASK_C_SW
riscv_encoding.h, 250
INSN_MASK_C_SWSP
riscv_encoding.h, 250
INSN_MASK_FENCE_TSO
riscv_encoding.h, 250
INSN_MASK_FLD
riscv_encoding.h, 250
INSN_MASK_FLQ
riscv_encoding.h, 250
INSN_MASK_FLW
riscv_encoding.h, 251
INSN_MASK_FSD
riscv_encoding.h, 251
INSN_MASK_FSQ
riscv_encoding.h, 251
INSN_MASK_FSW
riscv_encoding.h, 251
INSN_MASK_LB
riscv_encoding.h, 251
INSN_MASK_LBU
riscv_encoding.h, 251
INSN_MASK_LD
riscv_encoding.h, 251
INSN_MASK_LH
riscv_encoding.h, 251
INSN_MASK_LHU
riscv_encoding.h, 251
INSN_MASK_LW
riscv_encoding.h, 251
INSN_MASK_LWU
riscv_encoding.h, 251
INSN_MASK_SB
riscv_encoding.h, 251
INSN_MASK_SD
riscv_encoding.h, 251
INSN_MASK_SH
riscv_encoding.h, 251
INSN_MASK_SW
riscv_encoding.h, 251
INSN_MASK_VECTOR_FAULT_ONLY_FIRST
riscv_encoding.h, 252
INSN_MASK_VECTOR_INDEXED
riscv_encoding.h, 252
INSN_MASK_VECTOR_LOAD_STORE
riscv_encoding.h, 252
INSN_MASK_VECTOR_STRIDE
riscv_encoding.h, 252
INSN_MASK_VECTOR_UNIT_STRIDE
riscv_encoding.h, 252
INSN_MASK_VECTOR_WHOLE_REG
riscv_encoding.h, 252
INSN_MASK_WFI
riscv_encoding.h, 252
INSN_MATCH_C_FLD
riscv_encoding.h, 252
INSN_MATCH_C_FLDSP
riscv_encoding.h, 252
INSN_MATCH_C_FLW
riscv_encoding.h, 252
INSN_MATCH_C_FLWSP
riscv_encoding.h, 252
INSN_MATCH_C_FSD
riscv_encoding.h, 252
INSN_MATCH_C_FSDSP
riscv_encoding.h, 252
INSN_MATCH_C_FSW
riscv_encoding.h, 252
INSN_MATCH_C_FSWSP
riscv_encoding.h, 252
INSN_MATCH_C_LD
riscv_encoding.h, 253
INSN_MATCH_C_LDSP
riscv_encoding.h, 253
INSN_MATCH_C_LH
riscv_encoding.h, 253
INSN_MATCH_C_LHU
riscv_encoding.h, 253
INSN_MATCH_C_LW
riscv_encoding.h, 253
INSN_MATCH_C_LWSP
riscv_encoding.h, 253
INSN_MATCH_C_SD
riscv_encoding.h, 253

INSN_MATCH_C_SDSP
riscv_encoding.h, 253
INSN_MATCH_C_SH
riscv_encoding.h, 253
INSN_MATCH_C_SW
riscv_encoding.h, 253
INSN_MATCH_C_SWSP
riscv_encoding.h, 253
INSN_MATCH_FENCE_TSO
riscv_encoding.h, 253
INSN_MATCH_FLD
riscv_encoding.h, 253
INSN_MATCH_FLQ
riscv_encoding.h, 253
INSN_MATCH_FLW
riscv_encoding.h, 253
INSN_MATCH_FSD
riscv_encoding.h, 254
INSN_MATCH_FSQ
riscv_encoding.h, 254
INSN_MATCH_FSW
riscv_encoding.h, 254
INSN_MATCH_LB
riscv_encoding.h, 254
INSN_MATCH_LBU
riscv_encoding.h, 254
INSN_MATCH_LD
riscv_encoding.h, 254
INSN_MATCH_LH
riscv_encoding.h, 254
INSN_MATCH_LHU
riscv_encoding.h, 254
INSN_MATCH_LW
riscv_encoding.h, 254
INSN_MATCH_LWU
riscv_encoding.h, 254
INSN_MATCH_SB
riscv_encoding.h, 254
INSN_MATCH_SD
riscv_encoding.h, 254
INSN_MATCH_SH
riscv_encoding.h, 254
INSN_MATCH_SW
riscv_encoding.h, 254
INSN_MATCH_VECTOR_LOAD
riscv_encoding.h, 254
INSN_MATCH_VECTOR_STORE
riscv_encoding.h, 255
INSN_MATCH_VL1RE16V
riscv_encoding.h, 255
INSN_MATCH_VL1RE32V
riscv_encoding.h, 255
INSN_MATCH_VL1RE64V
riscv_encoding.h, 255
INSN_MATCH_VL1RE8V
riscv_encoding.h, 255
INSN_MATCH_VL2RE16V
riscv_encoding.h, 255
INSN_MATCH_VL2RE32V
riscv_encoding.h, 255
INSN_MATCH_VL2RE64V
riscv_encoding.h, 255
INSN_MATCH_VL2RE8V
riscv_encoding.h, 255
INSN_MATCH_VL4RE16V
riscv_encoding.h, 255
INSN_MATCH_VL4RE32V
riscv_encoding.h, 255
INSN_MATCH_VL4RE64V
riscv_encoding.h, 255
INSN_MATCH_VL4RE8V
riscv_encoding.h, 255
INSN_MATCH_VL8RE16V
riscv_encoding.h, 255
INSN_MATCH_VL8RE32V
riscv_encoding.h, 255
INSN_MATCH_VL8RE64V
riscv_encoding.h, 256
INSN_MATCH_VL8RE8V
riscv_encoding.h, 256
INSN_MATCH_VLE16FFV
riscv_encoding.h, 256
INSN_MATCH_VLE16V
riscv_encoding.h, 256
INSN_MATCH_VLE32FFV
riscv_encoding.h, 256
INSN_MATCH_VLE32V
riscv_encoding.h, 256
INSN_MATCH_VLE64FFV
riscv_encoding.h, 256
INSN_MATCH_VLE64V
riscv_encoding.h, 256
INSN_MATCH_VLOXEI16V
riscv_encoding.h, 256
INSN_MATCH_VLOXEI32V
riscv_encoding.h, 256
INSN_MATCH_VLOXEI64V
riscv_encoding.h, 256
INSN_MATCH_VLOXSEG
riscv_encoding.h, 256
INSN_MATCH_VLSE16V
riscv_encoding.h, 256
INSN_MATCH_VLSE32V
riscv_encoding.h, 256
INSN_MATCH_VLSE64V
riscv_encoding.h, 257
INSN_MATCH_VLSEG
riscv_encoding.h, 257
INSN_MATCH_VLSEGFF
riscv_encoding.h, 257
INSN_MATCH_VLSSEG
riscv_encoding.h, 257
INSN_MATCH_VLUXEI16V
riscv_encoding.h, 257
INSN_MATCH_VLUXEI32V
riscv_encoding.h, 257

INSN_MATCH_VLUXEI64V
riscv_encoding.h, 257
INSN_MATCH_VLUXSEG
riscv_encoding.h, 257
INSN_MATCH_VS1RV
riscv_encoding.h, 257
INSN_MATCH_VS2RV
riscv_encoding.h, 257
INSN_MATCH_VS4RV
riscv_encoding.h, 257
INSN_MATCH_VS8RV
riscv_encoding.h, 258
INSN_MATCH_VSE16V
riscv_encoding.h, 258
INSN_MATCH_VSE32V
riscv_encoding.h, 258
INSN_MATCH_VSE64V
riscv_encoding.h, 258
INSN_MATCH_VSOXEI16V
riscv_encoding.h, 258
INSN_MATCH_VSOXEI32V
riscv_encoding.h, 258
INSN_MATCH_VSOXEI64V
riscv_encoding.h, 258
INSN_MATCH_VSOXSEG
riscv_encoding.h, 258
INSN_MATCH_VSSE16V
riscv_encoding.h, 258
INSN_MATCH_VSSE32V
riscv_encoding.h, 258
INSN_MATCH_VSSE64V
riscv_encoding.h, 258
INSN_MATCH_VSSEG
riscv_encoding.h, 258
INSN_MATCH_VSSSEG
riscv_encoding.h, 258
INSN_MATCH_VSUXEI16V
riscv_encoding.h, 259
INSN_MATCH_VSUXEI32V
riscv_encoding.h, 259
INSN_MATCH_VSUXEI64V
riscv_encoding.h, 259
INSN_MATCH_VSUXSEG
riscv_encoding.h, 259
INSN_MATCH_WFI
riscv_encoding.h, 259
int16_t
sbi_types.h, 621
INT32_MAX
libfdt_env.h, 1429
int32_t
sbi_types.h, 621
INT_CLK_1024
fdt_reset_atcwdt200.c, 1507
INT_CLK_16384
fdt_reset_atcwdt200.c, 1507
INT_CLK_2048
fdt_reset_atcwdt200.c, 1507
INT_CLK_256
fdt_reset_atcwdt200.c, 1507
INT_CLK_32768
fdt_reset_atcwdt200.c, 1508
INT_CLK_4096
fdt_reset_atcwdt200.c, 1508
INT_CLK_64
fdt_reset_atcwdt200.c, 1508
INT_CLK_8192
fdt_reset_atcwdt200.c, 1508
INT_EN
fdt_reset_atcwdt200.c, 1508
INT_MAX
libfdt_env.h, 1429
INT_TIME_MSK
fdt_reset_atcwdt200.c, 1508
INT_TIME_OFF
fdt_reset_atcwdt200.c, 1508
INVALID_ADDR
sbi_mpxy.c, 997
ipi_clear
sbi_ipi_device, 136
ipi_data_off
sbi_ipi.c, 990
ipi_dev
sbi_ipi.c, 990
ipi_halt_event
sbi_ipi.c, 990
ipi_halt_ops
sbi_ipi.c, 990
ipi_init
sbi_platform_operations, 145
ipi_mswi_cold_init
fdt_ipi_mswi.c, 1226
ipi_mswi_match
fdt_ipi_mswi.c, 1226
ipi_ops_array
sbi_ipi.c, 990
ipi_plicsw_match
fdt_ipi_plicsw.c, 1228
ipi_send
sbi_ipi_device, 136
ipi_smode_event
sbi_ipi.c, 990
ipi_smode_ops
sbi_ipi.c, 990
irq_handle
sbi_irqchip_device, 138
IRQ_M_EXT
riscv_encoding.h, 259
IRQ_M_SOFT
riscv_encoding.h, 259
IRQ_M_TIMER
riscv_encoding.h, 259
IRQ_PMU_OVF
riscv_encoding.h, 259
IRQ_S_EXT
riscv_encoding.h, 259

IRQ_S_GEXT
riscv_encoding.h, 259
IRQ_S_SOFT
riscv_encoding.h, 259
IRQ_S_TIMER
riscv_encoding.h, 259
IRQ_VS_EXT
riscv_encoding.h, 260
IRQ_VS_SOFT
riscv_encoding.h, 260
IRQ_VS_TIMER
riscv_encoding.h, 260
irqchip
aplic_data, 80
plic_data, 98
irqchip_aplic_cold_init
fdt_irqchip_aplic.c, 1238
irqchip_aplic_match
fdt_irqchip_aplic.c, 1238
irqchip_imsic_cold_init
fdt_irqchip_imsic.c, 1239
irqchip_imsic_match
fdt_irqchip_imsic.c, 1240
irqchip_imsic_update_hartid_table
fdt_irqchip_imsic.c, 1239
irqchip_init
sbi_platform_operations, 145
irqchip_plic_cold_init
fdt_irqchip_plic.c, 1241
irqchip_plic_match
fdt_irqchip_plic.c, 1242
irqchip_plic_update_context_map
fdt_irqchip_plic.c, 1241
IS_FAULT_ONLY_FIRST_LOAD
riscv_encoding.h, 260
IS_FAULT_ONLY_FIRST_MATCH
riscv_encoding.h, 260
IS_INDEXED_LOAD
riscv_encoding.h, 260
IS_INDEXED_MATCH
riscv_encoding.h, 260
IS_INDEXED_STORE
riscv_encoding.h, 260
IS_MASKED
riscv_encoding.h, 260
is_pmp_entry_mapped
riscv_asm.c, 780
riscv_asm.h, 171
is_region_before
sbi_domain.c, 841
is_region_compatible
sbi_domain.c, 841
is_region_subset
sbi_domain.c, 842
is_region_valid
sbi_domain.c, 843
is_rpmi_hsm_susp_supported
fdt_hsm_rpmi.c, 1193
IS_STRIDE_LOAD
riscv_encoding.h, 261
IS_STRIDE_MATCH
riscv_encoding.h, 261
IS_STRIDE_STORE
riscv_encoding.h, 261
IS_UNIT_STRIDE_LOAD
riscv_encoding.h, 261
IS_UNIT_STRIDE_MATCH
riscv_encoding.h, 262
IS_UNIT_STRIDE_STORE
riscv_encoding.h, 262
IS_VECTOR_INSN_MATCH
riscv_encoding.h, 262
IS_VECTOR_LOAD_STORE
riscv_encoding.h, 262
IS_WHOLE_REG_LOAD
riscv_encoding.h, 263
IS_WHOLE_REG_MATCH
riscv_encoding.h, 263
IS_WHOLE_REG_STORE
riscv_encoding.h, 263

L
quad.h, 1454
last_comp_version
fdt_header, 84
last_hartindex_having_scratch
sbi_scratch.c, 1032
sbi_scratch.h, 562
last_irq
aplic_delegate_data, 80
le16_t
sbi_types.h, 621
le16_to_cpu
sbi_byteorder.h, 325
le32_t
sbi_types.h, 621
le32_to_cpu
sbi_byteorder.h, 325
le64_t
sbi_types.h, 621
le64_to_cpu
sbi_byteorder.h, 325
len
fdt_property, 87
LGREG
riscv_asm.h, 169
LHALF
quad.h, 1454
LHUP
quad.h, 1454
lhxs
aplic_msicfg_data, 81
lhxw
aplic_msicfg_data, 81
lib/sbi/riscv_asm.c, 775
lib/sbi/riscv_atomic.c, 785
lib/sbi/riscv_locks.c, 791

lib/sbi/sbi_bitmap.c, 795
lib/sbi/sbi_bitops.c, 797
lib/sbi/sbi_console.c, 799
lib/sbi/sbi_cppc.c, 821
lib/sbi/sbi_dbtr.c, 825
lib/sbi/sbi_domain.c, 838
lib/sbi/sbi_domain_context.c, 856
lib/sbi/sbi_domain_data.c, 862
lib/sbi/sbi_double_trap.c, 866
lib/sbi/sbi_ecall.c, 868
lib/sbi/sbi_ecall_base.c, 872
lib/sbi/sbi_ecall_cppc.c, 874
lib/sbi/sbi_ecall_dbcn.c, 876
lib/sbi/sbi_ecall_dbtr.c, 877
lib/sbi/sbi_ecall_fwft.c, 879
lib/sbi/sbi_ecall_hsm.c, 880
lib/sbi/sbi_ecall_ipi.c, 883
lib/sbi/sbi_ecall_legacy.c, 884
lib/sbi/sbi_ecall_mpxy.c, 886
lib/sbi/sbi_ecall_pmu.c, 888
lib/sbi/sbi_ecall_rfence.c, 890
lib/sbi/sbi_ecall_srst.c, 891
lib/sbi/sbi_ecall_sse.c, 894
lib/sbi/sbi_ecall_susp.c, 896
lib/sbi/sbi_ecall_time.c, 898
lib/sbi/sbi_ecall_vendor.c, 899
lib/sbi/sbi_emulate_csr.c, 902
lib/sbi/sbi_fifo.c, 905
lib/sbi/sbi_fwft.c, 914
lib/sbi/sbi_hart.c, 924
lib/sbi/sbi_heap.c, 945
lib/sbi/sbi_hsm.c, 953
lib/sbi/sbi_illegal_insn.c, 967
lib/sbi/sbi_init.c, 969
lib/sbi/sbi_ipi.c, 981
lib/sbi/sbi_irqchip.c, 991
lib/sbi/sbi_math.c, 994
lib/sbi/sbi_mpxy.c, 994
lib/sbi/sbi_platform.c, 1007
lib/sbi/sbi_pmu.c, 1009
lib/sbi/sbi_scratch.c, 1028
lib/sbi/sbi_sse.c, 1032
lib/sbi/sbi_string.c, 1064
lib/sbi/sbi_system.c, 1068
lib/sbi/sbi_timer.c, 1075
lib/sbi/sbi_tlb.c, 1082
lib/sbi/sbi_trap.c, 1094
lib/sbi/sbi_trap_ldst.c, 1099
lib/sbi/sbi_trap_v_ldst.c, 1106
lib/sbi/sbi_unpriv.c, 1108
lib/sbi/tests/riscv_atomic_test.c, 1110
lib/sbi/tests/riscv_locks_test.c, 1115
lib/sbi/tests/sbi_bitmap_test.c, 1118
lib/sbi/tests/sbi_console_test.c, 1120
lib/sbi/tests/sbi_math_test.c, 1125
lib/sbi/tests/sbi_unit_test.c, 1126
lib/utils/cppc/fdt_cppc.c, 1128
lib/utils/cppc/fdt_cppc_rpmi.c, 1129
lib/utils/fdt/fdt_domain.c, 1133
lib/utils/fdt/fdt_driver.c, 1140
lib/utils/fdt/fdt_fixup.c, 1145
lib/utils/fdt/fdt_helper.c, 1154
lib/utils/fdt/fdt_pmu.c, 1175
lib/utils/gpio/fdt_gpio.c, 1177
lib/utils/gpio/fdt_gpio_designware.c, 1180
lib/utils/gpio/fdt_gpio_sifive.c, 1182
lib/utils/gpio/fdt_gpio_starfive.c, 1185
lib/utils/gpio/gpio.c, 1187
lib/utils/hsm/fdt_hsm.c, 1191
lib/utils/hsm/fdt_hsm_rpmi.c, 1192
lib/utils/i2c/dw_i2c.c, 1198
lib/utils/i2c/fdt_i2c.c, 1206
lib/utils/i2c/fdt_i2c_dw.c, 1208
lib/utils/i2c/fdt_i2c_sifive.c, 1210
lib/utils/i2c/i2c.c, 1217
lib/utils/ipi/aclint_mswi.c, 1220
lib/utils/ipi/andes_plicsw.c, 1222
lib/utils/ipi/fdt_ipi.c, 1224
lib/utils/ipi/fdt_ipi_mswi.c, 1225
lib/utils/ipi/fdt_ipi_plicsw.c, 1227
lib/utils/irqchip/aplic.c, 1228
lib/utils/irqchip/fdt_irqchip.c, 1236
lib/utils/irqchip/fdt_irqchip_aplic.c, 1237
lib/utils/irqchip/fdt_irqchip_imsic.c, 1238
lib/utils/irqchip/fdt_irqchip_plic.c, 1240
lib/utils/irqchip/imsic.c, 1242
lib/utils/irqchip/plic.c, 1249
lib/utils/libfdt/fdt.c, 1256
lib/utils/libfdt/fdt.h, 1267, 1268
lib/utils/libfdt/fdt_addresses.c, 1269
lib/utils/libfdt/fdt_check.c, 1273
lib/utils/libfdt/fdt_empty_tree.c, 1275
lib/utils/libfdt/fdt_overlay.c, 1276
lib/utils/libfdt/fdt_ro.c, 1286
lib/utils/libfdt/fdt_rw.c, 1322
lib/utils/libfdt/fdt_strerror.c, 1337
lib/utils/libfdt/fdt_sw.c, 1338
lib/utils/libfdt/fdt_wip.c, 1348
lib/utils/libfdt/libfdt.h, 1352, 1421
lib/utils/libfdt/libfdt_env.h, 1428, 1433
lib/utils/libfdt/libfdt_internal.h, 1434, 1443
lib/utils/libquad/divdi3.c, 1445
lib/utils/libquad/include/limits.h, 1447
lib/utils/libquad/include/sys/cdefs.h, 1447, 1448
lib/utils/libquad/include/sys/types.h, 1448, 1449
lib/utils/libquad/moddi3.c, 1449
lib/utils/libquad/qdivrem.c, 1450
lib/utils/libquad/quad.h, 1453, 1457
lib/utils/libquad/udivdi3.c, 1458
lib/utils/libquad/umoddi3.c, 1459
lib/utils/mailbox/fdt_mailbox.c, 1461
lib/utils/mailbox/fdt_mailbox_rpmi_shmem.c, 1463
lib/utils/mailbox/mailbox.c, 1474
lib/utils/mailbox/rpmi_mailbox.c, 1479
lib/utils/mpxy/fdt_mpxy.c, 1482
lib/utils/mpxy/fdt_mpxy_rpmi_mbox.c, 1484

lib/utils/regmap/fdt_regmap.c, 1491
lib/utils/regmap/fdt_regmap_syscon.c, 1493
lib/utils/regmap/regmap.c, 1500
lib/utils/reset/fdt_reset.c, 1505
lib/utils/reset/fdt_reset_atcwdt200.c, 1506
lib/utils/reset/fdt_reset_gpio.c, 1510
lib/utils/reset/fdt_reset_htif.c, 1515
lib/utils/reset/fdt_reset_rpmi.c, 1516
lib/utils/reset/fdt_reset_sg2042_hwmon_mcu.c, 1519
lib/utils/reset/fdt_reset_sunxi_wdt.c, 1522
lib/utils/reset/fdt_reset_syscon.c, 1525
lib/utils/serial/cadence-uart.c, 1529
lib/utils/serial/fdt_serial.c, 1534
lib/utils/serial/fdt_serial_cadence.c, 1535
lib/utils/serial/fdt_serial_gaisler.c, 1537
lib/utils/serial/fdt_serial_htif.c, 1538
lib/utils/serial/fdt_serial_litex.c, 1540
lib/utils/serial/fdt_serial_renesas_scif.c, 1541
lib/utils/serial/fdt_serial_shakti.c, 1543
lib/utils/serial/fdt_serial_sifive.c, 1544
lib/utils/serial/fdt_serial_uart8250.c, 1546
lib/utils/serial/fdt_serial_xlnx_uartlite.c, 1547
lib/utils/serial/gaisler-uart.c, 1549
lib/utils/serial/litex-uart.c, 1553
lib/utils/serial/renesas_scif.c, 1556
lib/utils/serial/semihosting.c, 1561
lib/utils/serial/shakti-uart.c, 1567
lib/utils/serial/sifive-uart.c, 1569
lib/utils/serial/uart8250.c, 1574
lib/utils/serial/xlnx-uartlite.c, 1579
lib/utils/suspend/fdt_suspend.c, 1582
lib/utils/suspend/fdt_suspend_rpmi.c, 1584
lib/utils/sys/atcsmu.c, 1587
lib/utils/sys/htif.c, 1588
lib/utils/timer/aclint_mtimer.c, 1597
lib/utils/timer/andes_plmt.c, 1601
lib/utils/timer/fdt_timer.c, 1603
lib/utils/timer/fdt_timer_mtimer.c, 1605
lib/utils/timer/fdt_timer_plmt.c, 1607
libfdt.h
- fdt32_ld, 1362
- fdt32_st, 1362
- fdt64_ld, 1362
- fdt64_st, 1362
- fdt_add_mem_rsv, 1362
- fdt_add_reservemap_entry, 1363
- fdt_add_subnode, 1363
- fdt_add_subnode_namelen, 1364
- fdt_address_cells, 1364
- fdt_appendprop, 1365
- fdt_appendprop_addrrange, 1366
- fdt_appendprop_cell, 1366
- fdt_appendprop_string, 1358
- fdt_appendprop_u32, 1367
- fdt_appendprop_u64, 1367
- fdt_begin_node, 1368
- fdt_boot_cpuid_phys, 1358
- fdt_check_full, 1368
- fdt_check_header, 1369
- fdt_create, 1369
- fdt_create_empty_tree, 1370
- FDT_CREATE_FLAG_NO_NAME_DEDUP, 1358
- FDT_CREATE_FLAGS_ALL, 1358
- fdt_create_with_flags, 1371
- fdt_del_mem_rsv, 1371
- fdt_del_node, 1372
- fdt_delprop, 1372
- fdt_end_node, 1373
- FDT_ERR_ALIGNMENT, 1358
- FDT_ERR_BADFLAGS, 1358
- FDT_ERR_BADLAYOUT, 1358
- FDT_ERR_BADMAGIC, 1358
- FDT_ERR_BADNCELLS, 1358
- FDT_ERR_BADOFFSET, 1358
- FDT_ERR_BADOVERLAY, 1358
- FDT_ERR_BADPATH, 1358
- FDT_ERR_BADPHANDLE, 1358
- FDT_ERR_BADSTATE, 1358
- FDT_ERR_BADSTRUCTURE, 1359
- FDT_ERR_BADVALUE, 1359
- FDT_ERR_BADVERSION, 1359
- FDT_ERR_EXISTS, 1359
- FDT_ERR_INTERNAL, 1359
- FDT_ERR_MAX, 1359
- FDT_ERR_NOPHANDLES, 1359
- FDT_ERR_NOSPACE, 1359
- FDT_ERR_NOTFOUND, 1359
- FDT_ERR_TRUNCATED, 1359
- fdt_find_max_phandle, 1373
- fdt_finish, 1374
- fdt_finish_reservemap, 1375
- fdt_first_property_offset, 1375
- fdt_first_subnode, 1376
- FDT_FIRST_SUPPORTED_VERSION, 1359
- fdt_for_each_property_offset, 1359
- fdt_for_each_subnode, 1359
- fdt_generate_phandle, 1377
- fdt_get_alias, 1377
- fdt_get_alias_namelen, 1377
- fdt_get_header, 1360
- fdt_get_max_phandle, 1378
- fdt_get_mem_rsv, 1378
- fdt_get_name, 1379
- fdt_get_path, 1380
- fdt_get_phandle, 1381
- fdt_get_property, 1381
- fdt_get_property_by_offset, 1382
- fdt_get_property_namelen, 1382
- fdt_get_property_w, 1383
- fdt_get_string, 1383
- fdt_getprop, 1385
- fdt_getprop_by_offset, 1385
- fdt_getprop_namelen, 1386
- fdt_getprop_namelen_w, 1387
- fdt_getprop_w, 1388
- fdt_header_size, 1388

fdt_header_size_, 1389
fdt_last_comp_version, 1360
FDT_LAST_COMPATIBLE_VERSION, 1360
FDT_LAST_SUPPORTED_VERSION, 1360
fdt_magic, 1360
FDT_MAX_NCELLS, 1360
FDT_MAX_PHANDLE, 1360
fdt_move, 1389
fdt_next_node, 1389
fdt_next_property_offset, 1391
fdt_next_subnode, 1392
fdt_next_tag, 1393
fdt_node_check_compatible, 1394
fdt_node_depth, 1395
fdt_node_offset_by_compatible, 1396
fdt_node_offset_by_phandle, 1397
fdt_node_offset_by_prop_value, 1398
fdt_nop_node, 1399
fdt_nop_property, 1399
fdt_num_mem_rsv, 1400
fdt_off_dt_strings, 1360
fdt_off_dt_struct, 1360
fdt_off_mem_rsvmap, 1361
fdt_offset_ptr, 1400
fdt_offset_ptr_w, 1401
fdt_open_into, 1402
fdt_overlay_apply, 1402
fdt_pack, 1403
fdt_parent_offset, 1403
fdt_path_offset, 1404
fdt_path_offset_namelen, 1405
fdt_property, 1406
fdt_property_cell, 1406
fdt_property_placeholder, 1407
fdt_property_string, 1361
fdt_property_u32, 1407
fdt_property_u64, 1408
fdt_resize, 1408
fdt_set_hdr_, 1361, 1408, 1409
fdt_set_name, 1409
fdt_setprop, 1410
fdt_setprop_cell, 1410
fdt_setprop_empty, 1361
fdt_setprop_inplace, 1411
fdt_setprop_inplace_cell, 1411
fdt_setprop_inplace_namelen_partial, 1411
fdt_setprop_inplace_u32, 1412
fdt_setprop_inplace_u64, 1412
fdt_setprop_placeholder, 1413
fdt_setprop_string, 1361
fdt_setprop_u32, 1413
fdt_setprop_u64, 1414
fdt_size_cells, 1414
fdt_size_dt_strings, 1361
fdt_size_dt_struct, 1361
fdt_strerror, 1415
fdt_string, 1415
fdt_stringlist_contains, 1416
fdt_stringlist_count, 1417
fdt_stringlist_get, 1418
fdt_stringlist_search, 1418
fdt_subnode_offset, 1418
fdt_subnode_offset_namelen, 1419
fdt_supernode_atdepth_offset, 1420
fdt_totalsize, 1361
fdt_version, 1361
libfdt_env.h
cpu_to_fdt16, 1430
cpu_to_fdt32, 1430
cpu_to_fdt64, 1431
fdt16_t, 1430
fdt16_to_cpu, 1431
fdt32_t, 1430
fdt32_to_cpu, 1431
fdt64_t, 1430
fdt64_to_cpu, 1432
FDT_BITWISE, 1429
FDT_FORCE, 1429
INT32_MAX, 1429
INT_MAX, 1429
memchr, 1429
memcmp, 1429
memcpy, 1429
memmove, 1429
memset, 1429
strchr, 1429
strcmp, 1430
strcpy, 1430
strlen, 1430
strncmp, 1430
strncpy, 1430
strnlen, 1430
strrchr, 1430
UINT32_MAX, 1430
UINT_MAX, 1430
libfdt_internal.h
ASSUME_LATEST, 1436
ASSUME_LIBFDT_FLAWLESS, 1436
ASSUME_LIBFDT_ORDER, 1436
ASSUME_NO_ROLLBACK, 1436
ASSUME_PERFECT, 1435
ASSUME_VALID_DTB, 1436
ASSUME_VALID_INPUT, 1436
can_assume, 1435
can_assume_, 1436
fdt32_ld_, 1436
fdt64_ld_, 1437
FDT_ALIGN, 1435
FDT_ASSUME_MASK, 1435
fdt_check_node_offset_, 1437
fdt_check_prop_offset_, 1438
fdt_find_string_, 1439
fdt_mem_rsv_, 1440
fdt_mem_rsv_w_, 1440
fdt_node_end_offset_, 1441
fdt_offset_ptr_, 1441

fdt_offset_ptr_w_, 1442
FDT_RO_PROBE, 1435
fdt_ro_probe_, 1442
FDT_SW_MAGIC, 1435
FDT_TAGALIGN, 1435
LIGHT_SLEEP_CMD
atcsmu.h, 763
LIGHTSLEEP_MODE
atcsmu.h, 763
likely
sbi_types.h, 620
limits.h
CHAR_BIT, 1447
Linux as a direct payload to OpenSBI, 33
litex-uart.c
get_reg, 1554
litex_console, 1556
litex_uart_getc, 1554
litex_uart_init, 1555
litex_uart_putc, 1555
set_reg, 1556
uart_base, 1556
UART_EV_RX, 1554
UART_REG_EV_ENABLE, 1554
UART_REG_EV_PENDING, 1554
UART_REG_EV_STATUS, 1554
UART_REG_RXEMPTY, 1554
UART_REG_RXTX, 1554
UART_REG_TXFULL, 1554
litex-uart.h
litex_uart_init, 748
litex_console
litex-uart.c, 1556
litex_uart_getc
litex-uart.c, 1554
litex_uart_init
litex-uart.c, 1555
litex-uart.h, 748
litex_uart_putc
litex-uart.c, 1555
local_event_count
sbi_sse.c, 1063
local_timer_stop
sbi_cpu_idle_state, 124
locks_test_cases
riscv_locks_test.c, 1118
log2roundup
sbi_math.c, 994
sbi_math.h, 509
log2roundup_test
sbi_math_test.c, 1125
LOG_REGBYTES
riscv_encoding.h, 263
LONG_BITS
quad.h, 1454

magic
fdt_header, 84
fw_dynamic_info, 89
mailbox.c
mbox_chan_get_attribute, 1475
mbox_chan_set_attribute, 1475
mbox_chan_xfer, 1475
mbox_controller_add, 1476
mbox_controller_find, 1477
mbox_controller_free_chan, 1477
mbox_controller_remove, 1477
mbox_controller_request_chan, 1478
SBI_LIST_HEAD, 1479
mailbox.h
mbox_chan_get_attribute, 704
MBOX_CHAN_MAX_ARGS, 703
mbox_chan_set_attribute, 704
mbox_chan_xfer, 704
mbox_controller_add, 705
mbox_controller_find, 706
mbox_controller_free_chan, 706
mbox_controller_remove, 707
mbox_controller_request_chan, 707
mbox_xfer_init_rx, 703
mbox_xfer_init_tx, 703
mbox_xfer_init_txrx, 703
MBOX_XFER_SEQ, 704
mbox_xfer_set_sequence, 704
to_mbox_chan, 704
to_mbox_controller, 704
MAKE_MESSAGE_ID
fdt_mailbox_rpmi_shmem.c, 1465
MANGO_BOARD_TYPE_MASK
fdt_reset_sg2042_hwmon_mcu.c, 1520
MASK_CSR
riscv_encoding.h, 263
MASK_FUNCT3
riscv_encoding.h, 263
MASK_RS1
riscv_encoding.h, 263
match_table
fdt_driver, 82
math_test_cases
sbi_math_test.c, 1126
MAX
sbi_types.h, 620
MAX_HSM_SUPSEND_STATE_NAMELEN
fdt_hsm_rpmi.c, 1193
max_xfer_len
mbox_controller, 95
mb
riscv_barrier.h, 185
mbox
mbox_chan, 94
mbox_chan, 93
chan_args, 94
mbox, 94
node, 94
mbox_chan_get_attribute
mailbox.c, 1475
mailbox.h, 704

MBOX_CHAN_MAX_ARGS
mailbox.h, 703
mbox_chan_set_attribute
mailbox.c, 1475
mailbox.h, 704
mbox_chan_xfer
mailbox.c, 1475
mailbox.h, 704
mbox_controller, 94
chan_list, 94
driver, 94
free_chan, 94
get_attribute, 94
id, 95
max_xfer_len, 95
node, 95
request_chan, 95
set_attribute, 95
xfer, 95
xfer_next_seq, 95
mbox_controller_add
mailbox.c, 1476
mailbox.h, 705
mbox_controller_find
mailbox.c, 1477
mailbox.h, 706
mbox_controller_free_chan
mailbox.c, 1477
mailbox.h, 706
mbox_controller_remove
mailbox.c, 1477
mailbox.h, 707
mbox_controller_request_chan
mailbox.c, 1478
mailbox.h, 707
mbox_xfer, 95
args, 96
flags, 96
rx, 96
rx_len, 96
rx_timeout, 96
seq, 96
tx, 96
tx_len, 96
tx_timeout, 96
mbox_xfer_init_rx
mailbox.h, 703
mbox_xfer_init_tx
mailbox.h, 703
mbox_xfer_init_txrx
mailbox.h, 703
MBOX_XFER_SEQ
mailbox.h, 704
mbox_xfer_set_sequence
mailbox.h, 704
MCAUSE_IRQ_MASK
riscv_encoding.h, 263
mcu_adapter
fdt_reset_sg2042_hwmon_mcu.c, 1522
mcu_reg
fdt_reset_sg2042_hwmon_mcu.c, 1522
memchr
libfdt_env.h, 1429
memcmp
libfdt_env.h, 1429
memcpy
libfdt_env.h, 1429
memmove
libfdt_env.h, 1429
memset
libfdt_env.h, 1429
mepc
sbi_trap_regs, 156
mhpm_bits
sbi_hart_features, 134
mhpm_mask
sbi_hart_features, 134
MHPMEVENT_MINH
riscv_encoding.h, 263
MHPMEVENT_OF
riscv_encoding.h, 263
MHPMEVENT_SINH
riscv_encoding.h, 264
MHPMEVENT_SSCOF_MASK
riscv_encoding.h, 264
MHPMEVENT_UINH
riscv_encoding.h, 264
MHPMEVENT_VSINH
riscv_encoding.h, 264
MHPMEVENT_VUINH
riscv_encoding.h, 264
MHPMEVENTH_MINH
riscv_encoding.h, 264
MHPMEVENTH_OF
riscv_encoding.h, 264
MHPMEVENTH_SINH
riscv_encoding.h, 264
MHPMEVENTH_UINH
riscv_encoding.h, 264
MHPMEVENTH_VSINH
riscv_encoding.h, 264
MHPMEVENTH_VUINH
riscv_encoding.h, 264
MIN
sbi_types.h, 620
min_residency_us
rpmi_hsm_get_susp_info_resp, 115
sbi_cpu_idle_state, 124
MIP_LCOFIP
riscv_encoding.h, 264
MIP_MEIP
riscv_encoding.h, 264
MIP_MSIP
riscv_encoding.h, 264
MIP_MTIP
riscv_encoding.h, 264

MIP_SEIP
riscv_encoding.h, 265
MIP_SGEIP
riscv_encoding.h, 265
MIP_SSIP
riscv_encoding.h, 265
MIP_STIP
riscv_encoding.h, 265
MIP_VSEIP
riscv_encoding.h, 265
MIP_VSSIP
riscv_encoding.h, 265
MIP_VSTIP
riscv_encoding.h, 265
MIS_DELEG
sbi_fwft.c, 915
misa_check_extension
sbi_platform_operations, 145
misa_extension
riscv_asm.h, 169
misa_extension_imp
riscv_asm.c, 780
riscv_asm.h, 171
misa_get_xlen
sbi_platform_operations, 145
misa_string
riscv_asm.c, 781
riscv_asm.h, 172
misa_xlen
riscv_asm.c, 781
riscv_asm.h, 172
misc_mem_opcode_insn
sbi_illegal_insn.c, 968
moddi3.c
__FBSDID, 1450
__moddi3, 1450
MODE_APPEND
semihosting.h, 752
MODE_BINARY
semihosting.h, 752
MODE_PLUS
semihosting.h, 752
MODE_READ
semihosting.h, 752
MODE_WRITE
semihosting.h, 752
mpxy_check_write_attr
fdt_mpxy_rpmi_mbox.c, 1486
mpxy_check_write_std_attr
sbi_mpxy.c, 999
mpxy_copy_attrs
fdt_mpxy_rpmi_mbox.c, 1486
mpxy_copy_std_attrs
sbi_mpxy.c, 999
mpxy_find_channel
sbi_mpxy.c, 1000
mpxy_find_rpmi_srvid
fdt_mpxy_rpmi_mbox.c, 1487
mpxy_init
sbi_platform_operations, 145
mpxy_is_std_attr
sbi_mpxy.c, 1000
mpxy_mbox_get_notifications
fdt_mpxy_rpmi_mbox.c, 1487
mpxy_mbox_init
fdt_mpxy_rpmi_mbox.c, 1487
mpxy_mbox_match
fdt_mpxy_rpmi_mbox.c, 1491
mpxy_mbox_read_attributes
fdt_mpxy_rpmi_mbox.c, 1488
mpxy_mbox_send_message_withoutresp
fdt_mpxy_rpmi_mbox.c, 1488
mpxy_mbox_send_message_withresp
fdt_mpxy_rpmi_mbox.c, 1489
mpxy_mbox_write_attributes
fdt_mpxy_rpmi_mbox.c, 1489
mpxy_msgprot_rpmi_attr_id
fdt_mpxy_rpmi_mbox.c, 1485
MPXY_MSGPROT_RPMI_ATTR_MAX_ID
fdt_mpxy_rpmi_mbox.c, 1485
MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_ID
fdt_mpxy_rpmi_mbox.c, 1485
MPXY_MSGPROT_RPMI_ATTR_SERVICEGROUP_VERSION
fdt_mpxy_rpmi_mbox.c, 1485
mpxy_shmem_enabled
sbi_mpxy.c, 1000
mpxy_state_offset
sbi_mpxy.c, 1007
mpxy_std_attrs_init
sbi_mpxy.c, 1001
mpxy_write_attr
fdt_mpxy_rpmi_mbox.c, 1490
mpxy_write_std_attr
sbi_mpxy.c, 1001
MSECCFG_MML
riscv_encoding.h, 265
MSECCFG_MML_SHIFT
riscv_encoding.h, 265
MSECCFG_MMWP
riscv_encoding.h, 265
MSECCFG_MMWP_SHIFT
riscv_encoding.h, 265
MSECCFG_RLB
riscv_encoding.h, 265
MSECCFG_RLB_SHIFT
riscv_encoding.h, 265
MSECCFG_SSEED
riscv_encoding.h, 265
MSECCFG_SSEED_SHIFT
riscv_encoding.h, 265
MSECCFG_USEED
riscv_encoding.h, 266
MSECCFG_USEED_SHIFT
riscv_encoding.h, 266
msg_completion_timeout
sbi_mpxy_channel_attrs, 140

msg_data_maxlen
sbi_mpxy_channel_attrs, 141
msg_proto_id
sbi_mpxy_channel_attrs, 141
msg_proto_version
sbi_mpxy_channel_attrs, 141
msg_send_timeout
sbi_mpxy_channel_attrs, 141
msi_addr_hi
sbi_mpxy_msi_info, 141
msi_addr_lo
sbi_mpxy_msi_info, 141
msi_control
sbi_mpxy_channel_attrs, 141
msi_data
sbi_mpxy_msi_info, 141
msi_info
sbi_mpxy_channel_attrs, 141
msicfg_mmode
aplic_data, 80
msicfg_smode
aplic_data, 80
mstatus
sbi_trap_regs, 156
MSTATUS32_SD
riscv_encoding.h, 266
MSTATUS64_SD
riscv_encoding.h, 266
MSTATUS_FS
riscv_encoding.h, 266
mstatus_init
sbi_hart.c, 932
MSTATUS_MIE
riscv_encoding.h, 266
MSTATUS_MPIE
riscv_encoding.h, 266
MSTATUS_MPP
riscv_encoding.h, 266
MSTATUS_MPP_SHIFT
riscv_encoding.h, 266
MSTATUS_MPRV
riscv_encoding.h, 266
MSTATUS_MXR
riscv_encoding.h, 266
MSTATUS_SD
riscv_encoding.h, 266
MSTATUS_SIE
riscv_encoding.h, 266
MSTATUS_SPELP
riscv_encoding.h, 266
MSTATUS_SPIE
riscv_encoding.h, 267
MSTATUS_SPIE_SHIFT
riscv_encoding.h, 267
MSTATUS_SPP
riscv_encoding.h, 267
MSTATUS_SPP_SHIFT
riscv_encoding.h, 267
MSTATUS_SUM
riscv_encoding.h, 267
MSTATUS_TSR
riscv_encoding.h, 267
MSTATUS_TVM
riscv_encoding.h, 267
MSTATUS_TW
riscv_encoding.h, 267
MSTATUS_UBE
riscv_encoding.h, 267
MSTATUS_VS
riscv_encoding.h, 267
MSTATUS_XS
riscv_encoding.h, 267
mstatusH
sbi_trap_regs, 156
MSTATUSH_GVA
riscv_encoding.h, 267
MSTATUSH_GVA_SHIFT
riscv_encoding.h, 267
MSTATUSH_MBE
riscv_encoding.h, 267
MSTATUSH_MDT
riscv_encoding.h, 267
MSTATUSH_MPELP
riscv_encoding.h, 268
MSTATUSH_MPV
riscv_encoding.h, 268
MSTATUSH_SBE
riscv_encoding.h, 268
mswi_get_hart_data_ptr
aclint_mswi.c, 1221
mswi_ipi_clear
aclint_mswi.c, 1222
mswi_ipi_send
aclint_mswi.c, 1222
mswi_ptr_offset
aclint_mswi.c, 1222
mswi_set_hart_data_ptr
aclint_mswi.c, 1221
mt_reference
fdt_timer_mtimer.c, 1606
mtime_addr
aclint_mtimer_data, 78
mtime_freq
aclint_mtimer_data, 78
mtime_size
aclint_mtimer_data, 78
mtimecmp_addr
aclint_mtimer_data, 78
mtimecmp_size
aclint_mtimer_data, 79
mtimer
aclint_mtimer.c, 1601
mtimer_event_start
aclint_mtimer.c, 1599
mtimer_event_stop
aclint_mtimer.c, 1600

mtimer_get_hart_data_ptr
aclint_mtimer.c, 1598
mtimer_ptr_offset
aclint_mtimer.c, 1601
MTIMER_REGION_ALIGN
aclint_mtimer.h, 770
mtimer_set_hart_data_ptr
aclint_mtimer.c, 1598
mtimer_time_rd32
aclint_mtimer.c, 1600
mtimer_time_rd64
aclint_mtimer.c, 1600
mtimer_time_wr32
aclint_mtimer.c, 1600
mtimer_time_wr64
aclint_mtimer.c, 1600
mtimer_value
aclint_mtimer.c, 1601

name
fdt_general_fixup, 83
fdt_node_header, 85
rpmi_clock_get_attributes_resp, 102
sbi_console_device, 123
sbi_cppc_device, 123
sbi_cpu_idle_state, 124
sbi_domain, 129
sbi_ecall_extension, 132
sbi_hart_ext_data, 134
sbi_hsm_device, 136
sbi_ipi_device, 136
sbi_ipi_event_ops, 137
sbi_platform, 143
sbi_pmu_device, 147
sbi_system_reset_device, 151
sbi_system_suspend_device, 151
sbi_timer_device, 152
nameoff
fdt_property, 88
nascent_init
sbi_platform_operations, 145
next
__aligned, 77
sbi_dlist, 128
next_addr
fw_dynamic_info, 89
sbi_domain, 129
sbi_hart.h, 439
sbi_scratch, 149
next_arg1
sbi_domain, 129
sbi_scratch, 149
next_mode
fw_dynamic_info, 89
sbi_domain, 129
sbi_hart.h, 440
sbi_scratch, 149
next_virt
sbi_hart.h, 440
nextprop_
fdt_ro.c, 1321
ngpio
gpio_chip, 90
node
gpio_chip, 90
i2c_adapter, 92
mbox_chan, 94
mbox_controller, 95
regmap, 100
sbi_domain, 129
sbi_irqchip_device, 138
sbi_system_reset_device, 151
node_offset
fdt_phandle_args, 86
nop_delay_fn
sbi_timer.c, 1076
nputs
sbi_console.c, 802
nputs_all
sbi_console.c, 803
Nuclei UX600 Platform, 47
NULL
sbi_types.h, 620
num_clocks
rpmi_clock_get_num_clocks_resp, 103
num_entries
sbi_fifo, 133
num_hw_ctrs
sbi_pmu.c, 1027
num_hw_events
sbi_pmu.c, 1027
num_idc
aplic_data, 80
num_ids
imsic_data, 93
num_rates
rpmi_clock_get_attributes_resp, 102
num_source
aplic_data, 80
num_src
plic_data, 98

off_dt_strings
fdt_header, 84
off_dt_struct
fdt_header, 84
off_mem_rsvmap
fdt_header, 84
offset
gpio_pin, 91
offsetof
sbi_types.h, 620
old_dev
sbi_console_test.c, 1124
OpenPiton FPGA SoC Platform, 43
OpenSBI Contribution Guideline, 7
OpenSBI Device Tree Configuration Guideline, 21
OpenSBI Domain Support, 15

OpenSBI Firmware with Dynamic Information (FW_DYNAMIC), 27
OpenSBI Firmware with Jump Address (FW_JUMP), 29
OpenSBI Firmware with Payload (FW_PAYLOAD), 31
OpenSBI Library Usage, 13
OpenSBI Platform Firmwares, 25
OpenSBI Platform Requirements, 11
OpenSBI Platform Support Guideline, 9
OpenSBI Supported Platforms, 49
OPENSBI_VERSION
    sbi_version.h, 628
opensbi_version
    sbi_platform, 143
OPENSBI_VERSION_MAJOR
    sbi_version.h, 628
OPENSBI_VERSION_MINOR
    sbi_version.h, 628
options
    fw_dynamic_info, 89
    sbi_scratch, 149
order
    sbi_domain_memregion, 131
out_sz
    sbi_console.h, 333
output
    sbi_pmu_event_info, 147
overlay_adjust_local_phandles
    fdt_overlay.c, 1278
overlay_adjust_node_phandles
    fdt_overlay.c, 1279
overlay_apply_node
    fdt_overlay.c, 1279
overlay_fixup_one_phandle
    fdt_overlay.c, 1280
overlay_fixup_phandle
    fdt_overlay.c, 1281
overlay_fixup_phandles
    fdt_overlay.c, 1281
overlay_get_target
    fdt_overlay.c, 1282
overlay_get_target_phandle
    fdt_overlay.c, 1282
overlay_merge
    fdt_overlay.c, 1283
overlay_phandle_add_offset
    fdt_overlay.c, 1284
overlay_symbol_update
    fdt_overlay.c, 1284
overlay_update_local_node_references
    fdt_overlay.c, 1285
overlay_update_local_references
    fdt_overlay.c, 1286
owner
    __aligned, 77

PAD_ALTERNATE
    sbi_console.c, 801
PAD_RIGHT
    sbi_console.c, 801
PAD_SIGN
    sbi_console.c, 801
PAD_ZERO
    sbi_console.c, 801
PAGE_MASK
    riscv_asm.h, 169
PAGE_SHIFT
    riscv_asm.h, 169
PAGE_SIZE
    riscv_asm.h, 169
PCS0_CFG_OFFSET
    atcsmu.h, 763
PCS0_CTL_OFFSET
    atcsmu.h, 763
PCS0_WE_OFFSET
    atcsmu.h, 763
PCS_CFG_DEEP_SLEEP
    atcsmu.h, 763
PCS_CFG_DEEP_SLEEP_SHIFT
    atcsmu.h, 763
PCS_CFG_LIGHT_SLEEP
    atcsmu.h, 763
PCS_CFG_LIGHT_SLEEP_SHIFT
    atcsmu.h, 763
PCS_CTL_CMD_SHIFT
    atcsmu.h, 763
PCS_CTL_PARAM_SHIFT
    atcsmu.h, 763
PCS_MAX_NR
    atcsmu.h, 763
PCSm_CFG_OFFSET
    atcsmu.h, 763
PCSm_CTL_OFFSET
    atcsmu.h, 763
PCSm_WE_OFFSET
    atcsmu.h, 764
phs_ptr_offset
    sbi_pmu.c, 1028
phys_hi
    sbi_dbtr_shmem, 126
phys_lo
    sbi_dbtr_shmem, 126
physical_addr_t
    sbi_types.h, 621
physical_size_t
    sbi_types.h, 621
pin_to_chip
    fdt_gpio_designware.c, 1181
PK_SYS_write
    htif.c, 1591
plat_info
    rpmi_base_get_platform_info_resp, 101
plat_info_len
    rpmi_base_get_platform_info_resp, 101
platform_addr
    sbi_scratch, 149
platform_ops_addr
    sbi_platform, 143

platform_uart_data, 97
addr, 97
baud, 97
freq, 97
reg_io_width, 97
reg_offset, 97
reg_shift, 97
platform_version
sbi_platform, 143
plic.c
plic_cold_irqchip_init, 1251
PLIC_CONTEXT_BASE, 1250
plic_context_init, 1252
PLIC_CONTEXT_STRIDE, 1250
plic_delegate, 1252
PLIC_ENABLE_BASE, 1251
PLIC_ENABLE_STRIDE, 1251
plic_get, 1253
plic_get_hart_data_ptr, 1251
plic_get_ie, 1253
plic_get_priority, 1253
plic_get_thresh, 1253
PLIC_PENDING_BASE, 1251
PLIC_PRIORITY_BASE, 1251
plic_ptr_offset, 1256
plic_resume, 1254
plic_set_hart_data_ptr, 1251
plic_set_ie, 1254
plic_set_priority, 1255
plic_set_thresh, 1255
plic_suspend, 1255
plic_warm_irqchip_init, 1255
THEAD_PLIC_CTRL_REG, 1251
plic.h
plic_cold_irqchip_init, 697
PLIC_DATA_SIZE, 696
PLIC_FLAG_ARIANE_BUG, 696
PLIC_FLAG_ENABLE_PM, 696
PLIC_FLAG_THEAD_DELEGATION, 696
plic_get, 697
PLIC_IE_WORDS, 696
PLIC_M_CONTEXT, 697
plic_resume, 697
PLIC_S_CONTEXT, 697
plic_suspend, 698
plic_cold_irqchip_init
plic.c, 1251
plic.h, 697
PLIC_CONTEXT_BASE
plic.c, 1250
plic_context_init
plic.c, 1252
PLIC_CONTEXT_STRIDE
plic.c, 1250
plic_data, 97
addr, 98
context_map, 98
flags, 98
irqchip, 98
num_src, 98
pm_data, 98
size, 98
PLIC_DATA_SIZE
plic.h, 696
plic_delegate
plic.c, 1252
PLIC_ENABLE_BASE
plic.c, 1251
PLIC_ENABLE_STRIDE
plic.c, 1251
PLIC_FLAG_ARIANE_BUG
plic.h, 696
PLIC_FLAG_ENABLE_PM
plic.h, 696
PLIC_FLAG_THEAD_DELEGATION
plic.h, 696
plic_get
plic.c, 1253
plic.h, 697
plic_get_hart_data_ptr
plic.c, 1251
plic_get_ie
plic.c, 1253
plic_get_priority
plic.c, 1253
plic_get_thresh
plic.c, 1253
PLIC_IE_WORDS
plic.h, 696
PLIC_M_CONTEXT
plic.h, 697
PLIC_PENDING_BASE
plic.c, 1251
PLIC_PRIORITY_BASE
plic.c, 1251
plic_ptr_offset
plic.c, 1256
plic_resume
plic.c, 1254
plic.h, 697
PLIC_S_CONTEXT
plic.h, 697
plic_set_hart_data_ptr
plic.c, 1251
plic_set_ie
plic.c, 1254
plic_set_priority
plic.c, 1255
plic_set_thresh
plic.c, 1255
plic_suspend
plic.c, 1255
plic.h, 698
plic_warm_irqchip_init
plic.c, 1255
plicsw

andes_plicsw.c, 1224
fdt_ipi_plicsw.c, 1228
plicsw_cold_ipi_init
andes_plicsw.c, 1223
andes_plicsw.h, 686
PLICSW_CONTEXT_BASE
andes_plicsw.h, 686
PLICSW_CONTEXT_CLAIM
andes_plicsw.h, 686
PLICSW_CONTEXT_STRIDE
andes_plicsw.h, 686
plicsw_data, 98
addr, 98
hart_count, 98
size, 98
PLICSW_ENABLE_BASE
andes_plicsw.h, 686
PLICSW_ENABLE_STRIDE
andes_plicsw.h, 686
plicsw_ipi
andes_plicsw.c, 1224
plicsw_ipi_clear
andes_plicsw.c, 1223
plicsw_ipi_send
andes_plicsw.c, 1224
PLICSW_PENDING_BASE
andes_plicsw.h, 686
PLICSW_PRIORITY_BASE
andes_plicsw.h, 686
PLICSW_REGION_ALIGN
andes_plicsw.h, 686
plmt
andes_plmt.c, 1603
fdt_timer_plmt.c, 1608
plmt_cold_timer_init
andes_plmt.c, 1602
andes_plmt.h, 773
plmt_data, 99
hart_count, 99
size, 99
time_cmp, 99
time_val, 99
timer_freq, 99
PLMT_REGION_ALIGN
andes_plmt.h, 772
plmt_timer
andes_plmt.c, 1603
plmt_timer_event_start
andes_plmt.c, 1602
plmt_timer_event_stop
andes_plmt.c, 1602
plmt_timer_value
andes_plmt.c, 1603
plmt_warm_timer_init
andes_plmt.c, 1603
pm_data
plic_data, 98
PMP_A
riscv_encoding.h, 268
PMP_A_NA4
riscv_encoding.h, 268
PMP_A_NAPOT
riscv_encoding.h, 268
PMP_A_TOR
riscv_encoding.h, 268
pmp_addr_bits
sbi_hart_features, 134
PMP_ADDR_MASK
riscv_encoding.h, 268
PMP_COUNT
riscv_encoding.h, 268
pmp_count
sbi_hart_features, 134
pmp_disable
riscv_asm.c, 782
riscv_asm.h, 173
pmp_get
riscv_asm.c, 783
riscv_asm.h, 174
PMP_L
riscv_encoding.h, 268
pmp_log2gran
sbi_hart_features, 134
PMP_R
riscv_encoding.h, 268
pmp_set
riscv_asm.c, 784
riscv_asm.h, 175
PMP_SHIFT
riscv_encoding.h, 268
PMP_W
riscv_encoding.h, 268
PMP_X
riscv_encoding.h, 268
pmu_add_hw_event_map
sbi_pmu.c, 1011
pmu_ctr_enable_irq_hw
sbi_pmu.c, 1011
pmu_ctr_find_fixed_hw
sbi_pmu.c, 1012
pmu_ctr_find_fw
sbi_pmu.c, 1012
pmu_ctr_find_hw
sbi_pmu.c, 1012
pmu_ctr_start_fw
sbi_pmu.c, 1013
pmu_ctr_start_hw
sbi_pmu.c, 1013
pmu_ctr_stop_fw
sbi_pmu.c, 1014
pmu_ctr_stop_hw
sbi_pmu.c, 1014
pmu_ctr_validate
sbi_pmu.c, 1015
pmu_ctr_write_hw
sbi_pmu.c, 1015

pmu_dev
sbi_pmu.c, 1028
pmu_event_range_overlap
sbi_pmu.c, 1016
pmu_event_select_overlap
sbi_pmu.c, 1016
pmu_event_validate
sbi_pmu.c, 1017
pmu_fixed_ctr_update_inhibit_bits
sbi_pmu.c, 1017
pmu_get_hart_state_ptr
sbi_pmu.c, 1011
pmu_init
sbi_platform_operations, 145
pmu_reset_event_map
sbi_pmu.c, 1017
pmu_reset_hw_mhpmevent
sbi_pmu.c, 1018
pmu_set_hart_state_ptr
sbi_pmu.c, 1011
pmu_sse_cb_ops
sbi_pmu.c, 1028
pmu_sse_complete
sbi_pmu.c, 1018
pmu_sse_disable
sbi_pmu.c, 1018
pmu_sse_enable
sbi_pmu.c, 1018
pmu_thishart_state_ptr
sbi_pmu.c, 1011
pmu_update_hw_mhpmevent
sbi_pmu.c, 1019
pmu_update_inhibit_flags
sbi_pmu.c, 1019
pmu_xlate_to_mhpmevent
sbi_platform_operations, 145
possible_harts
sbi_domain, 129
poweroff
fdt_reset_gpio.c, 1514
fdt_reset_syscon.c, 1528
PRECISION_D
riscv_fp.h, 292
PRECISION_S
riscv_fp.h, 292
prev
sbi_dlist, 128
prev_context
sbi_trap_context, 154
print
sbi_console.c, 804
PRINT_BUF_LEN
sbi_console.c, 801
printc
sbi_console.c, 805
PRINTF_TEST
sbi_console_test.c, 1121
printf_test
sbi_console_test.c, 1122
printi
sbi_console.c, 807
prints
sbi_console.c, 809
priv_version
sbi_hart_features, 135
probe
sbi_ecall_extension, 132
probed
sbi_dbtr_hart_triggers_state, 125
process
sbi_ipi_event_ops, 137
PRV_M
riscv_encoding.h, 268
PRV_S
riscv_encoding.h, 269
PRV_U
riscv_encoding.h, 269
putc_test
sbi_console_test.c, 1122
PUTS_TEST
sbi_console_test.c, 1122
puts_test
sbi_console_test.c, 1123

q
uu, 159
qdivrem.c
__FBSDID, 1452
__qdivrem, 1452
B, 1451
COMBINE, 1451
digit, 1451
shl, 1452
QEMU RISC-V Virt Machine Platform, 51
qlock
sbi_fifo, 133
qshift_t
quad.h, 1455
quad.h
__cmpdi2, 1455
__divdi3, 1455
__moddi3, 1455
__qdivrem, 1455
__ucmpdi2, 1456
__udivdi3, 1456
__umoddi3, 1457
H, 1454
HALF_BITS, 1454
HHALF, 1454
L, 1454
LHALF, 1454
LHUP, 1454
LONG_BITS, 1454
qshift_t, 1455
QUAD_BITS, 1455
QUAD_BITS
quad.h, 1455

quad_t
types.h, 1449
queue
sbi_fifo, 133

R_RISCV_RELATIVE
riscv_elf.h, 197
ra
sbi_trap_regs, 156
read
i2c_adapter, 92
read_attributes
sbi_mpxy_channel, 139
readb
riscv_io.h, 295
readb_relaxed
riscv_io.h, 295
readl
riscv_io.h, 295
readl_relaxed
riscv_io.h, 295
README.md, 1608
readq
riscv_io.h, 296
readq_relaxed
riscv_io.h, 296
readw
riscv_io.h, 296
readw_relaxed
riscv_io.h, 296
reboot
fdt_reset_syscon.c, 1529
reg_base
regmap, 100
REG_BAUD
shakti-uart.c, 1568
REG_BOARD_TYPE
fdt_reset_sg2042_hwmon_mcu.c, 1520
REG_CMD
fdt_reset_sg2042_hwmon_mcu.c, 1521
REG_CONTROL
shakti-uart.c, 1568
REG_DELAY
shakti-uart.c, 1568
reg_id
rpmi_cppc_probe_req, 110
rpmi_cppc_read_reg_req, 110
rpmi_cppc_write_reg_req, 111
REG_INT_EN
shakti-uart.c, 1568
reg_io_width
platform_uart_data, 97
REG_IQ_CYCLES
shakti-uart.c, 1568
REG_L
riscv_asm.h, 169
reg_len
rpmi_cppc_probe_resp, 110
REG_MASK
riscv_encoding.h, 269
reg_max
regmap, 100
REG_OFFSET
riscv_encoding.h, 269
reg_offset
platform_uart_data, 97
REG_PTR
riscv_encoding.h, 269
reg_read
regmap, 100
REG_RX
shakti-uart.c, 1568
REG_RX_THRES
shakti-uart.c, 1568
REG_S
riscv_asm.h, 169
reg_shift
platform_uart_data, 97
regmap, 100
REG_STATUS
shakti-uart.c, 1568
reg_stride
regmap, 100
REG_TX
shakti-uart.c, 1568
reg_update_bits
regmap, 100
reg_write
regmap, 100
REGBYTES
riscv_encoding.h, 269
region_addr_hi
rpmi_cppc_get_fastchan_region_resp, 108
region_addr_lo
rpmi_cppc_get_fastchan_region_resp, 108
region_size_hi
rpmi_cppc_get_fastchan_region_resp, 108
region_size_lo
rpmi_cppc_get_fastchan_region_resp, 108
regions
sbi_domain, 129
register_cb
sbi_sse_cb_ops, 150
register_extensions
sbi_ecall_extension, 132
regmap, 99
id, 100
node, 100
reg_base, 100
reg_max, 100
reg_read, 100
reg_shift, 100
reg_stride, 100
reg_update_bits, 100
reg_write, 100
regmap.c
regmap_add, 1501

regmap_find, 1502
regmap_read, 1502
regmap_reg_addr, 1503
regmap_reg_valid, 1503
regmap_remove, 1503
regmap_update_bits, 1504
regmap_write, 1504
SBI_LIST_HEAD, 1505
regmap.h
regmap_add, 737
regmap_find, 738
regmap_read, 738
regmap_remove, 739
regmap_update_bits, 739
regmap_write, 740
to_regmap, 740
regmap_add
regmap.c, 1501
regmap.h, 737
regmap_find
regmap.c, 1502
regmap.h, 738
regmap_read
regmap.c, 1502
regmap.h, 738
regmap_reg_addr
regmap.c, 1503
regmap_reg_valid
regmap.c, 1503
regmap_remove
regmap.c, 1503
regmap.h, 739
regmap_syscon_init
fdt_regmap_syscon.c, 1495
regmap_syscon_match
fdt_regmap_syscon.c, 1500
regmap_syscon_read_16
fdt_regmap_syscon.c, 1495
regmap_syscon_read_32
fdt_regmap_syscon.c, 1496
regmap_syscon_read_8
fdt_regmap_syscon.c, 1496
regmap_syscon_read_be16
fdt_regmap_syscon.c, 1496
regmap_syscon_read_be32
fdt_regmap_syscon.c, 1497
regmap_syscon_read_le16
fdt_regmap_syscon.c, 1497
regmap_syscon_read_le32
fdt_regmap_syscon.c, 1497
regmap_syscon_write_16
fdt_regmap_syscon.c, 1498
regmap_syscon_write_32
fdt_regmap_syscon.c, 1498
regmap_syscon_write_8
fdt_regmap_syscon.c, 1498
regmap_syscon_write_be16
fdt_regmap_syscon.c, 1499
regmap_syscon_write_be32
fdt_regmap_syscon.c, 1499
regmap_syscon_write_le16
fdt_regmap_syscon.c, 1499
regmap_syscon_write_le32
fdt_regmap_syscon.c, 1500
regmap_update_bits
regmap.c, 1504
regmap.h, 739
regmap_write
regmap.c, 1504
regmap.h, 740
regs
imsic_data, 93
sbi_trap_context, 154
remaining
rpmi_clock_get_supported_rates_resp, 105
rpmi_cppc_hart_list_resp, 109
rpmi_hsm_get_hart_list_resp, 113
rpmi_hsm_get_susp_types_resp, 116
Renesas RZ/Five SoC (R9A07G043F) Platform, 53
renesas-scif.h
renesas_scif_init, 750
renesas_scif.c
get_reg, 1559
renesas_scif_console, 1561
renesas_scif_init, 1560
renesas_scif_putc, 1560
SCBRR_VALUE, 1558
scif_base, 1561
SCIF_FCR_RFRST, 1558
SCIF_FCR_RST_ASSRT_RFTF, 1558
SCIF_FCR_RST_NGATE_RFTF, 1558
SCIF_FCR_TFRST, 1558
SCIF_FSR_BRK, 1558
SCIF_FSR_DR, 1558
SCIF_FSR_ER, 1558
SCIF_FSR_TDFE, 1558
SCIF_FSR_TEND, 1558
SCIF_FSR_TXD_CHK, 1558
SCIF_REG_8BIT, 1558
SCIF_REG_BRR, 1558
SCIF_REG_FCR, 1558
SCIF_REG_FSR, 1559
SCIF_REG_FTDR, 1559
SCIF_REG_LSR, 1559
SCIF_REG_SCR, 1559
SCIF_REG_SEMR, 1559
SCIF_REG_SMR, 1559
SCIF_SCR_RCV_TRN_DIS, 1559
SCIF_SCR_RCV_TRN_EN, 1559
SCIF_SCR_RE, 1559
SCIF_SCR_TE, 1559
SCIF_SEMR_MDDRS, 1559
set_reg, 1561
renesas_scif_console
renesas_scif.c, 1561
renesas_scif_init

renesas-scif.h, 750
renesas_scif.c, 1560
renesas_scif_putc
renesas_scif.c, 1560
request_chan
mbox_controller, 95
reserved
sbi_platform, 143
RESET_VEC_8CORE_OFFSET
atcsmu.h, 764
RESET_VEC_HI_OFFSET
atcsmu.h, 764
RESET_VEC_LO_OFFSET
atcsmu.h, 764
restart
fdt_reset_gpio.c, 1514
resume_addr_hi
rpmi_hsm_hart_susp_req, 117
rpmi_syssusp_suspend_req, 122
resume_addr_lo
rpmi_hsm_hart_susp_req, 118
rpmi_syssusp_suspend_req, 122
returned
rpmi_clock_get_supported_rates_resp, 105
rpmi_cppc_hart_list_resp, 109
rpmi_hsm_get_hart_list_resp, 113
rpmi_hsm_get_susp_types_resp, 116
RISC-V Open Source Supervisor Binary Interface (OpenSBI), 1
RISCV_ACQUIRE_BARRIER
riscv_barrier.h, 185
riscv_asm.c
csr_read_num, 778
csr_write_num, 779
ctz, 779
is_pmp_entry_mapped, 780
misa_extension_imp, 780
misa_string, 781
misa_xlen, 781
pmp_disable, 782
pmp_get, 783
pmp_set, 784
switchcase_csr_read, 776
switchcase_csr_read_16, 776
switchcase_csr_read_2, 776
switchcase_csr_read_32, 776
switchcase_csr_read_4, 777
switchcase_csr_read_64, 777
switchcase_csr_read_8, 777
switchcase_csr_write, 777
switchcase_csr_write_16, 777
switchcase_csr_write_2, 777
switchcase_csr_write_32, 777
switchcase_csr_write_4, 778
switchcase_csr_write_64, 778
switchcase_csr_write_8, 778
riscv_asm.h
__ASM_STR, 167
csr_clear, 167
csr_read, 167
csr_read_clear, 167
csr_read_num, 169
csr_read_relaxed, 167
csr_read_set, 168
csr_set, 168
csr_swap, 168
csr_write, 168
csr_write_num, 170
current_hartid, 168
ebreak, 168
is_pmp_entry_mapped, 171
LGREG, 169
misa_extension, 169
misa_extension_imp, 171
misa_string, 172
misa_xlen, 172
PAGE_MASK, 169
PAGE_SHIFT, 169
PAGE_SIZE, 169
pmp_disable, 173
pmp_get, 174
pmp_set, 175
REG_L, 169
REG_S, 169
SZREG, 169
wfi, 169
riscv_atomic.c
__axchg, 785
atomic_add_return, 786
atomic_clear_bit, 786
atomic_cmpxchg, 787
atomic_raw_clear_bit, 787
atomic_raw_set_bit, 788
atomic_raw_xchg_uint, 788
atomic_raw_xchg_ulong, 788
atomic_read, 788
atomic_set_bit, 789
atomic_sub_return, 789
atomic_write, 790
atomic_xchg, 790
axchg, 786
riscv_atomic.h
atomic_add_return, 179
atomic_clear_bit, 179
atomic_cmpxchg, 180
ATOMIC_INIT, 179
ATOMIC_INITIALIZER, 179
atomic_raw_clear_bit, 180
atomic_raw_set_bit, 181
atomic_raw_xchg_uint, 181
atomic_raw_xchg_ulong, 181
atomic_read, 181
atomic_set_bit, 182
atomic_sub_return, 182
atomic_write, 183
atomic_xchg, 183

riscv_atomic_test.c
add_return_test, 1111
atomic_clear_bit_test, 1112
atomic_raw_clear_bit_test, 1112
atomic_raw_set_bit_test, 1112
atomic_rw_test, 1113
atomic_set_bit_test, 1113
ATOMIC_TEST_BIT_NUM, 1111
atomic_test_cases, 1115
ATOMIC_TEST_RAW_BIT_CELL, 1111
ATOMIC_TEST_RAW_BIT_NUM, 1111
atomic_test_suite, 1115
atomic_test_suite_init, 1113
ATOMIC_TEST_VAL1, 1111
ATOMIC_TEST_VAL2, 1111
ATOMIC_TEST_VAL3, 1111
atomic_xchg_test, 1113
cmpxchg_test, 1114
sub_return_test, 1114
test_atomic, 1115
riscv_barrier.h
__smp_load_acquire, 185
__smp_store_release, 185
cpu_relax, 185
mb, 185
RISCV_ACQUIRE_BARRIER, 185
RISCV_FENCE, 185
RISCV_FENCE_I, 186
RISCV_RELEASE_BARRIER, 186
rmb, 186
smp_mb, 186
smp_rmb, 186
smp_wmb, 186
wmb, 186
riscv_dbtr.h
CLEAR_DBTR_BIT, 189
riscv_dbtr_tdata1_mcontrol6_t, 191
riscv_dbtr_tdata1_mcontrol_t, 191
riscv_dbtr_tdata1_t, 191
RISCV_DBTR_TRIG_ETRIGGER, 192
RISCV_DBTR_TRIG_ICOUNT, 192
RISCV_DBTR_TRIG_ITRIGGER, 192
RISCV_DBTR_TRIG_LEGACY, 192
RISCV_DBTR_TRIG_MCONTROL, 192
RISCV_DBTR_TRIG_MCONTROL6, 192
RISCV_DBTR_TRIG_NONE, 191
RV_DBTR_BIT, 189
RV_DBTR_BIT_MASK, 189
RV_DBTR_DECLARE_BIT, 189, 192, 193
RV_DBTR_DECLARE_BIT_MASK, 189, 192–194
RV_DBTR_SET_MC6_EXEC, 189
RV_DBTR_SET_MC6_LOAD, 189
RV_DBTR_SET_MC6_SIZE, 189
RV_DBTR_SET_MC6_STORE, 190
RV_DBTR_SET_MC6_TYPE, 190
RV_DBTR_SET_MC_EXEC, 190
RV_DBTR_SET_MC_LOAD, 190
RV_DBTR_SET_MC_SIZEHI, 190
RV_DBTR_SET_MC_SIZELO, 190
RV_DBTR_SET_MC_STORE, 190
RV_DBTR_SET_MC_TYPE, 191
RV_DBTR_SET_TDATA1_TYPE, 191
RV_MAX_TRIGGERS, 191
SET_DBTR_BIT, 191
riscv_dbtr_tdata1_mcontrol6_t
riscv_dbtr.h, 191
riscv_dbtr_tdata1_mcontrol_t
riscv_dbtr.h, 191
riscv_dbtr_tdata1_t
riscv_dbtr.h, 191
RISCV_DBTR_TRIG_ETRIGGER
riscv_dbtr.h, 192
RISCV_DBTR_TRIG_ICOUNT
riscv_dbtr.h, 192
RISCV_DBTR_TRIG_ITRIGGER
riscv_dbtr.h, 192
RISCV_DBTR_TRIG_LEGACY
riscv_dbtr.h, 192
RISCV_DBTR_TRIG_MCONTROL
riscv_dbtr.h, 192
RISCV_DBTR_TRIG_MCONTROL6
riscv_dbtr.h, 192
RISCV_DBTR_TRIG_NONE
riscv_dbtr.h, 191
riscv_elf.h
R_RISCV_RELATIVE, 197
riscv_encoding.h
CAUSE_BREAKPOINT, 213
CAUSE_DOUBLE_TRAP, 213
CAUSE_FETCH_ACCESS, 214
CAUSE_FETCH_GUEST_PAGE_FAULT, 214
CAUSE_FETCH_PAGE_FAULT, 214
CAUSE_ILLEGAL_INSTRUCTION, 214
CAUSE_LOAD_ACCESS, 214
CAUSE_LOAD_GUEST_PAGE_FAULT, 214
CAUSE_LOAD_PAGE_FAULT, 214
CAUSE_MACHINE_ECALL, 214
CAUSE_MISALIGNED_FETCH, 214
CAUSE_MISALIGNED_LOAD, 214
CAUSE_MISALIGNED_STORE, 214
CAUSE_STORE_ACCESS, 214
CAUSE_STORE_GUEST_PAGE_FAULT, 214
CAUSE_STORE_PAGE_FAULT, 214
CAUSE_SUPERVISOR_ECALL, 214
CAUSE_SW_CHECK_EXCP, 215
CAUSE_USER_ECALL, 215
CAUSE_VIRTUAL_INST_FAULT, 215
CAUSE_VIRTUAL_SUPERVISOR_ECALL, 215
CSR_CYCLE, 215
CSR_CYCLEH, 215
CSR_DCSR, 215
CSR_DPC, 215
CSR_DSCRATCH0, 215
CSR_DSCRATCH1, 215
CSR_FCSR, 215
CSR_FFLAGS, 215

CSR_FRM, 215
CSR_HCOUNTEREN, 215
CSR_HEDELEG, 215
CSR_HENVCFG, 216
CSR_HENVCFGH, 216
CSR_HGATP, 216
CSR_HGEIE, 216
CSR_HGEIP, 216
CSR_HIDELEG, 216
CSR_HIDELEGH, 216
CSR_HIE, 216
CSR_HIP, 216
CSR_HPMCOUNTER10, 216
CSR_HPMCOUNTER10H, 216
CSR_HPMCOUNTER11, 216
CSR_HPMCOUNTER11H, 216
CSR_HPMCOUNTER12, 216
CSR_HPMCOUNTER12H, 216
CSR_HPMCOUNTER13, 217
CSR_HPMCOUNTER13H, 217
CSR_HPMCOUNTER14, 217
CSR_HPMCOUNTER14H, 217
CSR_HPMCOUNTER15, 217
CSR_HPMCOUNTER15H, 217
CSR_HPMCOUNTER16, 217
CSR_HPMCOUNTER16H, 217
CSR_HPMCOUNTER17, 217
CSR_HPMCOUNTER17H, 217
CSR_HPMCOUNTER18, 217
CSR_HPMCOUNTER18H, 217
CSR_HPMCOUNTER19, 217
CSR_HPMCOUNTER19H, 217
CSR_HPMCOUNTER20, 217
CSR_HPMCOUNTER20H, 218
CSR_HPMCOUNTER21, 218
CSR_HPMCOUNTER21H, 218
CSR_HPMCOUNTER22, 218
CSR_HPMCOUNTER22H, 218
CSR_HPMCOUNTER23, 218
CSR_HPMCOUNTER23H, 218
CSR_HPMCOUNTER24, 218
CSR_HPMCOUNTER24H, 218
CSR_HPMCOUNTER25, 218
CSR_HPMCOUNTER25H, 218
CSR_HPMCOUNTER26, 218
CSR_HPMCOUNTER26H, 218
CSR_HPMCOUNTER27, 218
CSR_HPMCOUNTER27H, 218
CSR_HPMCOUNTER28, 219
CSR_HPMCOUNTER28H, 219
CSR_HPMCOUNTER29, 219
CSR_HPMCOUNTER29H, 219
CSR_HPMCOUNTER3, 219
CSR_HPMCOUNTER30, 219
CSR_HPMCOUNTER30H, 219
CSR_HPMCOUNTER31, 219
CSR_HPMCOUNTER31H, 219
CSR_HPMCOUNTER3H, 219
CSR_HPMCOUNTER4, 219
CSR_HPMCOUNTER4H, 219
CSR_HPMCOUNTER5, 219
CSR_HPMCOUNTER5H, 219
CSR_HPMCOUNTER6, 219
CSR_HPMCOUNTER6H, 220
CSR_HPMCOUNTER7, 220
CSR_HPMCOUNTER7H, 220
CSR_HPMCOUNTER8, 220
CSR_HPMCOUNTER8H, 220
CSR_HPMCOUNTER9, 220
CSR_HPMCOUNTER9H, 220
CSR_HSTATEEN0, 220
CSR_HSTATEEN0H, 220
CSR_HSTATEEN1, 220
CSR_HSTATEEN1H, 220
CSR_HSTATEEN2, 220
CSR_HSTATEEN2H, 220
CSR_HSTATEEN3, 220
CSR_HSTATEEN3H, 220
CSR_HSTATUS, 221
CSR_HTIMEDELTA, 221
CSR_HTIMEDELTAH, 221
CSR_HTINST, 221
CSR_HTVAL, 221
CSR_HVICTL, 221
CSR_HVIEN, 221
CSR_HVIENH, 221
CSR_HVIP, 221
CSR_HVIPH, 221
CSR_HVIPRIO1, 221
CSR_HVIPRIO1H, 221
CSR_HVIPRIO2, 221
CSR_HVIPRIO2H, 221
CSR_INSTRET, 221
CSR_INSTRETH, 222
CSR_MARCHID, 222
CSR_MCAUSE, 222
CSR_MCONFIGPTR, 222
CSR_MCOUNTEREN, 222
CSR_MCOUNTINHIBIT, 222
CSR_MCYCLE, 222
CSR_MCYCLECFG, 222
CSR_MCYCLECFGH, 222
CSR_MCYCLEH, 222
CSR_MEDELEG, 222
CSR_MENVCFG, 222
CSR_MENVCFGH, 222
CSR_MEPC, 222
CSR_MHARTID, 222
CSR_MHPMCOUNTER10, 223
CSR_MHPMCOUNTER10H, 223
CSR_MHPMCOUNTER11, 223
CSR_MHPMCOUNTER11H, 223
CSR_MHPMCOUNTER12, 223
CSR_MHPMCOUNTER12H, 223
CSR_MHPMCOUNTER13, 223
CSR_MHPMCOUNTER13H, 223

CSR_MHPMCOUNTER14, 223
CSR_MHPMCOUNTER14H, 223
CSR_MHPMCOUNTER15, 223
CSR_MHPMCOUNTER15H, 223
CSR_MHPMCOUNTER16, 223
CSR_MHPMCOUNTER16H, 223
CSR_MHPMCOUNTER17, 223
CSR_MHPMCOUNTER17H, 224
CSR_MHPMCOUNTER18, 224
CSR_MHPMCOUNTER18H, 224
CSR_MHPMCOUNTER19, 224
CSR_MHPMCOUNTER19H, 224
CSR_MHPMCOUNTER20, 224
CSR_MHPMCOUNTER20H, 224
CSR_MHPMCOUNTER21, 224
CSR_MHPMCOUNTER21H, 224
CSR_MHPMCOUNTER22, 224
CSR_MHPMCOUNTER22H, 224
CSR_MHPMCOUNTER23, 224
CSR_MHPMCOUNTER23H, 224
CSR_MHPMCOUNTER24, 224
CSR_MHPMCOUNTER24H, 224
CSR_MHPMCOUNTER25, 225
CSR_MHPMCOUNTER25H, 225
CSR_MHPMCOUNTER26, 225
CSR_MHPMCOUNTER26H, 225
CSR_MHPMCOUNTER27, 225
CSR_MHPMCOUNTER27H, 225
CSR_MHPMCOUNTER28, 225
CSR_MHPMCOUNTER28H, 225
CSR_MHPMCOUNTER29, 225
CSR_MHPMCOUNTER29H, 225
CSR_MHPMCOUNTER3, 225
CSR_MHPMCOUNTER30, 225
CSR_MHPMCOUNTER30H, 225
CSR_MHPMCOUNTER31, 225
CSR_MHPMCOUNTER31H, 225
CSR_MHPMCOUNTER3H, 226
CSR_MHPMCOUNTER4, 226
CSR_MHPMCOUNTER4H, 226
CSR_MHPMCOUNTER5, 226
CSR_MHPMCOUNTER5H, 226
CSR_MHPMCOUNTER6, 226
CSR_MHPMCOUNTER6H, 226
CSR_MHPMCOUNTER7, 226
CSR_MHPMCOUNTER7H, 226
CSR_MHPMCOUNTER8, 226
CSR_MHPMCOUNTER8H, 226
CSR_MHPMCOUNTER9, 226
CSR_MHPMCOUNTER9H, 226
CSR_MHPMEVENT10, 226
CSR_MHPMEVENT10H, 226
CSR_MHPMEVENT11, 227
CSR_MHPMEVENT11H, 227
CSR_MHPMEVENT12, 227
CSR_MHPMEVENT12H, 227
CSR_MHPMEVENT13, 227
CSR_MHPMEVENT13H, 227
CSR_MHPMEVENT14, 227
CSR_MHPMEVENT14H, 227
CSR_MHPMEVENT15, 227
CSR_MHPMEVENT15H, 227
CSR_MHPMEVENT16, 227
CSR_MHPMEVENT16H, 227
CSR_MHPMEVENT17, 227
CSR_MHPMEVENT17H, 227
CSR_MHPMEVENT18, 227
CSR_MHPMEVENT18H, 228
CSR_MHPMEVENT19, 228
CSR_MHPMEVENT19H, 228
CSR_MHPMEVENT20, 228
CSR_MHPMEVENT20H, 228
CSR_MHPMEVENT21, 228
CSR_MHPMEVENT21H, 228
CSR_MHPMEVENT22, 228
CSR_MHPMEVENT22H, 228
CSR_MHPMEVENT23, 228
CSR_MHPMEVENT23H, 228
CSR_MHPMEVENT24, 228
CSR_MHPMEVENT24H, 228
CSR_MHPMEVENT25, 228
CSR_MHPMEVENT25H, 228
CSR_MHPMEVENT26, 229
CSR_MHPMEVENT26H, 229
CSR_MHPMEVENT27, 229
CSR_MHPMEVENT27H, 229
CSR_MHPMEVENT28, 229
CSR_MHPMEVENT28H, 229
CSR_MHPMEVENT29, 229
CSR_MHPMEVENT29H, 229
CSR_MHPMEVENT3, 229
CSR_MHPMEVENT30, 229
CSR_MHPMEVENT30H, 229
CSR_MHPMEVENT31, 229
CSR_MHPMEVENT31H, 229
CSR_MHPMEVENT3H, 229
CSR_MHPMEVENT4, 229
CSR_MHPMEVENT4H, 230
CSR_MHPMEVENT5, 230
CSR_MHPMEVENT5H, 230
CSR_MHPMEVENT6, 230
CSR_MHPMEVENT6H, 230
CSR_MHPMEVENT7, 230
CSR_MHPMEVENT7H, 230
CSR_MHPMEVENT8, 230
CSR_MHPMEVENT8H, 230
CSR_MHPMEVENT9, 230
CSR_MHPMEVENT9H, 230
CSR_MIDELEG, 230
CSR_MIDELEGH, 230
CSR_MIE, 230
CSR_MIEH, 230
CSR_MIMPID, 231
CSR_MINSTRET, 231
CSR_MINSTRETCFG, 231
CSR_MINSTRETCFGH, 231

CSR_MINSTRETH, 231
CSR_MIP, 231
CSR_MIPH, 231
CSR_MIREG, 231
CSR_MIREG2, 231
CSR_MIREG3, 231
CSR_MIREG4, 231
CSR_MIREG5, 231
CSR_MIREG6, 231
CSR_MISA, 231
CSR_MISELECT, 231
CSR_MSCRATCH, 232
CSR_MSECCFG, 232
CSR_MSECCFGH, 232
CSR_MSTATEEN0, 232
CSR_MSTATEEN0H, 232
CSR_MSTATEEN1, 232
CSR_MSTATEEN1H, 232
CSR_MSTATEEN2, 232
CSR_MSTATEEN2H, 232
CSR_MSTATEEN3, 232
CSR_MSTATEEN3H, 232
CSR_MSTATUS, 232
CSR_MSTATUSH, 232
CSR_MTINST, 232
CSR_MTOPEI, 232
CSR_MTOPI, 233
CSR_MTVAL, 233
CSR_MTVAL2, 233
CSR_MTVEC, 233
CSR_MVENDORID, 233
CSR_MVIEN, 233
CSR_MVIENH, 233
CSR_MVIP, 233
CSR_MVIPH, 233
CSR_PMPADDR0, 233
CSR_PMPADDR1, 233
CSR_PMPADDR10, 233
CSR_PMPADDR11, 233
CSR_PMPADDR12, 233
CSR_PMPADDR13, 233
CSR_PMPADDR14, 234
CSR_PMPADDR15, 234
CSR_PMPADDR16, 234
CSR_PMPADDR17, 234
CSR_PMPADDR18, 234
CSR_PMPADDR19, 234
CSR_PMPADDR2, 234
CSR_PMPADDR20, 234
CSR_PMPADDR21, 234
CSR_PMPADDR22, 234
CSR_PMPADDR23, 234
CSR_PMPADDR24, 234
CSR_PMPADDR25, 234
CSR_PMPADDR26, 234
CSR_PMPADDR27, 234
CSR_PMPADDR28, 235
CSR_PMPADDR29, 235
CSR_PMPADDR3, 235
CSR_PMPADDR30, 235
CSR_PMPADDR31, 235
CSR_PMPADDR32, 235
CSR_PMPADDR33, 235
CSR_PMPADDR34, 235
CSR_PMPADDR35, 235
CSR_PMPADDR36, 235
CSR_PMPADDR37, 235
CSR_PMPADDR38, 235
CSR_PMPADDR39, 235
CSR_PMPADDR4, 235
CSR_PMPADDR40, 235
CSR_PMPADDR41, 236
CSR_PMPADDR42, 236
CSR_PMPADDR43, 236
CSR_PMPADDR44, 236
CSR_PMPADDR45, 236
CSR_PMPADDR46, 236
CSR_PMPADDR47, 236
CSR_PMPADDR48, 236
CSR_PMPADDR49, 236
CSR_PMPADDR5, 236
CSR_PMPADDR50, 236
CSR_PMPADDR51, 236
CSR_PMPADDR52, 236
CSR_PMPADDR53, 236
CSR_PMPADDR54, 236
CSR_PMPADDR55, 237
CSR_PMPADDR56, 237
CSR_PMPADDR57, 237
CSR_PMPADDR58, 237
CSR_PMPADDR59, 237
CSR_PMPADDR6, 237
CSR_PMPADDR60, 237
CSR_PMPADDR61, 237
CSR_PMPADDR62, 237
CSR_PMPADDR63, 237
CSR_PMPADDR7, 237
CSR_PMPADDR8, 237
CSR_PMPADDR9, 237
CSR_PMPCFG0, 237
CSR_PMPCFG1, 237
CSR_PMPCFG10, 238
CSR_PMPCFG11, 238
CSR_PMPCFG12, 238
CSR_PMPCFG13, 238
CSR_PMPCFG14, 238
CSR_PMPCFG15, 238
CSR_PMPCFG2, 238
CSR_PMPCFG3, 238
CSR_PMPCFG4, 238
CSR_PMPCFG5, 238
CSR_PMPCFG6, 238
CSR_PMPCFG7, 238
CSR_PMPCFG8, 238
CSR_PMPCFG9, 238
CSR_SATP, 238

CSR_SCAUSE, 239
CSR_SCOUNTEREN, 239
CSR_SCOUNTINHIBIT, 239
CSR_SCOUNTOVF, 239
CSR_SENVCFG, 239
CSR_SEPC, 239
CSR_SIE, 239
CSR_SIEH, 239
CSR_SIP, 239
CSR_SIPH, 239
CSR_SIREG, 239
CSR_SIREG2, 239
CSR_SIREG3, 239
CSR_SIREG4, 239
CSR_SIREG5, 239
CSR_SIREG6, 240
CSR_SISELECT, 240
CSR_SSCRATCH, 240
CSR_SSP, 240
CSR_SSTATEEN0, 240
CSR_SSTATEEN1, 240
CSR_SSTATEEN2, 240
CSR_SSTATEEN3, 240
CSR_SSTATUS, 240
CSR_STIMECMP, 240
CSR_STIMECMPH, 240
CSR_STOPEI, 240
CSR_STOPI, 240
CSR_STVAL, 240
CSR_STVEC, 240
CSR_TDATA1, 241
CSR_TDATA2, 241
CSR_TDATA3, 241
CSR_TIME, 241
CSR_TIMEH, 241
CSR_TINFO, 241
CSR_TSELECT, 241
CSR_UCAUSE, 241
CSR_UEPC, 241
CSR_UIE, 241
CSR_UIP, 241
CSR_USCRATCH, 241
CSR_USTATUS, 241
CSR_UTVAL, 241
CSR_UTVEC, 241
CSR_VL, 242
CSR_VLENB, 242
CSR_VSATP, 242
CSR_VSCAUSE, 242
CSR_VSEPC, 242
CSR_VSIE, 242
CSR_VSIEH, 242
CSR_VSIP, 242
CSR_VSIPH, 242
CSR_VSIREG, 242
CSR_VSIREG2, 242
CSR_VSIREG3, 242
CSR_VSIREG4, 242
CSR_VSIREG5, 242
CSR_VSIREG6, 242
CSR_VSISELECT, 243
CSR_VSSCRATCH, 243
CSR_VSSTATUS, 243
CSR_VSTART, 243
CSR_VSTOPEI, 243
CSR_VSTOPI, 243
CSR_VSTVAL, 243
CSR_VSTVEC, 243
CSR_VTYPE, 243
CSRRC, 243
CSRRCI, 243
CSRRS, 243
CSRRSI, 243
CSRRW, 243
CSRRWI, 243
ENVCFG_ADUE, 244
ENVCFG_ADUE_SHIFT, 244
ENVCFG_CBCFE, 244
ENVCFG_CBIE, 244
ENVCFG_CBIE_FLUSH, 244
ENVCFG_CBIE_ILL, 244
ENVCFG_CBIE_INV, 244
ENVCFG_CBIE_SHIFT, 244
ENVCFG_CBZE, 244
ENVCFG_CDE, 244
ENVCFG_DTE, 244
ENVCFG_DTE_SHIFT, 244
ENVCFG_FIOM, 244
ENVCFG_LPE, 244
ENVCFG_LPE_SHIFT, 244
ENVCFG_PBMTE, 245
ENVCFG_PMM, 245
ENVCFG_PMM_PMLEN_0, 245
ENVCFG_PMM_PMLEN_16, 245
ENVCFG_PMM_PMLEN_7, 245
ENVCFG_SSE, 245
ENVCFG_SSE_SHIFT, 245
ENVCFG_STCE, 245
GET_CSR_NUM, 245
GET_EMUL, 245
GET_LEN, 245
GET_MEW, 245
GET_NF, 245
GET_RM, 245
GET_RS1, 246
GET_RS1_NUM, 246
GET_RS1S, 246
GET_RS2, 246
GET_RS2C, 246
GET_RS2S, 246
GET_SP, 246
GET_VD, 246
GET_VEMUL, 246
GET_VIEW, 246
GET_VLMUL, 246
GET_VS2, 247

GET_VSEW, 247
HGATP32_MODE_SHIFT, 247
HGATP32_PPN, 247
HGATP32_VMID_MASK, 247
HGATP32_VMID_SHIFT, 247
HGATP64_MODE_SHIFT, 247
HGATP64_PPN, 247
HGATP64_VMID_MASK, 247
HGATP64_VMID_SHIFT, 247
HGATP_MODE_OFF, 247
HGATP_MODE_SHIFT, 247
HGATP_MODE_SV32X4, 247
HGATP_MODE_SV39X4, 247
HGATP_MODE_SV48X4, 248
HGATP_PPN, 248
HGATP_VMID_MASK, 248
HGATP_VMID_SHIFT, 248
HSTATUS_GVA, 248
HSTATUS_HU, 248
HSTATUS_SPV, 248
HSTATUS_SPVP, 248
HSTATUS_VGEIN, 248
HSTATUS_VGEIN_SHIFT, 248
HSTATUS_VSBE, 248
HSTATUS_VTSR, 248
HSTATUS_VTVM, 248
HSTATUS_VTW, 248
IMM_I, 248
IMM_S, 249
INSN_16BIT_MASK, 249
INSN_32BIT_MASK, 249
INSN_IS_16BIT, 249
INSN_IS_32BIT, 249
INSN_LEN, 249
INSN_MASK_C_FLD, 249
INSN_MASK_C_FLDSP, 249
INSN_MASK_C_FLW, 249
INSN_MASK_C_FLWSP, 249
INSN_MASK_C_FSD, 249
INSN_MASK_C_FSDSP, 249
INSN_MASK_C_FSW, 249
INSN_MASK_C_FSWSP, 250
INSN_MASK_C_LD, 250
INSN_MASK_C_LDSP, 250
INSN_MASK_C_LH, 250
INSN_MASK_C_LHU, 250
INSN_MASK_C_LW, 250
INSN_MASK_C_LWSP, 250
INSN_MASK_C_SD, 250
INSN_MASK_C_SDSP, 250
INSN_MASK_C_SH, 250
INSN_MASK_C_SW, 250
INSN_MASK_C_SWSP, 250
INSN_MASK_FENCE_TSO, 250
INSN_MASK_FLD, 250
INSN_MASK_FLQ, 250
INSN_MASK_FLW, 251
INSN_MASK_FSD, 251
INSN_MASK_FSQ, 251
INSN_MASK_FSW, 251
INSN_MASK_LB, 251
INSN_MASK_LBU, 251
INSN_MASK_LD, 251
INSN_MASK_LH, 251
INSN_MASK_LHU, 251
INSN_MASK_LW, 251
INSN_MASK_LWU, 251
INSN_MASK_SB, 251
INSN_MASK_SD, 251
INSN_MASK_SH, 251
INSN_MASK_SW, 251
INSN_MASK_VECTOR_FAULT_ONLY_FIRST, 252
INSN_MASK_VECTOR_INDEXED, 252
INSN_MASK_VECTOR_LOAD_STORE, 252
INSN_MASK_VECTOR_STRIDE, 252
INSN_MASK_VECTOR_UNIT_STRIDE, 252
INSN_MASK_VECTOR_WHOLE_REG, 252
INSN_MASK_WFI, 252
INSN_MATCH_C_FLD, 252
INSN_MATCH_C_FLDSP, 252
INSN_MATCH_C_FLW, 252
INSN_MATCH_C_FLWSP, 252
INSN_MATCH_C_FSD, 252
INSN_MATCH_C_FSDSP, 252
INSN_MATCH_C_FSW, 252
INSN_MATCH_C_FSWSP, 252
INSN_MATCH_C_LD, 253
INSN_MATCH_C_LDSP, 253
INSN_MATCH_C_LH, 253
INSN_MATCH_C_LHU, 253
INSN_MATCH_C_LW, 253
INSN_MATCH_C_LWSP, 253
INSN_MATCH_C_SD, 253
INSN_MATCH_C_SDSP, 253
INSN_MATCH_C_SH, 253
INSN_MATCH_C_SW, 253
INSN_MATCH_C_SWSP, 253
INSN_MATCH_FENCE_TSO, 253
INSN_MATCH_FLD, 253
INSN_MATCH_FLQ, 253
INSN_MATCH_FLW, 253
INSN_MATCH_FSD, 254
INSN_MATCH_FSQ, 254
INSN_MATCH_FSW, 254
INSN_MATCH_LB, 254
INSN_MATCH_LBU, 254
INSN_MATCH_LD, 254
INSN_MATCH_LH, 254
INSN_MATCH_LHU, 254
INSN_MATCH_LW, 254
INSN_MATCH_LWU, 254
INSN_MATCH_SB, 254
INSN_MATCH_SD, 254
INSN_MATCH_SH, 254
INSN_MATCH_SW, 254

INSN_MATCH_VECTOR_LOAD, 254
INSN_MATCH_VECTOR_STORE, 255
INSN_MATCH_VL1RE16V, 255
INSN_MATCH_VL1RE32V, 255
INSN_MATCH_VL1RE64V, 255
INSN_MATCH_VL1RE8V, 255
INSN_MATCH_VL2RE16V, 255
INSN_MATCH_VL2RE32V, 255
INSN_MATCH_VL2RE64V, 255
INSN_MATCH_VL2RE8V, 255
INSN_MATCH_VL4RE16V, 255
INSN_MATCH_VL4RE32V, 255
INSN_MATCH_VL4RE64V, 255
INSN_MATCH_VL4RE8V, 255
INSN_MATCH_VL8RE16V, 255
INSN_MATCH_VL8RE32V, 255
INSN_MATCH_VL8RE64V, 256
INSN_MATCH_VL8RE8V, 256
INSN_MATCH_VLE16FFV, 256
INSN_MATCH_VLE16V, 256
INSN_MATCH_VLE32FFV, 256
INSN_MATCH_VLE32V, 256
INSN_MATCH_VLE64FFV, 256
INSN_MATCH_VLE64V, 256
INSN_MATCH_VLOXEI16V, 256
INSN_MATCH_VLOXEI32V, 256
INSN_MATCH_VLOXEI64V, 256
INSN_MATCH_VLOXSEG, 256
INSN_MATCH_VLSE16V, 256
INSN_MATCH_VLSE32V, 256
INSN_MATCH_VLSE64V, 257
INSN_MATCH_VLSEG, 257
INSN_MATCH_VLSEGFF, 257
INSN_MATCH_VLSSEG, 257
INSN_MATCH_VLUXEI16V, 257
INSN_MATCH_VLUXEI32V, 257
INSN_MATCH_VLUXEI64V, 257
INSN_MATCH_VLUXSEG, 257
INSN_MATCH_VS1RV, 257
INSN_MATCH_VS2RV, 257
INSN_MATCH_VS4RV, 257
INSN_MATCH_VS8RV, 258
INSN_MATCH_VSE16V, 258
INSN_MATCH_VSE32V, 258
INSN_MATCH_VSE64V, 258
INSN_MATCH_VSOXEI16V, 258
INSN_MATCH_VSOXEI32V, 258
INSN_MATCH_VSOXEI64V, 258
INSN_MATCH_VSOXSEG, 258
INSN_MATCH_VSSE16V, 258
INSN_MATCH_VSSE32V, 258
INSN_MATCH_VSSE64V, 258
INSN_MATCH_VSSEG, 258
INSN_MATCH_VSSSEG, 258
INSN_MATCH_VSUXEI16V, 259
INSN_MATCH_VSUXEI32V, 259
INSN_MATCH_VSUXEI64V, 259
INSN_MATCH_VSUXSEG, 259
INSN_MATCH_WFI, 259
IRQ_M_EXT, 259
IRQ_M_SOFT, 259
IRQ_M_TIMER, 259
IRQ_PMU_OVF, 259
IRQ_S_EXT, 259
IRQ_S_GEXT, 259
IRQ_S_SOFT, 259
IRQ_S_TIMER, 259
IRQ_VS_EXT, 260
IRQ_VS_SOFT, 260
IRQ_VS_TIMER, 260
IS_FAULT_ONLY_FIRST_LOAD, 260
IS_FAULT_ONLY_FIRST_MATCH, 260
IS_INDEXED_LOAD, 260
IS_INDEXED_MATCH, 260
IS_INDEXED_STORE, 260
IS_MASKED, 260
IS_STRIDE_LOAD, 261
IS_STRIDE_MATCH, 261
IS_STRIDE_STORE, 261
IS_UNIT_STRIDE_LOAD, 261
IS_UNIT_STRIDE_MATCH, 262
IS_UNIT_STRIDE_STORE, 262
IS_VECTOR_INSN_MATCH, 262
IS_VECTOR_LOAD_STORE, 262
IS_WHOLE_REG_LOAD, 263
IS_WHOLE_REG_MATCH, 263
IS_WHOLE_REG_STORE, 263
LOG_REGBYTES, 263
MASK_CSR, 263
MASK_FUNCT3, 263
MASK_RS1, 263
MCAUSE_IRQ_MASK, 263
MHPMEVENT_MINH, 263
MHPMEVENT_OF, 263
MHPMEVENT_SINH, 264
MHPMEVENT_SSCOF_MASK, 264
MHPMEVENT_UINH, 264
MHPMEVENT_VSINH, 264
MHPMEVENT_VUINH, 264
MHPMEVENTH_MINH, 264
MHPMEVENTH_OF, 264
MHPMEVENTH_SINH, 264
MHPMEVENTH_UINH, 264
MHPMEVENTH_VSINH, 264
MHPMEVENTH_VUINH, 264
MIP_LCOFIP, 264
MIP_MEIP, 264
MIP_MSIP, 264
MIP_MTIP, 264
MIP_SEIP, 265
MIP_SGEIP, 265
MIP_SSIP, 265
MIP_STIP, 265
MIP_VSEIP, 265
MIP_VSSIP, 265
MIP_VSTIP, 265

MSECCFG_MML, 265
MSECCFG_MML_SHIFT, 265
MSECCFG_MMWP, 265
MSECCFG_MMWP_SHIFT, 265
MSECCFG_RLB, 265
MSECCFG_RLB_SHIFT, 265
MSECCFG_SSEED, 265
MSECCFG_SSEED_SHIFT, 265
MSECCFG_USEED, 266
MSECCFG_USEED_SHIFT, 266
MSTATUS32_SD, 266
MSTATUS64_SD, 266
MSTATUS_FS, 266
MSTATUS_MIE, 266
MSTATUS_MPIE, 266
MSTATUS_MPP, 266
MSTATUS_MPP_SHIFT, 266
MSTATUS_MPRV, 266
MSTATUS_MXR, 266
MSTATUS_SD, 266
MSTATUS_SIE, 266
MSTATUS_SPELP, 266
MSTATUS_SPIE, 267
MSTATUS_SPIE_SHIFT, 267
MSTATUS_SPP, 267
MSTATUS_SPP_SHIFT, 267
MSTATUS_SUM, 267
MSTATUS_TSR, 267
MSTATUS_TVM, 267
MSTATUS_TW, 267
MSTATUS_UBE, 267
MSTATUS_VS, 267
MSTATUS_XS, 267
MSTATUSH_GVA, 267
MSTATUSH_GVA_SHIFT, 267
MSTATUSH_MBE, 267
MSTATUSH_MDT, 267
MSTATUSH_MPELP, 268
MSTATUSH_MPV, 268
MSTATUSH_SBE, 268
PMP_A, 268
PMP_A_NA4, 268
PMP_A_NAPOT, 268
PMP_A_TOR, 268
PMP_ADDR_MASK, 268
PMP_COUNT, 268
PMP_L, 268
PMP_R, 268
PMP_SHIFT, 268
PMP_W, 268
PMP_X, 268
PRV_M, 268
PRV_S, 269
PRV_U, 269
REG_MASK, 269
REG_OFFSET, 269
REG_PTR, 269
REGBYTES, 269
RV_X, 269
RVC_LD_IMM, 269
RVC_LDSP_IMM, 269
RVC_LW_IMM, 269
RVC_LWSP_IMM, 270
RVC_RS1S, 270
RVC_RS2, 270
RVC_RS2S, 270
RVC_SDSP_IMM, 270
RVC_SWSP_IMM, 270
SATP32_ASID, 270
SATP32_MODE, 270
SATP32_PPN, 270
SATP64_ASID, 270
SATP64_MODE, 270
SATP64_PPN, 271
SATP_MODE, 271
SATP_MODE_OFF, 271
SATP_MODE_SV32, 271
SATP_MODE_SV39, 271
SATP_MODE_SV48, 271
SATP_MODE_SV57, 271
SATP_MODE_SV64, 271
SET_RD, 271
SH_MEW, 271
SH_RD, 271
SH_RS1, 271
SH_RS2, 271
SH_RS2C, 271
SH_VD, 272
SH_VIEW, 272
SH_VM, 272
SH_VS2, 272
SH_VSEW, 272
SHIFT_CSR, 272
SHIFT_FUNCT3, 272
SHIFT_RIGHT, 272
SIP_SSIP, 272
SIP_STIP, 272
SMSTATEEN0_AIA, 272
SMSTATEEN0_AIA_SHIFT, 272
SMSTATEEN0_CONTEXT, 272
SMSTATEEN0_CONTEXT_SHIFT, 272
SMSTATEEN0_CS, 272
SMSTATEEN0_CS_SHIFT, 273
SMSTATEEN0_FCSR, 273
SMSTATEEN0_FCSR_SHIFT, 273
SMSTATEEN0_HSENVCFG, 273
SMSTATEEN0_HSENVCFG_SHIFT, 273
SMSTATEEN0_IMSIC, 273
SMSTATEEN0_IMSIC_SHIFT, 273
SMSTATEEN0_SVSLCT, 273
SMSTATEEN0_SVSLCT_SHIFT, 273
SMSTATEEN_MAX_COUNT, 273
SMSTATEEN_STATEN, 273
SMSTATEEN_STATEN_SHIFT, 273
SSTATUS32_SD, 273
SSTATUS64_SD, 273

SSTATUS64_UXL, 273
SSTATUS_FS, 274
SSTATUS_MXR, 274
SSTATUS_SD, 274
SSTATUS_SIE, 274
SSTATUS_SPIE, 274
SSTATUS_SPIE_SHIFT, 274
SSTATUS_SPP, 274
SSTATUS_SPP_SHIFT, 274
SSTATUS_SUM, 274
SSTATUS_VS, 274
SSTATUS_XS, 274
TOPI_IID_MASK, 274
TOPI_IID_SHIFT, 274
TOPI_IPRIO_MASK, 274
VD_MASK, 274
VIEW_MASK, 275
VLMUL_MASK, 275
VM_MASK, 275
VS2_MASK, 275
VSEW_MASK, 275
RISCV_FENCE
riscv_barrier.h, 185
RISCV_FENCE_I
riscv_barrier.h, 186
riscv_fp.h
GET_PRECISION, 292
PRECISION_D, 292
PRECISION_S, 292
riscv_io.h
__io_ar, 295
__io_aw, 295
__io_br, 295
__io_bw, 295
__io_rar, 295
__io_raw, 295
__io_rbr, 295
__io_rbw, 295
__raw_readb, 297
__raw_readl, 297
__raw_readq, 297
__raw_readw, 297
__raw_writeb, 297
__raw_writel, 297
__raw_writeq, 297
__raw_writew, 297
readb, 295
readb_relaxed, 295
readl, 295
readl_relaxed, 295
readq, 296
readq_relaxed, 296
readw, 296
readw_relaxed, 296
writeb, 296
writeb_relaxed, 296
writel, 296
writel_relaxed, 296
writeq, 296
writeq_relaxed, 296
writew, 296
writew_relaxed, 297
RISCV_ISA_EXT_NAME_LEN_MAX
fdt_helper.c, 1156
riscv_locks.c
spin_lock, 791
spin_lock_check, 792
spin_lock_unlocked, 793
spin_trylock, 793
spin_unlock, 794
riscv_locks.h
__SPIN_LOCK_UNLOCKED, 300
DEFINE_SPIN_LOCK, 300
spin_lock, 300
spin_lock_check, 301
SPIN_LOCK_INIT, 300
SPIN_LOCK_INITIALIZER, 300
spin_trylock, 302
spin_unlock, 302
TICKET_SHIFT, 300
riscv_locks_test.c
locks_test_cases, 1118
SBIUNIT_TEST_SUITE, 1116
spin_lock_test, 1116
spin_trylock_fail, 1117
spin_trylock_success, 1117
test_lock, 1118
RISCV_RELEASE_BARRIER
riscv_barrier.h, 186
rmb
riscv_barrier.h, 186
root
sbi_domain.c, 856
sbi_domain.h, 367
root_add_memregion
sbi_domain.c, 843
root_memregs_count
sbi_domain.c, 856
ROOT_REGION_MAX
sbi_domain.c, 839
ROUNDDOWN
sbi_types.h, 620
ROUNDUP
sbi_types.h, 620
rpmi_base_context_priv_level
rpmi_msgprot.h, 722
RPMI_BASE_CONTEXT_PRIV_M_MODE
rpmi_msgprot.h, 722
RPMI_BASE_CONTEXT_PRIV_S_MODE
rpmi_msgprot.h, 722
RPMI_BASE_FLAGS_F0_EV_NOTIFY
rpmi_msgprot.h, 718
RPMI_BASE_FLAGS_F0_MSI_EN
rpmi_msgprot.h, 718
RPMI_BASE_FLAGS_F0_PRIVILEGE
rpmi_msgprot.h, 718

rpmi_base_get_attributes_resp, 100
f0, 101
f1, 101
f2, 101
f3, 101
status_code, 101
rpmi_base_get_platform_info_resp, 101
plat_info, 101
plat_info_len, 101
status, 101
rpmi_base_service_id
rpmi_msgprot.h, 722
RPMI_BASE_SRV_ENABLE_NOTIFICATION
rpmi_msgprot.h, 722
RPMI_BASE_SRV_GET_ATTRIBUTES
rpmi_msgprot.h, 722
RPMI_BASE_SRV_GET_IMPLEMENTATION_IDN
rpmi_msgprot.h, 722
RPMI_BASE_SRV_GET_IMPLEMENTATION_VERSION
rpmi_msgprot.h, 722
RPMI_BASE_SRV_GET_PLATFORM_INFO
rpmi_msgprot.h, 722
RPMI_BASE_SRV_GET_SPEC_VERSION
rpmi_msgprot.h, 722
RPMI_BASE_SRV_PROBE_SERVICE_GROUP
rpmi_msgprot.h, 722
RPMI_BASE_SRV_SET_MSI
rpmi_msgprot.h, 722
RPMI_BASE_VERSION_MIN
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_CLOCK_CONFIG_ENABLE
rpmi_msgprot.h, 718
RPMI_CLOCK_FLAGS_FORMAT_DISCRETE
rpmi_msgprot.h, 718
RPMI_CLOCK_FLAGS_FORMAT_LINEAR
rpmi_msgprot.h, 718
RPMI_CLOCK_FLAGS_FORMAT_MASK
rpmi_msgprot.h, 718
RPMI_CLOCK_FLAGS_FORMAT_POS
rpmi_msgprot.h, 718
rpmi_clock_get_attributes_req, 101
clock_id, 102
rpmi_clock_get_attributes_resp, 102
flags, 102
name, 102
num_rates, 102
status, 102
transition_latency, 102
rpmi_clock_get_config_req, 102
clock_id, 102
rpmi_clock_get_config_resp, 103
config, 103
status, 103
rpmi_clock_get_num_clocks_resp, 103
num_clocks, 103
status, 103
rpmi_clock_get_rate_req, 103
clock_id, 103
rpmi_clock_get_rate_resp, 104
clock_rate_high, 104
clock_rate_low, 104
status, 104
rpmi_clock_get_supported_rates_req, 104
clock_id, 104
clock_rate_index, 104
rpmi_clock_get_supported_rates_resp, 104
clock_rate, 105
flags, 105
remaining, 105
returned, 105
status, 105
rpmi_clock_service_id
rpmi_msgprot.h, 722
rpmi_clock_set_config_req, 105
clock_id, 105
config, 105
rpmi_clock_set_config_resp, 105
status, 106
RPMI_CLOCK_SET_RATE_FLAGS_MASK
rpmi_msgprot.h, 718
RPMI_CLOCK_SET_RATE_FLAGS_ROUND_DOWN
rpmi_msgprot.h, 718
RPMI_CLOCK_SET_RATE_FLAGS_ROUND_PLAT
rpmi_msgprot.h, 718
RPMI_CLOCK_SET_RATE_FLAGS_ROUND_UP
rpmi_msgprot.h, 718
rpmi_clock_set_rate_req, 106
clock_id, 106
clock_rate_high, 106
clock_rate_low, 106
flags, 106
rpmi_clock_set_rate_resp, 106
status, 106
RPMI_CLOCK_SRV_ENABLE_NOTIFICATION
rpmi_msgprot.h, 722
RPMI_CLOCK_SRV_GET_ATTRIBUTES
rpmi_msgprot.h, 722
RPMI_CLOCK_SRV_GET_CONFIG
rpmi_msgprot.h, 722
RPMI_CLOCK_SRV_GET_NUM_CLOCKS
rpmi_msgprot.h, 722
RPMI_CLOCK_SRV_GET_RATE
rpmi_msgprot.h, 722
RPMI_CLOCK_SRV_GET_SUPPORTED_RATES
rpmi_msgprot.h, 722
RPMI_CLOCK_SRV_MAX_COUNT
rpmi_msgprot.h, 722
RPMI_CLOCK_SRV_SET_CONFIG
rpmi_msgprot.h, 722
RPMI_CLOCK_SRV_SET_RATE
rpmi_msgprot.h, 722
rpmi_cppc_cold_init
fdt_cppc_rpmi.c, 1130
rpmi_cppc_fast_channel_cppc_mode
rpmi_msgprot.h, 722
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_ACTIVE

rpmi_msgprot.h, 723
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MASK
rpmi_msgprot.h, 718
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MAX_IDX
rpmi_msgprot.h, 723
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_PASSIVE
rpmi_msgprot.h, 723
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS
rpmi_msgprot.h, 718
rpmi_cppc_fast_channel_db_width
rpmi_msgprot.h, 723
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_16
rpmi_msgprot.h, 723
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_32
rpmi_msgprot.h, 723
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_64
rpmi_msgprot.h, 723
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_8
rpmi_msgprot.h, 723
RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_SUPPORTED
rpmi_msgprot.h, 719
RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_MASK
rpmi_msgprot.h, 719
RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS
rpmi_msgprot.h, 719
rpmi_cppc_fc_db_trigger
fdt_cppc_rpmi.c, 1130
rpmi_cppc_get_fastchan_offset_req, 107
hart_id, 107
rpmi_cppc_get_fastchan_offset_resp, 107
fc_perf_feedback_offset_hi, 107
fc_perf_feedback_offset_lo, 107
fc_perf_request_offset_hi, 107
fc_perf_request_offset_lo, 107
status, 107
rpmi_cppc_get_fastchan_region_resp, 107
db_addr_hi, 108
db_addr_lo, 108
db_preservemask_hi, 108
db_preservemask_lo, 108
db_setmask_hi, 108
db_setmask_lo, 108
flags, 108
region_addr_hi, 108
region_addr_lo, 108
region_size_hi, 108
region_size_lo, 108
status, 108
rpmi_cppc_get_pointer
fdt_cppc_rpmi.c, 1130
RPMI_CPPC_HART_FASTCHAN_SIZE
fdt_cppc_rpmi.c, 1130
rpmi_cppc_hart_list_req, 109
start_index, 109
rpmi_cppc_hart_list_resp, 109
hartid, 109
remaining, 109
returned, 109
status, 109
rpmi_cppc_match
fdt_cppc_rpmi.c, 1132
rpmi_cppc_offset
fdt_cppc_rpmi.c, 1132
rpmi_cppc_probe
fdt_cppc_rpmi.c, 1131
rpmi_cppc_probe_req, 109
hart_id, 110
reg_id, 110
rpmi_cppc_probe_resp, 110
reg_len, 110
status, 110
rpmi_cppc_read
fdt_cppc_rpmi.c, 1131
rpmi_cppc_read_reg_req, 110
hart_id, 110
reg_id, 110
rpmi_cppc_read_reg_resp, 111
data_hi, 111
data_lo, 111
status, 111
rpmi_cppc_service_id
rpmi_msgprot.h, 723
RPMI_CPPC_SRV_ENABLE_NOTIFICATION
rpmi_msgprot.h, 723
RPMI_CPPC_SRV_GET_FAST_CHANNEL_OFFSET
rpmi_msgprot.h, 723
RPMI_CPPC_SRV_GET_FAST_CHANNEL_REGION
rpmi_msgprot.h, 723
RPMI_CPPC_SRV_GET_HART_LIST
rpmi_msgprot.h, 723
RPMI_CPPC_SRV_MAX_COUNT
rpmi_msgprot.h, 723
RPMI_CPPC_SRV_PROBE_REG
rpmi_msgprot.h, 723
RPMI_CPPC_SRV_READ_REG
rpmi_msgprot.h, 723
RPMI_CPPC_SRV_WRITE_REG
rpmi_msgprot.h, 723
rpmi_cppc_update_hart_scratch
fdt_cppc_rpmi.c, 1131
rpmi_cppc_write
fdt_cppc_rpmi.c, 1132
rpmi_cppc_write_reg_req, 111
data_hi, 111
data_lo, 111
hart_id, 111
reg_id, 111
rpmi_cppc_write_reg_resp, 112
status, 112
RPMI_DEF_RX_TIMEOUT
rpmi_msgprot.h, 719
RPMI_DEF_TX_TIMEOUT
rpmi_msgprot.h, 719
rpmi_do_system_reset
fdt_reset_rpmi.c, 1517
rpmi_enable_notification_req, 112

eventid, 112
rpmi_enable_notification_resp, 112
status, 112
RPMI_ERR_ALREADY
rpmi_msgprot.h, 723
RPMI_ERR_BAD_RANGE
rpmi_msgprot.h, 724
RPMI_ERR_BUSY
rpmi_msgprot.h, 724
RPMI_ERR_DENIED
rpmi_msgprot.h, 723
RPMI_ERR_EXTENSION
rpmi_msgprot.h, 723
RPMI_ERR_FAILED
rpmi_msgprot.h, 723
RPMI_ERR_HW_FAULT
rpmi_msgprot.h, 723
RPMI_ERR_INVALID_ADDR
rpmi_msgprot.h, 723
RPMI_ERR_INVALID_PARAM
rpmi_msgprot.h, 723
RPMI_ERR_INVALID_STATE
rpmi_msgprot.h, 724
RPMI_ERR_IO
rpmi_msgprot.h, 724
RPMI_ERR_NO_DATA
rpmi_msgprot.h, 724
RPMI_ERR_NOTSUPP
rpmi_msgprot.h, 723
RPMI_ERR_RESERVED_END
rpmi_msgprot.h, 724
RPMI_ERR_RESERVED_START
rpmi_msgprot.h, 724
RPMI_ERR_TIMEOUT
rpmi_msgprot.h, 724
RPMI_ERR_VENDOR_START
rpmi_msgprot.h, 724
rpmi_error
rpmi_msgprot.h, 723
rpmi_get_platform_info
fdt_mailbox_rpmi_shmem.c, 1468
rpmi_hsm_cold_init
fdt_hsm_rpmi.c, 1193
rpmi_hsm_do_fixup
fdt_hsm_rpmi.c, 1194
rpmi_hsm_fixup
fdt_hsm_rpmi.c, 1198
rpmi_hsm_get_hart_list_req, 113
start_index, 113
rpmi_hsm_get_hart_list_resp, 113
hartid, 113
remaining, 113
returned, 113
status, 113
rpmi_hsm_get_hart_status_req, 113
hartid, 114
rpmi_hsm_get_hart_status_resp, 114
hart_status, 114
status, 114
rpmi_hsm_get_num_suspend_states
fdt_hsm_rpmi.c, 1194
rpmi_hsm_get_pointer
fdt_hsm_rpmi.c, 1195
rpmi_hsm_get_susp_info_req, 114
suspend_type, 114
rpmi_hsm_get_susp_info_resp, 114
entry_latency_us, 115
exit_latency_us, 115
flags, 115
min_residency_us, 115
status, 115
wakeup_latency_us, 115
rpmi_hsm_get_susp_types_req, 115
start_index, 115
rpmi_hsm_get_susp_types_resp, 115
remaining, 116
returned, 116
status, 116
types, 116
rpmi_hsm_get_suspend_states
fdt_hsm_rpmi.c, 1195
rpmi_hsm_hart_start_req, 116
hartid, 116
start_addr_hi, 116
start_addr_lo, 116
rpmi_hsm_hart_start_resp, 116
status, 117
rpmi_hsm_hart_stop_req, 117
hartid, 117
rpmi_hsm_hart_stop_resp, 117
status, 117
rpmi_hsm_hart_susp_req, 117
hartid, 117
resume_addr_hi, 117
resume_addr_lo, 118
suspend_type, 118
rpmi_hsm_hart_susp_resp, 118
status, 118
rpmi_hsm_match
fdt_hsm_rpmi.c, 1198
rpmi_hsm_offset
fdt_hsm_rpmi.c, 1198
rpmi_hsm_service_id
rpmi_msgprot.h, 724
RPMI_HSM_SRV_ENABLE_NOTIFICATION
rpmi_msgprot.h, 724
RPMI_HSM_SRV_GET_HART_LIST
rpmi_msgprot.h, 724
RPMI_HSM_SRV_GET_HART_STATUS
rpmi_msgprot.h, 724
RPMI_HSM_SRV_GET_SUSPEND_INFO
rpmi_msgprot.h, 724
RPMI_HSM_SRV_GET_SUSPEND_TYPES
rpmi_msgprot.h, 724
RPMI_HSM_SRV_HART_START
rpmi_msgprot.h, 724

RPMI_HSM_SRV_HART_STOP
rpmi_msgprot.h, 724
RPMI_HSM_SRV_HART_SUSPEND
rpmi_msgprot.h, 724
RPMI_HSM_SRV_ID_MAX
rpmi_msgprot.h, 724
rpmi_hsm_start
fdt_hsm_rpmi.c, 1196
rpmi_hsm_stop
fdt_hsm_rpmi.c, 1196
rpmi_hsm_suspend
fdt_hsm_rpmi.c, 1196
RPMI_HSM_SUSPEND_INFO_FLAGS_TIMER_STOP
rpmi_msgprot.h, 719
rpmi_hsm_update_hart_scratch
fdt_hsm_rpmi.c, 1197
rpmi_mailbox.c
rpmi_normal_request_with_status, 1480
rpmi_posted_request, 1481
rpmi_xlate_error, 1482
rpmi_mailbox.h
rpmi_normal_request_with_status, 711
rpmi_posted_request, 712
rpmi_u32_count, 711
rpmi_xlate_error, 713
RPMI_MAJOR_VER
fdt_mpxy_rpmi_mbox.c, 1485
rpmi_message, 118
data, 118
header, 118
rpmi_message_args, 119
flags, 119
rx_data_len, 119
rx_endian_words, 119
rx_token, 119
service_id, 119
tx_endian_words, 119
type, 119
rpmi_message_header, 119
datalen, 120
flags, 120
service_id, 120
servicegroup_id, 120
token, 120
rpmi_message_type
rpmi_msgprot.h, 724
RPMI_MINOR_VER
fdt_mpxy_rpmi_mbox.c, 1485
RPMI_MSG_ACKNOWLDGEMENT
rpmi_msgprot.h, 724
RPMI_MSG_DATA_OFFSET
rpmi_msgprot.h, 719
RPMI_MSG_DATA_SIZE
rpmi_msgprot.h, 719
RPMI_MSG_DATALEN_OFFSET
rpmi_msgprot.h, 719
RPMI_MSG_DATALEN_SIZE
rpmi_msgprot.h, 719
RPMI_MSG_FLAGS_DOORBELL
rpmi_msgprot.h, 719
RPMI_MSG_FLAGS_DOORBELL_MASK
rpmi_msgprot.h, 719
RPMI_MSG_FLAGS_DOORBELL_POS
rpmi_msgprot.h, 719
RPMI_MSG_FLAGS_NO_RX
rpmi_msgprot.h, 719
RPMI_MSG_FLAGS_NO_RX_TOKEN
rpmi_msgprot.h, 720
RPMI_MSG_FLAGS_NO_TX
rpmi_msgprot.h, 720
RPMI_MSG_FLAGS_OFFSET
rpmi_msgprot.h, 720
RPMI_MSG_FLAGS_SIZE
rpmi_msgprot.h, 720
RPMI_MSG_FLAGS_TYPE
rpmi_msgprot.h, 720
RPMI_MSG_FLAGS_TYPE_MASK
rpmi_msgprot.h, 720
RPMI_MSG_FLAGS_TYPE_POS
rpmi_msgprot.h, 720
RPMI_MSG_HDR_OFFSET
rpmi_msgprot.h, 720
RPMI_MSG_HDR_SIZE
rpmi_msgprot.h, 720
RPMI_MSG_NORMAL_REQUEST
rpmi_msgprot.h, 724
RPMI_MSG_NOTIFICATION
rpmi_msgprot.h, 724
RPMI_MSG_POSTED_REQUEST
rpmi_msgprot.h, 724
RPMI_MSG_SERVICE_ID_OFFSET
rpmi_msgprot.h, 720
RPMI_MSG_SERVICE_ID_SIZE
rpmi_msgprot.h, 720
RPMI_MSG_SERVICEGROUP_ID_OFFSET
rpmi_msgprot.h, 720
RPMI_MSG_SERVICEGROUP_ID_SIZE
rpmi_msgprot.h, 720
RPMI_MSG_TOKEN_MASK
rpmi_msgprot.h, 721
RPMI_MSG_TOKEN_OFFSET
rpmi_msgprot.h, 721
RPMI_MSG_TOKEN_SIZE
rpmi_msgprot.h, 721
rpmi_msgprot.h
__packed, 725
rpmi_base_context_priv_level, 722
RPMI_BASE_CONTEXT_PRIV_M_MODE, 722
RPMI_BASE_CONTEXT_PRIV_S_MODE, 722
RPMI_BASE_FLAGS_F0_EV_NOTIFY, 718
RPMI_BASE_FLAGS_F0_MSI_EN, 718
RPMI_BASE_FLAGS_F0_PRIVILEGE, 718
rpmi_base_service_id, 722
RPMI_BASE_SRV_ENABLE_NOTIFICATION, 722
RPMI_BASE_SRV_GET_ATTRIBUTES, 722

RPMI_BASE_SRV_GET_IMPLEMENTATION_IDN, 722
RPMI_BASE_SRV_GET_IMPLEMENTATION_VERSION, 722
RPMI_BASE_SRV_GET_PLATFORM_INFO, 722
RPMI_BASE_SRV_GET_SPEC_VERSION, 722
RPMI_BASE_SRV_PROBE_SERVICE_GROUP, 722
RPMI_BASE_SRV_SET_MSI, 722
RPMI_CLOCK_CONFIG_ENABLE, 718
RPMI_CLOCK_FLAGS_FORMAT_DISCRETE, 718
RPMI_CLOCK_FLAGS_FORMAT_LINEAR, 718
RPMI_CLOCK_FLAGS_FORMAT_MASK, 718
RPMI_CLOCK_FLAGS_FORMAT_POS, 718
rpmi_clock_service_id, 722
RPMI_CLOCK_SET_RATE_FLAGS_MASK, 718
RPMI_CLOCK_SET_RATE_FLAGS_ROUND_DOWN, 718
RPMI_CLOCK_SET_RATE_FLAGS_ROUND_PLAT, 718
RPMI_CLOCK_SET_RATE_FLAGS_ROUND_UP, 718
RPMI_CLOCK_SRV_ENABLE_NOTIFICATION, 722
RPMI_CLOCK_SRV_GET_ATTRIBUTES, 722
RPMI_CLOCK_SRV_GET_CONFIG, 722
RPMI_CLOCK_SRV_GET_NUM_CLOCKS, 722
RPMI_CLOCK_SRV_GET_RATE, 722
RPMI_CLOCK_SRV_GET_SUPPORTED_RATES, 722
RPMI_CLOCK_SRV_MAX_COUNT, 722
RPMI_CLOCK_SRV_SET_CONFIG, 722
RPMI_CLOCK_SRV_SET_RATE, 722
rpmi_cppc_fast_channel_cppc_mode, 722
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_ACTIVE, 723
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MASK, 718
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_MAX_IDX, 723
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_PASSIVE, 723
RPMI_CPPC_FAST_CHANNEL_CPPC_MODE_POS, 718
rpmi_cppc_fast_channel_db_width, 723
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_16, 723
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_32, 723
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_64, 723
RPMI_CPPC_FAST_CHANNEL_DB_WIDTH_8, 723
RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_SUPPORTED, 719
RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_MASK, 719
RPMI_CPPC_FAST_CHANNEL_FLAGS_DB_WIDTH_POS, 719
rpmi_cppc_service_id, 723
RPMI_CPPC_SRV_ENABLE_NOTIFICATION, 723
RPMI_CPPC_SRV_GET_FAST_CHANNEL_OFFSET, 723
RPMI_CPPC_SRV_GET_FAST_CHANNEL_REGION, 723
RPMI_CPPC_SRV_GET_HART_LIST, 723
RPMI_CPPC_SRV_MAX_COUNT, 723
RPMI_CPPC_SRV_PROBE_REG, 723
RPMI_CPPC_SRV_READ_REG, 723
RPMI_CPPC_SRV_WRITE_REG, 723
RPMI_DEF_RX_TIMEOUT, 719
RPMI_DEF_TX_TIMEOUT, 719
RPMI_ERR_ALREADY, 723
RPMI_ERR_BAD_RANGE, 724
RPMI_ERR_BUSY, 724
RPMI_ERR_DENIED, 723
RPMI_ERR_EXTENSION, 723
RPMI_ERR_FAILED, 723
RPMI_ERR_HW_FAULT, 723
RPMI_ERR_INVALID_ADDR, 723
RPMI_ERR_INVALID_PARAM, 723
RPMI_ERR_INVALID_STATE, 724
RPMI_ERR_IO, 724
RPMI_ERR_NO_DATA, 724
RPMI_ERR_NOTSUPP, 723
RPMI_ERR_RESERVED_END, 724
RPMI_ERR_RESERVED_START, 724
RPMI_ERR_TIMEOUT, 724
RPMI_ERR_VENDOR_START, 724
rpmi_error, 723
rpmi_hsm_service_id, 724
RPMI_HSM_SRV_ENABLE_NOTIFICATION, 724
RPMI_HSM_SRV_GET_HART_LIST, 724
RPMI_HSM_SRV_GET_HART_STATUS, 724
RPMI_HSM_SRV_GET_SUSPEND_INFO, 724
RPMI_HSM_SRV_GET_SUSPEND_TYPES, 724
RPMI_HSM_SRV_HART_START, 724
RPMI_HSM_SRV_HART_STOP, 724
RPMI_HSM_SRV_HART_SUSPEND, 724
RPMI_HSM_SRV_ID_MAX, 724
RPMI_HSM_SUSPEND_INFO_FLAGS_TIMER_STOP, 719
rpmi_message_type, 724
RPMI_MSG_ACKNOWLDGEMENT, 724
RPMI_MSG_DATA_OFFSET, 719
RPMI_MSG_DATA_SIZE, 719
RPMI_MSG_DATALEN_OFFSET, 719
RPMI_MSG_DATALEN_SIZE, 719
RPMI_MSG_FLAGS_DOORBELL, 719
RPMI_MSG_FLAGS_DOORBELL_MASK, 719
RPMI_MSG_FLAGS_DOORBELL_POS, 719
RPMI_MSG_FLAGS_NO_RX, 719
RPMI_MSG_FLAGS_NO_RX_TOKEN, 720
RPMI_MSG_FLAGS_NO_TX, 720
RPMI_MSG_FLAGS_OFFSET, 720

RPMI_MSG_FLAGS_SIZE, 720
RPMI_MSG_FLAGS_TYPE, 720
RPMI_MSG_FLAGS_TYPE_MASK, 720
RPMI_MSG_FLAGS_TYPE_POS, 720
RPMI_MSG_HDR_OFFSET, 720
RPMI_MSG_HDR_SIZE, 720
RPMI_MSG_NORMAL_REQUEST, 724
RPMI_MSG_NOTIFICATION, 724
RPMI_MSG_POSTED_REQUEST, 724
RPMI_MSG_SERVICE_ID_OFFSET, 720
RPMI_MSG_SERVICE_ID_SIZE, 720
RPMI_MSG_SERVICEGROUP_ID_OFFSET, 720
RPMI_MSG_SERVICEGROUP_ID_SIZE, 720
RPMI_MSG_TOKEN_MASK, 721
RPMI_MSG_TOKEN_OFFSET, 721
RPMI_MSG_TOKEN_SIZE, 721
RPMI_NAME_CHARS_MAX, 721
RPMI_QUEUE_HEAD_SLOT, 721
RPMI_QUEUE_HEADER_SLOTS, 721
RPMI_QUEUE_TAIL_SLOT, 721
rpmi_servicegroup_id, 724
RPMI_SLOT_SIZE_MIN, 721
RPMI_SRVGRP_BASE, 724
RPMI_SRVGRP_CLOCK, 725
RPMI_SRVGRP_CPPC, 725
RPMI_SRVGRP_HSM, 725
RPMI_SRVGRP_ID_MAX_COUNT, 725
RPMI_SRVGRP_ID_MIN, 724
RPMI_SRVGRP_RESERVE_END, 725
RPMI_SRVGRP_RESERVE_START, 725
RPMI_SRVGRP_SYSTEM_RESET, 724
RPMI_SRVGRP_SYSTEM_SUSPEND, 724
RPMI_SRVGRP_VENDOR_END, 725
RPMI_SRVGRP_VENDOR_START, 725
RPMI_SUCCESS, 723
RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_MASK, 721
RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS, 721
rpmi_sysrst_reset_type, 725
RPMI_SYSRST_SRV_ENABLE_NOTIFICATION, 725
RPMI_SYSRST_SRV_GET_ATTRIBUTES, 725
RPMI_SYSRST_SRV_ID_MAX_COUNT, 725
RPMI_SYSRST_SRV_SYSTEM_RESET, 725
RPMI_SYSRST_TYPE_COLD_REBOOT, 725
RPMI_SYSRST_TYPE_MAX, 725
RPMI_SYSRST_TYPE_SHUTDOWN, 725
RPMI_SYSRST_TYPE_WARM_REBOOT, 725
RPMI_SYSSUSP_ATTRS_FLAGS_RESUMEADDR, 721
RPMI_SYSSUSP_ATTRS_FLAGS_SUSPENDTYPE, 721
RPMI_SYSSUSP_SRV_ENABLE_NOTIFICATION, 725
RPMI_SYSSUSP_SRV_GET_ATTRIBUTES, 725
RPMI_SYSSUSP_SRV_ID_MAX_COUNT, 725
RPMI_SYSSUSP_SRV_SYSTEM_SUSPEND, 725
rpmi_system_reset_service_id, 725
rpmi_system_suspend_service_id, 725
RPMI_VERSION, 721
RPMI_NAME_CHARS_MAX
rpmi_msgprot.h, 721
rpmi_normal_request_with_status
rpmi_mailbox.c, 1480
rpmi_mailbox.h, 711
rpmi_posted_request
rpmi_mailbox.c, 1481
rpmi_mailbox.h, 712
RPMI_QUEUE_HEAD_SLOT
rpmi_msgprot.h, 721
RPMI_QUEUE_HEADER_SLOTS
rpmi_msgprot.h, 721
rpmi_queue_idx
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_QUEUE_IDX_A2P_ACK
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_QUEUE_IDX_A2P_REQ
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_QUEUE_IDX_MAX_COUNT
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_QUEUE_IDX_P2A_ACK
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_QUEUE_IDX_P2A_REQ
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_QUEUE_TAIL_SLOT
rpmi_msgprot.h, 721
rpmi_queue_type
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_QUEUE_TYPE_ACK
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_QUEUE_TYPE_REQ
fdt_mailbox_rpmi_shmem.c, 1466
rpmi_reg_idx
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_REG_IDX_DB_REG
fdt_mailbox_rpmi_shmem.c, 1466
RPMI_REG_IDX_MAX_COUNT
fdt_mailbox_rpmi_shmem.c, 1466
rpmi_reset_dev
fdt_reset_rpmi.c, 1519
rpmi_reset_init
fdt_reset_rpmi.c, 1517
rpmi_reset_match
fdt_reset_rpmi.c, 1519
rpmi_servicegroup_id
rpmi_msgprot.h, 724
rpmi_shmem_mbox_free_chan
fdt_mailbox_rpmi_shmem.c, 1468
rpmi_shmem_mbox_init
fdt_mailbox_rpmi_shmem.c, 1469
rpmi_shmem_mbox_match
fdt_mailbox_rpmi_shmem.c, 1474
rpmi_shmem_mbox_request_chan

fdt_mailbox_rpmi_shmem.c, 1470
rpmi_shmem_mbox_xfer
fdt_mailbox_rpmi_shmem.c, 1471
rpmi_shmem_transport_init
fdt_mailbox_rpmi_shmem.c, 1471
RPMI_SLOT_SIZE_MIN
rpmi_msgprot.h, 721
RPMI_SRVGRP_BASE
rpmi_msgprot.h, 724
RPMI_SRVGRP_CLOCK
rpmi_msgprot.h, 725
RPMI_SRVGRP_CPPC
rpmi_msgprot.h, 725
RPMI_SRVGRP_HSM
rpmi_msgprot.h, 725
RPMI_SRVGRP_ID_MAX_COUNT
rpmi_msgprot.h, 725
RPMI_SRVGRP_ID_MIN
rpmi_msgprot.h, 724
RPMI_SRVGRP_RESERVE_END
rpmi_msgprot.h, 725
RPMI_SRVGRP_RESERVE_START
rpmi_msgprot.h, 725
RPMI_SRVGRP_SYSTEM_RESET
rpmi_msgprot.h, 724
RPMI_SRVGRP_SYSTEM_SUSPEND
rpmi_msgprot.h, 724
RPMI_SRVGRP_VENDOR_END
rpmi_msgprot.h, 725
RPMI_SRVGRP_VENDOR_START
rpmi_msgprot.h, 725
RPMI_SUCCESS
rpmi_msgprot.h, 723
rpmi_suspend_dev
fdt_suspend_rpmi.c, 1586
rpmi_suspend_init
fdt_suspend_rpmi.c, 1584
rpmi_suspend_match
fdt_suspend_rpmi.c, 1586
RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_MASK
rpmi_msgprot.h, 721
RPMI_SYSRST_ATTRS_FLAGS_RESETTYPE_POS
rpmi_msgprot.h, 721
rpmi_sysrst_get_reset_attributes_resp, 120
flags, 120
status, 120
rpmi_sysrst_reset_type
rpmi_msgprot.h, 725
RPMI_SYSRST_SRV_ENABLE_NOTIFICATION
rpmi_msgprot.h, 725
RPMI_SYSRST_SRV_GET_ATTRIBUTES
rpmi_msgprot.h, 725
RPMI_SYSRST_SRV_ID_MAX_COUNT
rpmi_msgprot.h, 725
RPMI_SYSRST_SRV_SYSTEM_RESET
rpmi_msgprot.h, 725
RPMI_SYSRST_TYPE_COLD_REBOOT
rpmi_msgprot.h, 725
RPMI_SYSRST_TYPE_MAX
rpmi_msgprot.h, 725
RPMI_SYSRST_TYPE_SHUTDOWN
rpmi_msgprot.h, 725
RPMI_SYSRST_TYPE_WARM_REBOOT
rpmi_msgprot.h, 725
rpmi_syssusp
fdt_suspend_rpmi.c, 1585
rpmi_syssusp_attrs
fdt_suspend_rpmi.c, 1585
RPMI_SYSSUSP_ATTRS_FLAGS_RESUMEADDR
rpmi_msgprot.h, 721
RPMI_SYSSUSP_ATTRS_FLAGS_SUSPENDTYPE
rpmi_msgprot.h, 721
rpmi_syssusp_get_attr_req, 121
susp_type, 121
rpmi_syssusp_get_attr_resp, 121
flags, 121
status, 121
RPMI_SYSSUSP_SRV_ENABLE_NOTIFICATION
rpmi_msgprot.h, 725
RPMI_SYSSUSP_SRV_GET_ATTRIBUTES
rpmi_msgprot.h, 725
RPMI_SYSSUSP_SRV_ID_MAX_COUNT
rpmi_msgprot.h, 725
RPMI_SYSSUSP_SRV_SYSTEM_SUSPEND
rpmi_msgprot.h, 725
rpmi_syssusp_suspend_req, 121
hartid, 122
resume_addr_hi, 122
resume_addr_lo, 122
suspend_type, 122
rpmi_syssusp_suspend_resp, 122
status, 122
rpmi_system_reset
fdt_reset_rpmi.c, 1518
rpmi_system_reset_check
fdt_reset_rpmi.c, 1518
rpmi_system_reset_service_id
rpmi_msgprot.h, 725
rpmi_system_reset_type_check
fdt_reset_rpmi.c, 1518
rpmi_system_suspend
fdt_suspend_rpmi.c, 1586
rpmi_system_suspend_check
fdt_suspend_rpmi.c, 1586
rpmi_system_suspend_service_id
rpmi_msgprot.h, 725
rpmi_u32_count
rpmi_mailbox.h, 711
RPMI_VERSION
rpmi_msgprot.h, 721
rpmi_xlate_error
rpmi_mailbox.c, 1482
rpmi_mailbox.h, 713
RST_CLK_1024
fdt_reset_atcwdt200.c, 1508
RST_CLK_128

fdt_reset_atcwdt200.c, 1508
RST_CLK_256
fdt_reset_atcwdt200.c, 1508
RST_CLK_512
fdt_reset_atcwdt200.c, 1508
RST_EN
fdt_reset_atcwdt200.c, 1508
RST_TIME_MSK
fdt_reset_atcwdt200.c, 1508
RST_TIME_OFF
fdt_reset_atcwdt200.c, 1508
run_all_tests
sbi_unit_test.c, 1127
sbi_unit_test.h, 624
run_test_suite
sbi_unit_test.c, 1127
RV_DBTR_BIT
riscv_dbtr.h, 189
RV_DBTR_BIT_MASK
riscv_dbtr.h, 189
RV_DBTR_DECLARE_BIT
riscv_dbtr.h, 189, 192, 193
sbi_dbtr.h, 344
RV_DBTR_DECLARE_BIT_MASK
riscv_dbtr.h, 189, 192–194
sbi_dbtr.h, 344
RV_DBTR_SET_MC6_EXEC
riscv_dbtr.h, 189
RV_DBTR_SET_MC6_LOAD
riscv_dbtr.h, 189
RV_DBTR_SET_MC6_SIZE
riscv_dbtr.h, 189
RV_DBTR_SET_MC6_STORE
riscv_dbtr.h, 190
RV_DBTR_SET_MC6_TYPE
riscv_dbtr.h, 190
RV_DBTR_SET_MC_EXEC
riscv_dbtr.h, 190
RV_DBTR_SET_MC_LOAD
riscv_dbtr.h, 190
RV_DBTR_SET_MC_SIZEHI
riscv_dbtr.h, 190
RV_DBTR_SET_MC_SIZELO
riscv_dbtr.h, 190
RV_DBTR_SET_MC_STORE
riscv_dbtr.h, 190
RV_DBTR_SET_MC_TYPE
riscv_dbtr.h, 191
RV_DBTR_SET_TDATA1_TYPE
riscv_dbtr.h, 191
RV_MAX_TRIGGERS
riscv_dbtr.h, 191
RV_X
riscv_encoding.h, 269
RVC_LD_IMM
riscv_encoding.h, 269
RVC_LDSP_IMM
riscv_encoding.h, 269
RVC_LW_IMM
riscv_encoding.h, 269
RVC_LWSP_IMM
riscv_encoding.h, 270
RVC_RS1S
riscv_encoding.h, 270
RVC_RS2
riscv_encoding.h, 270
RVC_RS2S
riscv_encoding.h, 270
RVC_SDSP_IMM
riscv_encoding.h, 270
RVC_SWSP_IMM
riscv_encoding.h, 270
rx
mbox_xfer, 96
rx_data_len
rpmi_message_args, 119
rx_endian_words
rpmi_message_args, 119
rx_len
mbox_xfer, 96
rx_timeout
mbox_xfer, 96
rx_token
rpmi_message_args, 119

s0
sbi_trap_regs, 157
s1
sbi_trap_regs, 157
s10
sbi_trap_regs, 157
s11
sbi_trap_regs, 157
s16
sbi_types.h, 621
s2
sbi_trap_regs, 157
s3
sbi_trap_regs, 157
s32
sbi_types.h, 621
s4
sbi_trap_regs, 157
s5
sbi_trap_regs, 157
s6
sbi_trap_regs, 157
s7
sbi_trap_regs, 157
s8
sbi_trap_regs, 157
sbi_types.h, 621
s9
sbi_trap_regs, 157
sanitize_domain
sbi_domain.c, 844
SATP32_ASID

riscv_encoding.h, 270
SATP32_MODE
riscv_encoding.h, 270
SATP32_PPN
riscv_encoding.h, 270
SATP64_ASID
riscv_encoding.h, 270
SATP64_MODE
riscv_encoding.h, 270
SATP64_PPN
riscv_encoding.h, 271
SATP_MODE
riscv_encoding.h, 271
SATP_MODE_OFF
riscv_encoding.h, 271
SATP_MODE_SV32
riscv_encoding.h, 271
SATP_MODE_SV39
riscv_encoding.h, 271
SATP_MODE_SV48
riscv_encoding.h, 271
SATP_MODE_SV57
riscv_encoding.h, 271
SATP_MODE_SV64
riscv_encoding.h, 271
sbi_aligned_alloc
sbi_heap.h, 453
sbi_aligned_alloc_from
sbi_heap.c, 947
sbi_heap.h, 453
sbi_alloc_trigger
sbi_dbtr.c, 830
SBI_ASSERT
sbi_console.h, 327
sbi_bitmap.c
__bitmap_and, 796
__bitmap_or, 796
__bitmap_xor, 797
sbi_bitmap.h
__bitmap_and, 306
__bitmap_or, 306
__bitmap_xor, 306
bitmap_and, 306
bitmap_clear, 307
bitmap_copy, 307
bitmap_estimate_size, 308
bitmap_fill, 308
BITMAP_FIRST_WORD_MASK, 305
BITMAP_LAST_WORD_MASK, 305
bitmap_or, 308
bitmap_set, 309
bitmap_test, 309
bitmap_xor, 309
bitmap_zero, 310
bitmap_zero_except, 310
DECLARE_BITMAP, 305
DEFINE_BITMAP, 305
small_const_nbits, 305
sbi_bitmap_test.c
bitmap_and_test, 1119
bitmap_or_test, 1119
bitmap_test_cases, 1120
bitmap_xor_test, 1119
data_a, 1120
data_b, 1120
DATA_BIT_SIZE, 1119
DATA_SIZE, 1119
data_zero, 1120
SBIUNIT_TEST_SUITE, 1120
sbi_bitops.c
BITOP_WORD, 798
find_first_bit, 798
find_first_zero_bit, 798
find_last_bit, 798
find_next_bit, 799
find_next_zero_bit, 799
sbi_bitops.h
__change_bit, 315
__clear_bit, 315
__set_bit, 316
__test_and_clear_bit, 316
__test_and_set_bit, 316
__test_bit, 316
BIT, 313
BIT_ALIGN, 313
BIT_MASK, 313
BIT_ULL, 314
BIT_WORD, 314
BIT_WORD_OFFSET, 314
BITS_PER_LONG, 314
BITS_PER_LONG_LONG, 314
BITS_TO_LONGS, 314
EXTRACT_FIELD, 314
find_first_bit, 317
find_first_zero_bit, 318
find_last_bit, 318
find_next_bit, 318
find_next_zero_bit, 319
for_each_clear_bit, 314
for_each_clear_bit_from, 314
for_each_set_bit, 314
for_each_set_bit_from, 315
GENMASK, 315
GENMASK_ULL, 315
INSERT_FIELD, 315
sbi_ffs, 319
sbi_ffz, 315
sbi_fls, 319
sbi_popcount, 320
sbi_boot_print_banner
sbi_init.c, 976
sbi_boot_print_domains
sbi_init.c, 976
sbi_boot_print_general
sbi_init.c, 977
sbi_boot_print_hart

sbi_init.c, 977
sbi_byteorder.h
_conv_cast, 323
be16_to_cpu, 323
be32_to_cpu, 323
be64_to_cpu, 324
BSWAP16, 324
BSWAP32, 324
BSWAP64, 324
cpu_to_be16, 324
cpu_to_be32, 324
cpu_to_be64, 324
cpu_to_le16, 324
cpu_to_le32, 324
cpu_to_le64, 325
le16_to_cpu, 325
le32_to_cpu, 325
le64_to_cpu, 325
sbi_calloc
sbi_heap.h, 454
sbi_calloc_from
sbi_heap.h, 455
sbi_console.c
console_dev, 821
console_early_buffer, 821
CONSOLE_EARLY_BUFFER_SIZE, 801
console_out_lock, 821
console_tbuf, 821
console_tbuf_len, 821
CONSOLE_TBUF_MAX, 801
nputs, 802
nputs_all, 803
PAD_ALTERNATE, 801
PAD_RIGHT, 801
PAD_SIGN, 801
PAD_ZERO, 801
print, 804
PRINT_BUF_LEN, 801
printc, 805
printi, 807
prints, 809
sbi_console_get_device, 811
sbi_console_set_device, 812
sbi_dprintf, 813
SBI_FIFO_DEFINE, 814
sbi_getc, 814
sbi_gets, 814
sbi_isprintable, 815
sbi_ngets, 815
sbi_nputs, 815
sbi_panic, 816
sbi_printf, 817
sbi_putc, 818
sbi_puts, 819
sbi_snprintf, 820
sbi_sprintf, 820
USE_TBUF, 801
va_arg, 801
va_end, 801
va_list, 802
va_start, 801
sbi_console.h
__printf, 327
format, 333
out_sz, 333
SBI_ASSERT, 327
sbi_console_get_device, 327
sbi_console_set_device, 328
sbi_getc, 329
sbi_gets, 329
sbi_isprintable, 330
sbi_ngets, 330
sbi_nputs, 330
sbi_putc, 331
sbi_puts, 332
sbi_console_device, 122
console_getc, 123
console_putc, 123
console_puts, 123
name, 123
sbi_console_get_device
sbi_console.c, 811
sbi_console.h, 327
sbi_console_set_device
sbi_console.c, 812
sbi_console.h, 328
sbi_console_test.c
clear_test_console_buf, 1122
console_test_cases, 1124
old_dev, 1124
PRINTF_TEST, 1121
printf_test, 1122
putc_test, 1122
PUTS_TEST, 1122
puts_test, 1123
SBIUNIT_TEST_SUITE, 1123
test_console_begin, 1123
test_console_buf, 1124
TEST_CONSOLE_BUF_LEN, 1122
test_console_buf_pos, 1124
test_console_dev, 1124
test_console_end, 1123
test_console_lock, 1124
test_console_putc, 1124
sbi_const.h
_AC, 334
_AT, 334
_BITUL, 334
_BITULL, 334
_UL, 334
_ULL, 335
__AC, 334
__STR, 334
STRINGIFY, 335
UL, 335
ULL, 335

sbi_cppc.c
cppc_dev, 825
sbi_cppc_get_device, 822
sbi_cppc_is_reserved, 822
sbi_cppc_probe, 823
sbi_cppc_read, 823
sbi_cppc_readable, 824
sbi_cppc_set_device, 824
sbi_cppc_writable, 824
sbi_cppc_write, 825
sbi_cppc.h
sbi_cppc_get_device, 336
sbi_cppc_probe, 337
sbi_cppc_read, 337
sbi_cppc_set_device, 338
sbi_cppc_write, 338
SBI_CPPC_ACPI_LAST
sbi_ecall_interface.h, 404
SBI_CPPC_AUTO_ACT_WINDOW
sbi_ecall_interface.h, 403
SBI_CPPC_AUTO_SEL_ENABLE
sbi_ecall_interface.h, 403
SBI_CPPC_CTR_WRAP_TIME
sbi_ecall_interface.h, 403
SBI_CPPC_DELIVERED_CTR
sbi_ecall_interface.h, 403
SBI_CPPC_DESIRED_PERF
sbi_ecall_interface.h, 403
sbi_cppc_device, 123
cppc_probe, 123
cppc_read, 123
cppc_write, 123
name, 123
SBI_CPPC_ENABLE
sbi_ecall_interface.h, 403
SBI_CPPC_ENERGY_PERF_PREFERENCE
sbi_ecall_interface.h, 403
sbi_cppc_get_device
sbi_cppc.c, 822
sbi_cppc.h, 336
SBI_CPPC_GUARANTEED_PERF
sbi_ecall_interface.h, 403
SBI_CPPC_HIGHEST_PERF
sbi_ecall_interface.h, 403
sbi_cppc_is_reserved
sbi_cppc.c, 822
SBI_CPPC_LOW_NON_LINEAR_PERF
sbi_ecall_interface.h, 403
SBI_CPPC_LOWEST_FREQ
sbi_ecall_interface.h, 403
SBI_CPPC_LOWEST_PERF
sbi_ecall_interface.h, 403
SBI_CPPC_MAX_PERF
sbi_ecall_interface.h, 403
SBI_CPPC_MIN_PERF
sbi_ecall_interface.h, 403
SBI_CPPC_NOMINAL_FREQ
sbi_ecall_interface.h, 403
SBI_CPPC_NOMINAL_PERF
sbi_ecall_interface.h, 403
SBI_CPPC_NON_ACPI_LAST
sbi_ecall_interface.h, 404
SBI_CPPC_PERF_LIMITED
sbi_ecall_interface.h, 403
SBI_CPPC_PERF_REDUC_TOLERANCE
sbi_ecall_interface.h, 403
sbi_cppc_probe
sbi_cppc.c, 823
sbi_cppc.h, 337
sbi_cppc_read
sbi_cppc.c, 823
sbi_cppc.h, 337
sbi_cppc_readable
sbi_cppc.c, 824
SBI_CPPC_REFERENCE_CTR
sbi_ecall_interface.h, 403
SBI_CPPC_REFERENCE_PERF
sbi_ecall_interface.h, 403
sbi_cppc_reg_id
sbi_ecall_interface.h, 403
sbi_cppc_set_device
sbi_cppc.c, 824
sbi_cppc.h, 338
SBI_CPPC_TIME_WINDOW
sbi_ecall_interface.h, 403
SBI_CPPC_TRANSITION_LATENCY
sbi_ecall_interface.h, 404
sbi_cppc_writable
sbi_cppc.c, 824
sbi_cppc_write
sbi_cppc.c, 825
sbi_cppc.h, 338
sbi_cpu_idle_state, 124
entry_latency_us, 124
exit_latency_us, 124
local_timer_stop, 124
min_residency_us, 124
name, 124
suspend_param, 124
wakeup_latency_us, 124
sbi_csr_detect.h
csr_read_allowed, 340
csr_write_allowed, 341
sbi_dbtr.c
dbtr_get_hart_state_ptr, 827
dbtr_set_hart_state_ptr, 827
DBTR_SHMEM_MAKE_PHYS, 827
dbtr_thishart_state_ptr, 827
dbtr_trigger_clear, 827
dbtr_trigger_disable, 828
dbtr_trigger_enable, 828
dbtr_trigger_setup, 829
dbtr_trigger_supported, 829
dbtr_trigger_valid, 829
for_each_trig_entry, 827
hart_shmem_base, 830

hart_state_ptr_offset, 838
INDEX_TO_TRIGGER, 827
sbi_alloc_trigger, 830
sbi_dbtr_disable_shmem, 830
sbi_dbtr_disable_trig, 831
sbi_dbtr_enable_trig, 831
sbi_dbtr_get_total_triggers, 832
sbi_dbtr_init, 832
sbi_dbtr_install_trig, 833
sbi_dbtr_num_trig, 833
sbi_dbtr_read_trig, 834
sbi_dbtr_setup_shmem, 834
sbi_dbtr_shmem_disabled, 835
sbi_dbtr_uninstall_trig, 835
sbi_dbtr_update_trig, 836
sbi_free_trigger, 836
sbi_trigger_init, 837
update_bit, 837
sbi_dbtr.h
RV_DBTR_DECLARE_BIT, 344
RV_DBTR_DECLARE_BIT_MASK, 344
sbi_dbtr_disable_trig, 344
sbi_dbtr_enable_trig, 345
sbi_dbtr_get_total_triggers, 345
sbi_dbtr_init, 346
sbi_dbtr_install_trig, 346
sbi_dbtr_num_trig, 347
sbi_dbtr_read_trig, 347
sbi_dbtr_setup_shmem, 348
SBI_DBTR_SHMEM_ALIGN_MASK, 343
sbi_dbtr_supported, 348
sbi_dbtr_uninstall_trig, 348
sbi_dbtr_update_trig, 349
SET_TRIG_HW_INDEX, 343
TDATA1_GET_TYPE, 343
sbi_dbtr_data_msg, 124
tdata1, 125
tdata2, 125
tdata3, 125
tstate, 125
sbi_dbtr_disable_shmem
sbi_dbtr.c, 830
sbi_dbtr_disable_trig
sbi_dbtr.c, 831
sbi_dbtr.h, 344
sbi_dbtr_enable_trig
sbi_dbtr.c, 831
sbi_dbtr.h, 345
sbi_dbtr_get_total_triggers
sbi_dbtr.c, 832
sbi_dbtr.h, 345
sbi_dbtr_hart_triggers_state, 125
available_trigs, 125
hartid, 125
probed, 125
shmem, 125
total_trigs, 125
triggers, 126
sbi_dbtr_id_msg, 126
idx, 126
sbi_dbtr_init
sbi_dbtr.c, 832
sbi_dbtr.h, 346
sbi_dbtr_install_trig
sbi_dbtr.c, 833
sbi_dbtr.h, 346
sbi_dbtr_num_trig
sbi_dbtr.c, 833
sbi_dbtr.h, 347
sbi_dbtr_read_trig
sbi_dbtr.c, 834
sbi_dbtr.h, 347
sbi_dbtr_setup_shmem
sbi_dbtr.c, 834
sbi_dbtr.h, 348
sbi_dbtr_shmem, 126
phys_hi, 126
phys_lo, 126
SBI_DBTR_SHMEM_ALIGN_MASK
sbi_dbtr.h, 343
sbi_dbtr_shmem_disabled
sbi_dbtr.c, 835
sbi_dbtr_shmem_entry, 126
data, 127
id, 127
sbi_dbtr_supported
sbi_dbtr.h, 348
sbi_dbtr_trigger, 127
index, 127
state, 127
tdata1, 127
tdata2, 127
tdata3, 127
type_mask, 127
sbi_dbtr_uninstall_trig
sbi_dbtr.c, 835
sbi_dbtr.h, 348
sbi_dbtr_update_trig
sbi_dbtr.c, 836
sbi_dbtr.h, 349
sbi_dlist, 128
next, 128
prev, 128
sbi_domain, 128
assigned_harts, 128
assigned_harts_lock, 128
boot_hartid, 128
data_priv, 129
fw_region_inited, 129
index, 129
name, 129
next_addr, 129
next_arg1, 129
next_mode, 129
node, 129
possible_harts, 129

regions, 129
system_reset_allowed, 129
system_suspend_allowed, 129
sbi_domain.c
clear_region, 839
domain_count, 856
domain_finalized, 856
domain_hart_ptr_offset, 856
find_next_subset_region, 840
find_region, 840
is_region_before, 841
is_region_compatible, 841
is_region_subset, 842
is_region_valid, 843
root, 856
root_add_memregion, 843
root_memregs_count, 856
ROOT_REGION_MAX, 839
sanitize_domain, 844
sbi_domain_check_addr, 845
sbi_domain_check_addr_range, 846
sbi_domain_dump, 847
sbi_domain_dump_all, 847
sbi_domain_finalize, 848
sbi_domain_get_assigned_hartmask, 849
sbi_domain_init, 849
sbi_domain_is_assigned_hart, 850
sbi_domain_memregion_init, 851
sbi_domain_register, 852
sbi_domain_root_add_memrange, 853
sbi_hartindex_to_domain, 854
SBI_LIST_HEAD, 855
sbi_update_hartindex_to_domain, 855
swap_region, 855
sbi_domain.h
domain_list, 367
root, 367
sbi_domain_access, 357
sbi_domain_check_addr, 357
sbi_domain_check_addr_range, 358
sbi_domain_dump, 359
sbi_domain_dump_all, 359
SBI_DOMAIN_EXECUTE, 357
sbi_domain_finalize, 360
sbi_domain_for_each, 353
sbi_domain_for_each_memregion, 353
sbi_domain_get_assigned_hartmask, 361
sbi_domain_init, 361
sbi_domain_is_assigned_hart, 362
SBI_DOMAIN_MEMREGION_ACCESS_MASK, 353
SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE, 353
SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS, 353
SBI_DOMAIN_MEMREGION_ENF_READABLE, 353
SBI_DOMAIN_MEMREGION_ENF_WRITABLE, 353
SBI_DOMAIN_MEMREGION_EXECUTABLE, 354
sbi_domain_memregion_init, 363
SBI_DOMAIN_MEMREGION_IS_SHARED, 354
SBI_DOMAIN_MEMREGION_IS_SUR_MR, 354
SBI_DOMAIN_MEMREGION_IS_SUR_MRW, 354
SBI_DOMAIN_MEMREGION_IS_SURW_MRW, 354
SBI_DOMAIN_MEMREGION_IS_SUX_MRX, 354
SBI_DOMAIN_MEMREGION_IS_SUX_MX, 354
SBI_DOMAIN_MEMREGION_M_ACCESS_MASK, 354
SBI_DOMAIN_MEMREGION_M_EXECUTABLE, 355
SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS, 355
SBI_DOMAIN_MEMREGION_M_READABLE, 355
SBI_DOMAIN_MEMREGION_M_RWX, 355
SBI_DOMAIN_MEMREGION_M_WRITABLE, 355
SBI_DOMAIN_MEMREGION_MMIO, 355
SBI_DOMAIN_MEMREGION_READABLE, 355
SBI_DOMAIN_MEMREGION_SHARED_RDONLY, 355
SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW, 355
SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW, 355
SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX, 356
SBI_DOMAIN_MEMREGION_SHARED_SUX_MX, 356
SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK, 356
SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT, 356
SBI_DOMAIN_MEMREGION_SU_EXECUTABLE, 356
SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS, 356
SBI_DOMAIN_MEMREGION_SU_READABLE, 356
SBI_DOMAIN_MEMREGION_SU_RWX, 356
SBI_DOMAIN_MEMREGION_SU_WRITABLE, 356
SBI_DOMAIN_MEMREGION_WRITEABLE, 356
SBI_DOMAIN_MMIO, 357
SBI_DOMAIN_READ, 357
sbi_domain_register, 364
sbi_domain_root_add_memrange, 365
sbi_domain_thishart_ptr, 356
SBI_DOMAIN_WRITE, 357
sbi_hartindex_to_domain, 366
sbi_update_hartindex_to_domain, 367
sbi_domain_access
sbi_domain.h, 357
sbi_domain_check_addr
sbi_domain.c, 845

sbi_domain.h, 357
sbi_domain_check_addr_range
sbi_domain.c, 846
sbi_domain.h, 358
sbi_domain_cleanup_data
sbi_domain_data.c, 863
sbi_domain_data.h, 374
sbi_domain_context.c
dcpriv, 861
hart_context_get, 857
hart_context_set, 858
hart_context_thishart_get, 857
sbi_domain_context_deinit, 858
sbi_domain_context_enter, 859
sbi_domain_context_exit, 859
sbi_domain_context_init, 860
switch_to_next_domain_context, 860
sbi_domain_context.h
sbi_domain_context_deinit, 370
sbi_domain_context_enter, 371
sbi_domain_context_exit, 371
sbi_domain_context_init, 372
sbi_domain_context_deinit
sbi_domain_context.c, 858
sbi_domain_context.h, 370
sbi_domain_context_enter
sbi_domain_context.c, 859
sbi_domain_context.h, 371
sbi_domain_context_exit
sbi_domain_context.c, 859
sbi_domain_context.h, 371
sbi_domain_context_init
sbi_domain_context.c, 860
sbi_domain_context.h, 372
sbi_domain_data, 130
data_cleanup, 130
data_idx, 130
data_setup, 130
data_size, 130
head, 130
sbi_domain_data.c
DECLARE_BITMAP, 862
domain_cleanup_data_one, 862
domain_setup_data_one, 863
sbi_domain_cleanup_data, 863
sbi_domain_data_ptr, 864
sbi_domain_register_data, 864
sbi_domain_setup_data, 865
sbi_domain_unregister_data, 865
SBI_LIST_HEAD, 866
sbi_domain_data.h
sbi_domain_cleanup_data, 374
sbi_domain_data_ptr, 374
SBI_DOMAIN_MAX_DATA_PTRS, 374
sbi_domain_register_data, 375
sbi_domain_setup_data, 375
sbi_domain_unregister_data, 376
sbi_domain_data_priv, 131
idx_to_data_ptr, 131
sbi_domain_data_ptr
sbi_domain_data.c, 864
sbi_domain_data.h, 374
sbi_domain_dump
sbi_domain.c, 847
sbi_domain.h, 359
sbi_domain_dump_all
sbi_domain.c, 847
sbi_domain.h, 359
SBI_DOMAIN_EXECUTE
sbi_domain.h, 357
sbi_domain_finalize
sbi_domain.c, 848
sbi_domain.h, 360
sbi_domain_for_each
sbi_domain.h, 353
sbi_domain_for_each_memregion
sbi_domain.h, 353
sbi_domain_get_assigned_hartmask
sbi_domain.c, 849
sbi_domain.h, 361
sbi_domain_init
sbi_domain.c, 849
sbi_domain.h, 361
sbi_domain_is_assigned_hart
sbi_domain.c, 850
sbi_domain.h, 362
SBI_DOMAIN_MAX_DATA_PTRS
sbi_domain_data.h, 374
sbi_domain_memregion, 131
base, 131
flags, 131
order, 131
SBI_DOMAIN_MEMREGION_ACCESS_MASK
sbi_domain.h, 353
SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE
sbi_domain.h, 353
SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS
sbi_domain.h, 353
SBI_DOMAIN_MEMREGION_ENF_READABLE
sbi_domain.h, 353
SBI_DOMAIN_MEMREGION_ENF_WRITABLE
sbi_domain.h, 353
SBI_DOMAIN_MEMREGION_EXECUTABLE
sbi_domain.h, 354
sbi_domain_memregion_init
sbi_domain.c, 851
sbi_domain.h, 363
SBI_DOMAIN_MEMREGION_IS_SHARED
sbi_domain.h, 354
SBI_DOMAIN_MEMREGION_IS_SUR_MR
sbi_domain.h, 354
SBI_DOMAIN_MEMREGION_IS_SUR_MRW
sbi_domain.h, 354
SBI_DOMAIN_MEMREGION_IS_SURW_MRW
sbi_domain.h, 354
SBI_DOMAIN_MEMREGION_IS_SUX_MRX

sbi_domain.h, 354
SBI_DOMAIN_MEMREGION_IS_SUX_MX
sbi_domain.h, 354
SBI_DOMAIN_MEMREGION_M_ACCESS_MASK
sbi_domain.h, 354
SBI_DOMAIN_MEMREGION_M_EXECUTABLE
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_M_READABLE
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_M_RWX
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_M_WRITABLE
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_MMIO
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_READABLE
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_SHARED_RDONLY
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW
sbi_domain.h, 355
SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_SHARED_SUX_MX
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_SU_EXECUTABLE
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_SU_READABLE
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_SU_RWX
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_SU_WRITABLE
sbi_domain.h, 356
SBI_DOMAIN_MEMREGION_WRITEABLE
sbi_domain.h, 356
SBI_DOMAIN_MMIO
sbi_domain.h, 357
SBI_DOMAIN_READ
sbi_domain.h, 357
sbi_domain_register
sbi_domain.c, 852
sbi_domain.h, 364
sbi_domain_register_data
sbi_domain_data.c, 864
sbi_domain_data.h, 375
sbi_domain_root_add_memrange
sbi_domain.c, 853
sbi_domain.h, 365
sbi_domain_setup_data
sbi_domain_data.c, 865
sbi_domain_data.h, 375
sbi_domain_thishart_ptr
sbi_domain.h, 356
sbi_domain_unregister_data
sbi_domain_data.c, 865
sbi_domain_data.h, 376
SBI_DOMAIN_WRITE
sbi_domain.h, 357
sbi_double_trap.c
sbi_double_trap_handler, 867
sbi_double_trap_handler
sbi_double_trap.c, 867
sbi_trap_ldst.h, 614
sbi_dprintf
sbi_console.c, 813
SBI_EALREADY
sbi_error.h, 416
SBI_EALREADY_STARTED
sbi_error.h, 416
SBI_EALREADY_STOPPED
sbi_error.h, 416
SBI_EBAD_RANGE
sbi_error.h, 416
sbi_ecall.c
ecall_impid, 872
sbi_ecall_exts, 872
sbi_ecall_find_extension, 868
sbi_ecall_get_extensions_str, 869
sbi_ecall_get_impid, 869
sbi_ecall_handler, 869
sbi_ecall_init, 870
sbi_ecall_register_extension, 870
sbi_ecall_set_impid, 871
sbi_ecall_unregister_extension, 871
sbi_ecall_version_major, 872
sbi_ecall_version_minor, 872
SBI_LIST_HEAD, 872
sbi_ecall.h
sbi_ecall_find_extension, 379
sbi_ecall_get_extensions_str, 379
sbi_ecall_get_impid, 379
sbi_ecall_handler, 380
sbi_ecall_init, 380
sbi_ecall_register_extension, 380
sbi_ecall_set_impid, 382
sbi_ecall_unregister_extension, 382
SBI_ECALL_VERSION_MAJOR, 378
sbi_ecall_version_major, 383
SBI_ECALL_VERSION_MINOR, 378
sbi_ecall_version_minor, 383
SBI_OPENSBI_IMPID, 379
sbi_ecall_base.c
ecall_base, 874
sbi_ecall_base_handler, 873
sbi_ecall_base_probe, 873
sbi_ecall_base_register_extensions, 874

sbi_ecall_base_handler
sbi_ecall_base.c, 873
sbi_ecall_base_probe
sbi_ecall_base.c, 873
sbi_ecall_base_register_extensions
sbi_ecall_base.c, 874
sbi_ecall_cppc.c
ecall_cppc, 876
sbi_ecall_cppc_handler, 875
sbi_ecall_cppc_register_extensions, 875
sbi_ecall_cppc_handler
sbi_ecall_cppc.c, 875
sbi_ecall_cppc_register_extensions
sbi_ecall_cppc.c, 875
sbi_ecall_dbcn.c
ecall_dbcn, 877
sbi_ecall_dbcn_handler, 877
sbi_ecall_dbcn_register_extensions, 877
sbi_ecall_dbcn_handler
sbi_ecall_dbcn.c, 877
sbi_ecall_dbcn_register_extensions
sbi_ecall_dbcn.c, 877
sbi_ecall_dbtr.c
ecall_dbtr, 879
sbi_ecall_dbtr_handler, 878
sbi_ecall_dbtr_register_extensions, 878
sbi_ecall_dbtr_handler
sbi_ecall_dbtr.c, 878
sbi_ecall_dbtr_register_extensions
sbi_ecall_dbtr.c, 878
sbi_ecall_extension, 131
experimental, 132
extid_end, 132
extid_start, 132
handle, 132
head, 132
name, 132
probe, 132
register_extensions, 132
sbi_ecall_exts
sbi_ecall.c, 872
sbi_ecall_find_extension
sbi_ecall.c, 868
sbi_ecall.h, 379
sbi_ecall_fwft.c
ecall_fwft, 880
sbi_ecall_fwft_handler, 880
sbi_ecall_fwft_register_extensions, 880
sbi_ecall_fwft_handler
sbi_ecall_fwft.c, 880
sbi_ecall_fwft_register_extensions
sbi_ecall_fwft.c, 880
sbi_ecall_get_extensions_str
sbi_ecall.c, 869
sbi_ecall.h, 379
sbi_ecall_get_impid
sbi_ecall.c, 869
sbi_ecall.h, 379
sbi_ecall_handler
sbi_ecall.c, 869
sbi_ecall.h, 380
sbi_ecall_hsm.c
ecall_hsm, 882
sbi_ecall_hsm_handler, 881
sbi_ecall_hsm_register_extensions, 882
sbi_ecall_hsm_handler
sbi_ecall_hsm.c, 881
sbi_ecall_hsm_register_extensions
sbi_ecall_hsm.c, 882
sbi_ecall_init
sbi_ecall.c, 870
sbi_ecall.h, 380
sbi_ecall_interface.h
SBI_CPPC_ACPI_LAST, 404
SBI_CPPC_AUTO_ACT_WINDOW, 403
SBI_CPPC_AUTO_SEL_ENABLE, 403
SBI_CPPC_CTR_WRAP_TIME, 403
SBI_CPPC_DELIVERED_CTR, 403
SBI_CPPC_DESIRED_PERF, 403
SBI_CPPC_ENABLE, 403
SBI_CPPC_ENERGY_PERF_PREFERENCE, 403
SBI_CPPC_GUARANTEED_PERF, 403
SBI_CPPC_HIGHEST_PERF, 403
SBI_CPPC_LOW_NON_LINEAR_PERF, 403
SBI_CPPC_LOWEST_FREQ, 403
SBI_CPPC_LOWEST_PERF, 403
SBI_CPPC_MAX_PERF, 403
SBI_CPPC_MIN_PERF, 403
SBI_CPPC_NOMINAL_FREQ, 403
SBI_CPPC_NOMINAL_PERF, 403
SBI_CPPC_NON_ACPI_LAST, 404
SBI_CPPC_PERF_LIMITED, 403
SBI_CPPC_PERF_REDUC_TOLERANCE, 403
SBI_CPPC_REFERENCE_CTR, 403
SBI_CPPC_REFERENCE_PERF, 403
sbi_cppc_reg_id, 403
SBI_CPPC_TIME_WINDOW, 403
SBI_CPPC_TRANSITION_LATENCY, 404
SBI_ERR_ALREADY_AVAILABLE, 390
SBI_ERR_ALREADY_STARTED, 390
SBI_ERR_ALREADY_STOPPED, 390
SBI_ERR_BAD_RANGE, 390
SBI_ERR_DENIED, 390
SBI_ERR_FAILED, 390
SBI_ERR_INVALID_ADDRESS, 390
SBI_ERR_INVALID_PARAM, 390
SBI_ERR_INVALID_STATE, 390
SBI_ERR_IO, 390
SBI_ERR_NO_SHMEM, 391
SBI_ERR_NOT_SUPPORTED, 391
SBI_ERR_TIMEOUT, 391
SBI_EXT_0_1_CLEAR_IPI, 391
SBI_EXT_0_1_CONSOLE_GETCHAR, 391
SBI_EXT_0_1_CONSOLE_PUTCHAR, 391
SBI_EXT_0_1_REMOTE_FENCE_I, 391
SBI_EXT_0_1_REMOTE_SFENCE_VMA, 391

SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID, 391
SBI_EXT_0_1_SEND_IPI, 391
SBI_EXT_0_1_SET_TIMER, 391
SBI_EXT_0_1_SHUTDOWN, 391
SBI_EXT_BASE, 391
SBI_EXT_BASE_GET_IMP_ID, 391
SBI_EXT_BASE_GET_IMP_VERSION, 391
SBI_EXT_BASE_GET_MARCHID, 392
SBI_EXT_BASE_GET_MIMPID, 392
SBI_EXT_BASE_GET_MVENDORID, 392
SBI_EXT_BASE_GET_SPEC_VERSION, 392
SBI_EXT_BASE_PROBE_EXT, 392
SBI_EXT_CPPC, 392
SBI_EXT_CPPC_PROBE, 392
SBI_EXT_CPPC_READ, 392
SBI_EXT_CPPC_READ_HI, 392
SBI_EXT_CPPC_WRITE, 392
SBI_EXT_DBCN, 392
SBI_EXT_DBCN_CONSOLE_READ, 392
SBI_EXT_DBCN_CONSOLE_WRITE, 392
SBI_EXT_DBCN_CONSOLE_WRITE_BYTE, 392
SBI_EXT_DBTR, 392
SBI_EXT_DBTR_NUM_TRIGGERS, 393
SBI_EXT_DBTR_SETUP_SHMEM, 393
SBI_EXT_DBTR_TRIGGER_DISABLE, 393
SBI_EXT_DBTR_TRIGGER_ENABLE, 393
SBI_EXT_DBTR_TRIGGER_INSTALL, 393
SBI_EXT_DBTR_TRIGGER_READ, 393
SBI_EXT_DBTR_TRIGGER_UNINSTALL, 393
SBI_EXT_DBTR_TRIGGER_UPDATE, 393
SBI_EXT_FIRMWARE_END, 393
SBI_EXT_FIRMWARE_START, 393
SBI_EXT_FWFT, 393
SBI_EXT_FWFT_GET, 393
SBI_EXT_FWFT_SET, 393
SBI_EXT_HSM, 393
SBI_EXT_HSM_HART_GET_STATUS, 393
SBI_EXT_HSM_HART_START, 394
SBI_EXT_HSM_HART_STOP, 394
SBI_EXT_HSM_HART_SUSPEND, 394
SBI_EXT_IPI, 394
SBI_EXT_IPI_SEND_IPI, 394
SBI_EXT_MPXY, 394
SBI_EXT_MPXY_GET_CHANNEL_IDS, 394
SBI_EXT_MPXY_GET_NOTIFICATION_EVENTS, 394
SBI_EXT_MPXY_READ_ATTRS, 394
SBI_EXT_MPXY_SEND_MSG_NO_RESP, 394
SBI_EXT_MPXY_SEND_MSG_WITH_RESP, 394
SBI_EXT_MPXY_SET_SHMEM, 394
SBI_EXT_MPXY_WRITE_ATTRS, 394
SBI_EXT_PMU, 394
SBI_EXT_PMU_COUNTER_CFG_MATCH, 394
SBI_EXT_PMU_COUNTER_FW_READ, 395
SBI_EXT_PMU_COUNTER_FW_READ_HI, 395
SBI_EXT_PMU_COUNTER_GET_INFO, 395
SBI_EXT_PMU_COUNTER_START, 395
SBI_EXT_PMU_COUNTER_STOP, 395
SBI_EXT_PMU_EVENT_GET_INFO, 395
SBI_EXT_PMU_NUM_COUNTERS, 395
SBI_EXT_PMU_SNAPSHOT_SET_SHMEM, 395
SBI_EXT_RFENCE, 395
SBI_EXT_RFENCE_REMOTE_FENCE_I, 395
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA, 395
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID, 395
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA, 395
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID, 395
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA, 395
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID, 396
SBI_EXT_SRST, 396
SBI_EXT_SRST_RESET, 396
SBI_EXT_SSE, 396
SBI_EXT_SSE_COMPLETE, 396
SBI_EXT_SSE_DISABLE, 396
SBI_EXT_SSE_ENABLE, 396
SBI_EXT_SSE_HART_MASK, 396
SBI_EXT_SSE_HART_UNMASK, 396
SBI_EXT_SSE_INJECT, 396
SBI_EXT_SSE_READ_ATTR, 396
SBI_EXT_SSE_REGISTER, 396
SBI_EXT_SSE_UNREGISTER, 396
SBI_EXT_SSE_WRITE_ATTR, 396
SBI_EXT_SUSP, 396
SBI_EXT_SUSP_SUSPEND, 397
SBI_EXT_TIME, 397
SBI_EXT_TIME_SET_TIMER, 397
SBI_EXT_VENDOR_END, 397
SBI_EXT_VENDOR_START, 397
SBI_FWFT_DOUBLE_TRAP, 404
sbi_fwft_feature_t, 404
SBI_FWFT_GLOBAL_FEATURE_BIT, 397
SBI_FWFT_GLOBAL_PLATFORM_END, 404
SBI_FWFT_GLOBAL_PLATFORM_START, 404
SBI_FWFT_GLOBAL_RESERVED_END, 404
SBI_FWFT_GLOBAL_RESERVED_START, 404
SBI_FWFT_LANDING_PAD, 404
SBI_FWFT_LOCAL_PLATFORM_END, 404
SBI_FWFT_LOCAL_PLATFORM_START, 404
SBI_FWFT_LOCAL_RESERVED_END, 404
SBI_FWFT_LOCAL_RESERVED_START, 404
SBI_FWFT_MISALIGNED_EXC_DELEG, 404
SBI_FWFT_PLATFORM_FEATURE_BIT, 397
SBI_FWFT_POINTER_MASKING_PMLEN, 404
SBI_FWFT_PTE_AD_HW_UPDATING, 404
SBI_FWFT_SET_FLAG_LOCK, 397
SBI_FWFT_SHADOW_STACK, 404
SBI_HSM_STATE_RESUME_PENDING, 397
SBI_HSM_STATE_START_PENDING, 397
SBI_HSM_STATE_STARTED, 397

SBI_HSM_STATE_STOP_PENDING, 397
SBI_HSM_STATE_STOPPED, 397
SBI_HSM_STATE_SUSPEND_PENDING, 397
SBI_HSM_STATE_SUSPENDED, 397
SBI_HSM_SUSP_BASE_MASK, 398
SBI_HSM_SUSP_NON_RET_BIT, 398
SBI_HSM_SUSP_PLAT_BASE, 398
SBI_HSM_SUSPEND_NON_RET_DEFAULT, 398
SBI_HSM_SUSPEND_NON_RET_LAST, 398
SBI_HSM_SUSPEND_NON_RET_PLATFORM, 398
SBI_HSM_SUSPEND_RET_DEFAULT, 398
SBI_HSM_SUSPEND_RET_LAST, 398
SBI_HSM_SUSPEND_RET_PLATFORM, 398
SBI_LAST_ERR, 398
SBI_PMU_CFG_FLAG_AUTO_START, 398
SBI_PMU_CFG_FLAG_CLEAR_VALUE, 398
SBI_PMU_CFG_FLAG_SET_MINH, 398
SBI_PMU_CFG_FLAG_SET_SINH, 398
SBI_PMU_CFG_FLAG_SET_UINH, 399
SBI_PMU_CFG_FLAG_SET_VSINH, 399
SBI_PMU_CFG_FLAG_SET_VUINH, 399
SBI_PMU_CFG_FLAG_SKIP_MATCH, 399
sbi_pmu_ctr_type, 404
SBI_PMU_CTR_TYPE_FW, 404
SBI_PMU_CTR_TYPE_HW, 404
SBI_PMU_EVENT_HW_CACHE_ID_MASK, 399
SBI_PMU_EVENT_HW_CACHE_ID_OFFSET, 399
SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK, 399
SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET, 399
SBI_PMU_EVENT_HW_CACHE_OPS_RESULT, 399
SBI_PMU_EVENT_IDX_CODE_MASK, 399
SBI_PMU_EVENT_IDX_INVALID, 399
SBI_PMU_EVENT_IDX_MASK, 399
SBI_PMU_EVENT_IDX_TYPE_MASK, 399
SBI_PMU_EVENT_IDX_TYPE_OFFSET, 399
SBI_PMU_EVENT_RAW_IDX, 399
SBI_PMU_EVENT_RAW_V2_IDX, 400
SBI_PMU_EVENT_TYPE_FW, 404
SBI_PMU_EVENT_TYPE_HW, 404
SBI_PMU_EVENT_TYPE_HW_CACHE, 404
SBI_PMU_EVENT_TYPE_HW_RAW, 404
SBI_PMU_EVENT_TYPE_HW_RAW_V2, 404
sbi_pmu_event_type_id, 404
SBI_PMU_EVENT_TYPE_MAX, 404
SBI_PMU_FW_ACCESS_LOAD, 405
SBI_PMU_FW_ACCESS_STORE, 405
sbi_pmu_fw_event_code_id, 404
SBI_PMU_FW_FENCE_I_RECVD, 405
SBI_PMU_FW_FENCE_I_SENT, 405
SBI_PMU_FW_HFENCE_GVMA_RCVD, 405
SBI_PMU_FW_HFENCE_GVMA_SENT, 405
SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD, 405
SBI_PMU_FW_HFENCE_GVMA_VMID_SENT, 405
SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD, 405
SBI_PMU_FW_HFENCE_VVMA_ASID_SENT, 405
SBI_PMU_FW_HFENCE_VVMA_RCVD, 405
SBI_PMU_FW_HFENCE_VVMA_SENT, 405
SBI_PMU_FW_ILLEGAL_INSN, 405
SBI_PMU_FW_IPI_RECVD, 405
SBI_PMU_FW_IPI_SENT, 405
SBI_PMU_FW_MAX, 405
SBI_PMU_FW_MISALIGNED_LOAD, 405
SBI_PMU_FW_MISALIGNED_STORE, 405
SBI_PMU_FW_PLATFORM, 405
SBI_PMU_FW_RESERVED_MAX, 405
SBI_PMU_FW_SET_TIMER, 405
SBI_PMU_FW_SFENCE_VMA_ASID_RCVD, 405
SBI_PMU_FW_SFENCE_VMA_ASID_SENT, 405
SBI_PMU_FW_SFENCE_VMA_RCVD, 405
SBI_PMU_FW_SFENCE_VMA_SENT, 405
SBI_PMU_HW_BRANCH_INSTRUCTIONS, 406
SBI_PMU_HW_BRANCH_MISSES, 406
SBI_PMU_HW_BUS_CYCLES, 406
SBI_PMU_HW_CACHE_BPU, 405
SBI_PMU_HW_CACHE_DTLB, 405
sbi_pmu_hw_cache_id, 405
SBI_PMU_HW_CACHE_ITLB, 405
SBI_PMU_HW_CACHE_L1D, 405
SBI_PMU_HW_CACHE_L1I, 405
SBI_PMU_HW_CACHE_LL, 405
SBI_PMU_HW_CACHE_MAX, 405
SBI_PMU_HW_CACHE_MISSES, 406
SBI_PMU_HW_CACHE_NODE, 405
sbi_pmu_hw_cache_op_id, 406
SBI_PMU_HW_CACHE_OP_MAX, 406
SBI_PMU_HW_CACHE_OP_PREFETCH, 406
SBI_PMU_HW_CACHE_OP_READ, 406
sbi_pmu_hw_cache_op_result_id, 406
SBI_PMU_HW_CACHE_OP_WRITE, 406
SBI_PMU_HW_CACHE_REFERENCES, 406
SBI_PMU_HW_CACHE_RESULT_ACCESS, 406
SBI_PMU_HW_CACHE_RESULT_MAX, 406
SBI_PMU_HW_CACHE_RESULT_MISS, 406
SBI_PMU_HW_CPU_CYCLES, 406
SBI_PMU_HW_GENERAL_MAX, 406
sbi_pmu_hw_generic_events_t, 406
SBI_PMU_HW_INSTRUCTIONS, 406
SBI_PMU_HW_NO_EVENT, 406
SBI_PMU_HW_REF_CPU_CYCLES, 406
SBI_PMU_HW_STALLED_CYCLES_BACKEND, 406
SBI_PMU_HW_STALLED_CYCLES_FRONTEND, 406
SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT, 400
SBI_PMU_START_FLAG_SET_INIT_VALUE, 400
SBI_PMU_STOP_FLAG_RESET, 400

SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT, 400
SBI_SPEC_VERSION_MAJOR_MASK, 400
SBI_SPEC_VERSION_MAJOR_OFFSET, 400
SBI_SPEC_VERSION_MINOR_MASK, 400
SBI_SRST_RESET_REASON_NONE, 400
SBI_SRST_RESET_REASON_SYSFAIL, 400
SBI_SRST_RESET_TYPE_COLD_REBOOT, 400
SBI_SRST_RESET_TYPE_LAST, 400
SBI_SRST_RESET_TYPE_SHUTDOWN, 400
SBI_SRST_RESET_TYPE_WARM_REBOOT, 400
SBI_SSE_ATTR_CONFIG, 406
SBI_SSE_ATTR_CONFIG_ONESHOT, 400
SBI_SSE_ATTR_ENTRY_ARG, 406
SBI_SSE_ATTR_ENTRY_PC, 406
sbi_sse_attr_id, 406
SBI_SSE_ATTR_INTERRUPTED_A6, 407
SBI_SSE_ATTR_INTERRUPTED_A7, 407
SBI_SSE_ATTR_INTERRUPTED_FLAGS, 407
SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV, 401
SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP, 401
SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE, 401
SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP, 401
SBI_SSE_ATTR_INTERRUPTED_SEPC, 407
SBI_SSE_ATTR_MAX, 407
SBI_SSE_ATTR_PREFERRED_HART, 406
SBI_SSE_ATTR_PRIO, 406
SBI_SSE_ATTR_STATUS, 406
SBI_SSE_ATTR_STATUS_INJECT_OFFSET, 401
SBI_SSE_ATTR_STATUS_PENDING_OFFSET, 401
SBI_SSE_ATTR_STATUS_STATE_MASK, 401
SBI_SSE_ATTR_STATUS_STATE_OFFSET, 401
SBI_SSE_EVENT_GLOBAL_BIT, 401
SBI_SSE_EVENT_GLOBAL_PLAT_0_END, 401
SBI_SSE_EVENT_GLOBAL_PLAT_0_START, 401
SBI_SSE_EVENT_GLOBAL_PLAT_1_END, 401
SBI_SSE_EVENT_GLOBAL_PLAT_1_START, 401
SBI_SSE_EVENT_GLOBAL_PLAT_2_END, 401
SBI_SSE_EVENT_GLOBAL_PLAT_2_START, 401
SBI_SSE_EVENT_GLOBAL_PLAT_3_END, 402
SBI_SSE_EVENT_GLOBAL_PLAT_3_START, 402
SBI_SSE_EVENT_GLOBAL_RAS, 402
SBI_SSE_EVENT_GLOBAL_SOFTWARE, 402
SBI_SSE_EVENT_LOCAL_DOUBLE_TRAP, 402
SBI_SSE_EVENT_LOCAL_PLAT_0_END, 402
SBI_SSE_EVENT_LOCAL_PLAT_0_START, 402
SBI_SSE_EVENT_LOCAL_PLAT_1_END, 402
SBI_SSE_EVENT_LOCAL_PLAT_1_START, 402
SBI_SSE_EVENT_LOCAL_PLAT_2_END, 402
SBI_SSE_EVENT_LOCAL_PLAT_2_START, 402
SBI_SSE_EVENT_LOCAL_PLAT_3_END, 402
SBI_SSE_EVENT_LOCAL_PLAT_3_START, 402
SBI_SSE_EVENT_LOCAL_PMU, 402
SBI_SSE_EVENT_LOCAL_RAS, 402
SBI_SSE_EVENT_LOCAL_SOFTWARE, 403
SBI_SSE_EVENT_PLATFORM_BIT, 403
sbi_sse_state, 407
SBI_SSE_STATE_ENABLED, 407
SBI_SSE_STATE_REGISTERED, 407
SBI_SSE_STATE_RUNNING, 407
SBI_SSE_STATE_UNUSED, 407
SBI_SUCCESS, 403
SBI_SUSP_PLATFORM_SLEEP_START, 403
SBI_SUSP_SLEEP_TYPE_LAST, 403
SBI_SUSP_SLEEP_TYPE_SUSPEND, 403
sbi_ecall_ipi.c
ecall_ipi, 884
sbi_ecall_ipi_handler, 883
sbi_ecall_ipi_register_extensions, 884
sbi_ecall_ipi_handler
sbi_ecall_ipi.c, 883
sbi_ecall_ipi_register_extensions
sbi_ecall_ipi.c, 884
sbi_ecall_legacy.c
ecall_legacy, 886
sbi_ecall_legacy_handler, 885
sbi_ecall_legacy_register_extensions, 885
sbi_load_hart_mask_unpriv, 886
sbi_ecall_legacy_handler
sbi_ecall_legacy.c, 885
sbi_ecall_legacy_register_extensions
sbi_ecall_legacy.c, 885
sbi_ecall_mpxy.c
ecall_mpxy, 888
sbi_ecall_mpxy_handler, 887
sbi_ecall_mpxy_register_extensions, 887
sbi_ecall_mpxy_handler
sbi_ecall_mpxy.c, 887
sbi_ecall_mpxy_register_extensions
sbi_ecall_mpxy.c, 887
sbi_ecall_pmu.c
ecall_pmu, 890
sbi_ecall_pmu_handler, 889
sbi_ecall_pmu_register_extensions, 889
sbi_ecall_pmu_handler
sbi_ecall_pmu.c, 889
sbi_ecall_pmu_register_extensions
sbi_ecall_pmu.c, 889
sbi_ecall_register_extension
sbi_ecall.c, 870
sbi_ecall.h, 380
sbi_ecall_return, 132
skip_regs_update, 133
value, 133
sbi_ecall_rfence.c
ecall_rfence, 891
sbi_ecall_rfence_handler, 890
sbi_ecall_rfence_register_extensions, 891
sbi_ecall_rfence_handler
sbi_ecall_rfence.c, 890
sbi_ecall_rfence_register_extensions
sbi_ecall_rfence.c, 891

sbi_ecall_set_impid
sbi_ecall.c, 871
sbi_ecall.h, 382
sbi_ecall_srst.c
ecall_srst, 894
sbi_ecall_srst_handler, 892
sbi_ecall_srst_register_extensions, 893
srst_available, 893
sbi_ecall_srst_handler
sbi_ecall_srst.c, 892
sbi_ecall_srst_register_extensions
sbi_ecall_srst.c, 893
sbi_ecall_sse.c
ecall_sse, 896
sbi_ecall_sse_handler, 895
sbi_ecall_sse_register_extensions, 895
sbi_ecall_sse_handler
sbi_ecall_sse.c, 895
sbi_ecall_sse_register_extensions
sbi_ecall_sse.c, 895
sbi_ecall_susp.c
ecall_susp, 898
sbi_ecall_susp_handler, 896
sbi_ecall_susp_register_extensions, 897
susp_available, 897
sbi_ecall_susp_handler
sbi_ecall_susp.c, 896
sbi_ecall_susp_register_extensions
sbi_ecall_susp.c, 897
sbi_ecall_time.c
ecall_time, 899
sbi_ecall_time_handler, 899
sbi_ecall_time_register_extensions, 899
sbi_ecall_time_handler
sbi_ecall_time.c, 899
sbi_ecall_time_register_extensions
sbi_ecall_time.c, 899
sbi_ecall_unregister_extension
sbi_ecall.c, 871
sbi_ecall.h, 382
sbi_ecall_vendor.c
ecall_vendor, 901
sbi_ecall_vendor_handler, 900
sbi_ecall_vendor_id, 901
sbi_ecall_vendor_register_extensions, 901
sbi_ecall_vendor_handler
sbi_ecall_vendor.c, 900
sbi_ecall_vendor_id
sbi_ecall_vendor.c, 901
sbi_ecall_vendor_register_extensions
sbi_ecall_vendor.c, 901
SBI_ECALL_VERSION_MAJOR
sbi_ecall.h, 378
sbi_ecall_version_major
sbi_ecall.c, 872
sbi_ecall.h, 383
SBI_ECALL_VERSION_MINOR
sbi_ecall.h, 378
sbi_ecall_version_minor
sbi_ecall.c, 872
sbi_ecall.h, 383
SBI_EDENIED
sbi_error.h, 416
SBI_EFAIL
sbi_error.h, 416
SBI_EILL
sbi_error.h, 416
SBI_EINVAL
sbi_error.h, 416
SBI_EINVALID_ADDR
sbi_error.h, 416
SBI_EINVALID_STATE
sbi_error.h, 416
SBI_EIO
sbi_error.h, 416
sbi_emulate_csr.c
hpm_allowed, 903
sbi_emulate_csr_read, 904
sbi_emulate_csr_write, 904
switchcase_hpm, 902
switchcase_hpm_16, 902
switchcase_hpm_2, 903
switchcase_hpm_4, 903
switchcase_hpm_8, 903
sbi_emulate_csr.h
sbi_emulate_csr_read, 413
sbi_emulate_csr_write, 414
sbi_emulate_csr_read
sbi_emulate_csr.c, 904
sbi_emulate_csr.h, 413
sbi_emulate_csr_write
sbi_emulate_csr.c, 904
sbi_emulate_csr.h, 414
SBI_ENO_SHMEM
sbi_error.h, 416
SBI_ENODEV
sbi_error.h, 416
SBI_ENOENT
sbi_error.h, 416
SBI_ENOMEM
sbi_error.h, 416
SBI_ENOSPC
sbi_error.h, 417
SBI_ENOSYS
sbi_error.h, 417
SBI_ENOTSUPP
sbi_error.h, 417
sbi_entry_count
sbi_init.c, 978
sbi_init.h, 481
SBI_ERR_ALREADY_AVAILABLE
sbi_ecall_interface.h, 390
SBI_ERR_ALREADY_STARTED
sbi_ecall_interface.h, 390
SBI_ERR_ALREADY_STOPPED
sbi_ecall_interface.h, 390

SBI_ERR_BAD_RANGE
sbi_ecall_interface.h, 390
SBI_ERR_DENIED
sbi_ecall_interface.h, 390
SBI_ERR_FAILED
sbi_ecall_interface.h, 390
SBI_ERR_INVALID_ADDRESS
sbi_ecall_interface.h, 390
SBI_ERR_INVALID_PARAM
sbi_ecall_interface.h, 390
SBI_ERR_INVALID_STATE
sbi_ecall_interface.h, 390
SBI_ERR_IO
sbi_ecall_interface.h, 390
SBI_ERR_NO_SHMEM
sbi_ecall_interface.h, 391
SBI_ERR_NOT_SUPPORTED
sbi_ecall_interface.h, 391
SBI_ERR_TIMEOUT
sbi_ecall_interface.h, 391
sbi_error.h
SBI_EALREADY, 416
SBI_EALREADY_STARTED, 416
SBI_EALREADY_STOPPED, 416
SBI_EBAD_RANGE, 416
SBI_EDENIED, 416
SBI_EFAIL, 416
SBI_EILL, 416
SBI_EINVAL, 416
SBI_EINVALID_ADDR, 416
SBI_EINVALID_STATE, 416
SBI_EIO, 416
SBI_ENO_SHMEM, 416
SBI_ENODEV, 416
SBI_ENOENT, 416
SBI_ENOMEM, 416
SBI_ENOSPC, 417
SBI_ENOSYS, 417
SBI_ENOTSUPP, 417
SBI_ETIMEDOUT, 417
SBI_ETIMEOUT, 417
SBI_EUNKNOWN, 417
SBI_OK, 417
SBI_ETIMEDOUT
sbi_error.h, 417
SBI_ETIMEOUT
sbi_error.h, 417
SBI_EUNKNOWN
sbi_error.h, 417
sbi_exit
sbi_init.c, 978
sbi_init.h, 481
SBI_EXT_0_1_CLEAR_IPI
sbi_ecall_interface.h, 391
SBI_EXT_0_1_CONSOLE_GETCHAR
sbi_ecall_interface.h, 391
SBI_EXT_0_1_CONSOLE_PUTCHAR
sbi_ecall_interface.h, 391
SBI_EXT_0_1_REMOTE_FENCE_I
sbi_ecall_interface.h, 391
SBI_EXT_0_1_REMOTE_SFENCE_VMA
sbi_ecall_interface.h, 391
SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID
sbi_ecall_interface.h, 391
SBI_EXT_0_1_SEND_IPI
sbi_ecall_interface.h, 391
SBI_EXT_0_1_SET_TIMER
sbi_ecall_interface.h, 391
SBI_EXT_0_1_SHUTDOWN
sbi_ecall_interface.h, 391
SBI_EXT_BASE
sbi_ecall_interface.h, 391
SBI_EXT_BASE_GET_IMP_ID
sbi_ecall_interface.h, 391
SBI_EXT_BASE_GET_IMP_VERSION
sbi_ecall_interface.h, 391
SBI_EXT_BASE_GET_MARCHID
sbi_ecall_interface.h, 392
SBI_EXT_BASE_GET_MIMPID
sbi_ecall_interface.h, 392
SBI_EXT_BASE_GET_MVENDORID
sbi_ecall_interface.h, 392
SBI_EXT_BASE_GET_SPEC_VERSION
sbi_ecall_interface.h, 392
SBI_EXT_BASE_PROBE_EXT
sbi_ecall_interface.h, 392
SBI_EXT_CPPC
sbi_ecall_interface.h, 392
SBI_EXT_CPPC_PROBE
sbi_ecall_interface.h, 392
SBI_EXT_CPPC_READ
sbi_ecall_interface.h, 392
SBI_EXT_CPPC_READ_HI
sbi_ecall_interface.h, 392
SBI_EXT_CPPC_WRITE
sbi_ecall_interface.h, 392
SBI_EXT_DBCN
sbi_ecall_interface.h, 392
SBI_EXT_DBCN_CONSOLE_READ
sbi_ecall_interface.h, 392
SBI_EXT_DBCN_CONSOLE_WRITE
sbi_ecall_interface.h, 392
SBI_EXT_DBCN_CONSOLE_WRITE_BYTE
sbi_ecall_interface.h, 392
SBI_EXT_DBTR
sbi_ecall_interface.h, 392
SBI_EXT_DBTR_NUM_TRIGGERS
sbi_ecall_interface.h, 393
SBI_EXT_DBTR_SETUP_SHMEM
sbi_ecall_interface.h, 393
SBI_EXT_DBTR_TRIGGER_DISABLE
sbi_ecall_interface.h, 393
SBI_EXT_DBTR_TRIGGER_ENABLE
sbi_ecall_interface.h, 393
SBI_EXT_DBTR_TRIGGER_INSTALL
sbi_ecall_interface.h, 393

SBI_EXT_DBTR_TRIGGER_READ
sbi_ecall_interface.h, 393
SBI_EXT_DBTR_TRIGGER_UNINSTALL
sbi_ecall_interface.h, 393
SBI_EXT_DBTR_TRIGGER_UPDATE
sbi_ecall_interface.h, 393
SBI_EXT_FIRMWARE_END
sbi_ecall_interface.h, 393
SBI_EXT_FIRMWARE_START
sbi_ecall_interface.h, 393
SBI_EXT_FWFT
sbi_ecall_interface.h, 393
SBI_EXT_FWFT_GET
sbi_ecall_interface.h, 393
SBI_EXT_FWFT_SET
sbi_ecall_interface.h, 393
SBI_EXT_HSM
sbi_ecall_interface.h, 393
SBI_EXT_HSM_HART_GET_STATUS
sbi_ecall_interface.h, 393
SBI_EXT_HSM_HART_START
sbi_ecall_interface.h, 394
SBI_EXT_HSM_HART_STOP
sbi_ecall_interface.h, 394
SBI_EXT_HSM_HART_SUSPEND
sbi_ecall_interface.h, 394
SBI_EXT_IPI
sbi_ecall_interface.h, 394
SBI_EXT_IPI_SEND_IPI
sbi_ecall_interface.h, 394
SBI_EXT_MPXY
sbi_ecall_interface.h, 394
SBI_EXT_MPXY_GET_CHANNEL_IDS
sbi_ecall_interface.h, 394
SBI_EXT_MPXY_GET_NOTIFICATION_EVENTS
sbi_ecall_interface.h, 394
SBI_EXT_MPXY_READ_ATTRS
sbi_ecall_interface.h, 394
SBI_EXT_MPXY_SEND_MSG_NO_RESP
sbi_ecall_interface.h, 394
SBI_EXT_MPXY_SEND_MSG_WITH_RESP
sbi_ecall_interface.h, 394
SBI_EXT_MPXY_SET_SHMEM
sbi_ecall_interface.h, 394
SBI_EXT_MPXY_SHMEM_FLAG_MAX_IDX
sbi_mpxy.h, 512
SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE
sbi_mpxy.h, 512
SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE_RETURN
sbi_mpxy.h, 512
SBI_EXT_MPXY_SHMEM_FLAGS
sbi_mpxy.h, 512
SBI_EXT_MPXY_WRITE_ATTRS
sbi_ecall_interface.h, 394
SBI_EXT_PMU
sbi_ecall_interface.h, 394
SBI_EXT_PMU_COUNTER_CFG_MATCH
sbi_ecall_interface.h, 394
SBI_EXT_PMU_COUNTER_FW_READ
sbi_ecall_interface.h, 395
SBI_EXT_PMU_COUNTER_FW_READ_HI
sbi_ecall_interface.h, 395
SBI_EXT_PMU_COUNTER_GET_INFO
sbi_ecall_interface.h, 395
SBI_EXT_PMU_COUNTER_START
sbi_ecall_interface.h, 395
SBI_EXT_PMU_COUNTER_STOP
sbi_ecall_interface.h, 395
SBI_EXT_PMU_EVENT_GET_INFO
sbi_ecall_interface.h, 395
SBI_EXT_PMU_NUM_COUNTERS
sbi_ecall_interface.h, 395
SBI_EXT_PMU_SNAPSHOT_SET_SHMEM
sbi_ecall_interface.h, 395
SBI_EXT_RFENCE
sbi_ecall_interface.h, 395
SBI_EXT_RFENCE_REMOTE_FENCE_I
sbi_ecall_interface.h, 395
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA
sbi_ecall_interface.h, 395
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID
sbi_ecall_interface.h, 395
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA
sbi_ecall_interface.h, 395
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID
sbi_ecall_interface.h, 395
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA
sbi_ecall_interface.h, 395
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID
sbi_ecall_interface.h, 396
SBI_EXT_SRST
sbi_ecall_interface.h, 396
SBI_EXT_SRST_RESET
sbi_ecall_interface.h, 396
SBI_EXT_SSE
sbi_ecall_interface.h, 396
SBI_EXT_SSE_COMPLETE
sbi_ecall_interface.h, 396
SBI_EXT_SSE_DISABLE
sbi_ecall_interface.h, 396
SBI_EXT_SSE_ENABLE
sbi_ecall_interface.h, 396
SBI_EXT_SSE_HART_MASK
sbi_ecall_interface.h, 396
SBI_EXT_SSE_HART_UNMASK
sbi_ecall_interface.h, 396
SBI_EXT_SSE_INJECT
sbi_ecall_interface.h, 396
SBI_EXT_SSE_READ_ATTR
sbi_ecall_interface.h, 396
SBI_EXT_SSE_REGISTER
sbi_ecall_interface.h, 396
SBI_EXT_SSE_UNREGISTER
sbi_ecall_interface.h, 396
SBI_EXT_SSE_WRITE_ATTR
sbi_ecall_interface.h, 396

SBI_EXT_SUSP
sbi_ecall_interface.h, 396
SBI_EXT_SUSP_SUSPEND
sbi_ecall_interface.h, 397
SBI_EXT_TIME
sbi_ecall_interface.h, 397
SBI_EXT_TIME_SET_TIMER
sbi_ecall_interface.h, 397
SBI_EXT_VENDOR_END
sbi_ecall_interface.h, 397
SBI_EXT_VENDOR_START
sbi_ecall_interface.h, 397
sbi_ffs
sbi_bitops.h, 319
sbi_ffz
sbi_bitops.h, 315
sbi_fifo, 133
avail, 133
entry_size, 133
num_entries, 133
qlock, 133
queue, 133
tail, 133
sbi_fifo.c
__sbi_fifo_dequeue, 906
__sbi_fifo_enqueue, 906
__sbi_fifo_is_empty, 907
__sbi_fifo_is_full, 908
__sbi_fifo_reset, 908
sbi_fifo_avail, 909
sbi_fifo_dequeue, 909
sbi_fifo_enqueue, 910
sbi_fifo_init, 911
sbi_fifo_inplace_update, 912
sbi_fifo_is_empty, 912
sbi_fifo_is_full, 913
sbi_fifo_reset, 913
sbi_fifo.h
sbi_fifo_avail, 419
SBI_FIFO_DEFINE, 419
sbi_fifo_dequeue, 420
sbi_fifo_enqueue, 421
sbi_fifo_init, 422
SBI_FIFO_INITIALIZER, 419
sbi_fifo_inplace_update, 422
sbi_fifo_inplace_update_types, 419
sbi_fifo_is_empty, 423
sbi_fifo_is_full, 423
SBI_FIFO_SKIP, 419
SBI_FIFO_UNCHANGED, 419
SBI_FIFO_UPDATED, 419
sbi_fifo_avail
sbi_fifo.c, 909
sbi_fifo.h, 419
SBI_FIFO_DEFINE
sbi_console.c, 814
sbi_fifo.h, 419
sbi_fifo_dequeue
sbi_fifo.c, 909
sbi_fifo.h, 420
sbi_fifo_enqueue
sbi_fifo.c, 910
sbi_fifo.h, 421
sbi_fifo_init
sbi_fifo.c, 911
sbi_fifo.h, 422
SBI_FIFO_INITIALIZER
sbi_fifo.h, 419
sbi_fifo_inplace_update
sbi_fifo.c, 912
sbi_fifo.h, 422
sbi_fifo_inplace_update_types
sbi_fifo.h, 419
sbi_fifo_is_empty
sbi_fifo.c, 912
sbi_fifo.h, 423
sbi_fifo_is_full
sbi_fifo.c, 913
sbi_fifo.h, 423
sbi_fifo_reset
sbi_fifo.c, 913
SBI_FIFO_SKIP
sbi_fifo.h, 419
SBI_FIFO_UNCHANGED
sbi_fifo.h, 419
SBI_FIFO_UPDATED
sbi_fifo.h, 419
sbi_fls
sbi_bitops.h, 319
sbi_free
sbi_heap.h, 455
sbi_free_from
sbi_heap.c, 948
sbi_heap.h, 456
sbi_free_trigger
sbi_dbtr.c, 836
sbi_fwft.c
features, 924
fwft_adue_supported, 915
fwft_clear_config_lock, 916
fwft_defined_features, 924
fwft_double_trap_supported, 916
fwft_enable_lpad, 916
fwft_enable_sstack, 917
fwft_get_adue, 917
fwft_get_double_trap, 917
fwft_get_feature, 918
fwft_get_hart_state_ptr, 915
fwft_get_lpad, 918
fwft_get_misaligned_delegation, 919
fwft_get_sstack, 919
fwft_is_defined_feature, 919
fwft_lpad_supported, 919
fwft_menvcfg_read_bit, 920
fwft_menvcfg_set_bit, 920
fwft_misaligned_delegation_supported, 921

fwft_ptr_offset, 924
fwft_set_adue, 921
fwft_set_double_trap, 921
fwft_set_hart_state_ptr, 915
fwft_set_misaligned_delegation, 922
fwft_sstack_supported, 922
fwft_thishart_state_ptr, 915
get_feature_config, 922
MIS_DELEG, 915
sbi_fwft_get, 922
sbi_fwft_init, 923
sbi_fwft_set, 923
sbi_fwft.h
sbi_fwft_get, 425
sbi_fwft_init, 426
sbi_fwft_set, 426
SBI_FWFT_DOUBLE_TRAP
sbi_ecall_interface.h, 404
sbi_fwft_feature_t
sbi_ecall_interface.h, 404
sbi_fwft_get
sbi_fwft.c, 922
sbi_fwft.h, 425
SBI_FWFT_GLOBAL_FEATURE_BIT
sbi_ecall_interface.h, 397
SBI_FWFT_GLOBAL_PLATFORM_END
sbi_ecall_interface.h, 404
SBI_FWFT_GLOBAL_PLATFORM_START
sbi_ecall_interface.h, 404
SBI_FWFT_GLOBAL_RESERVED_END
sbi_ecall_interface.h, 404
SBI_FWFT_GLOBAL_RESERVED_START
sbi_ecall_interface.h, 404
sbi_fwft_init
sbi_fwft.c, 923
sbi_fwft.h, 426
SBI_FWFT_LANDING_PAD
sbi_ecall_interface.h, 404
SBI_FWFT_LOCAL_PLATFORM_END
sbi_ecall_interface.h, 404
SBI_FWFT_LOCAL_PLATFORM_START
sbi_ecall_interface.h, 404
SBI_FWFT_LOCAL_RESERVED_END
sbi_ecall_interface.h, 404
SBI_FWFT_LOCAL_RESERVED_START
sbi_ecall_interface.h, 404
SBI_FWFT_MISALIGNED_EXC_DELEG
sbi_ecall_interface.h, 404
SBI_FWFT_PLATFORM_FEATURE_BIT
sbi_ecall_interface.h, 397
SBI_FWFT_POINTER_MASKING_PMLEN
sbi_ecall_interface.h, 404
SBI_FWFT_PTE_AD_HW_UPDATING
sbi_ecall_interface.h, 404
sbi_fwft_set
sbi_fwft.c, 923
sbi_fwft.h, 426
SBI_FWFT_SET_FLAG_LOCK
sbi_ecall_interface.h, 397
SBI_FWFT_SHADOW_STACK
sbi_ecall_interface.h, 404
sbi_get_insn
sbi_unpriv.c, 1109
sbi_unpriv.h, 626
sbi_getc
sbi_console.c, 814
sbi_console.h, 329
sbi_gets
sbi_console.c, 814
sbi_console.h, 329
sbi_hart.c
__SBI_HART_EXT_DATA, 929
__attribute__, 929
__check_csr, 926
__check_csr_16, 926
__check_csr_2, 927
__check_csr_32, 927
__check_csr_4, 927
__check_csr_64, 927
__check_csr_8, 927
__check_ext_csr, 928
__check_hpm_csr, 928
__check_hpm_csr_16, 928
__check_hpm_csr_2, 928
__check_hpm_csr_4, 928
__check_hpm_csr_8, 928
__check_priv, 929
__sbi_expected_trap, 929
__sbi_expected_trap_hext, 929
__sbi_hart_update_extension, 930
__set_menvcfg_ext, 929
delegate_traps, 930
fp_init, 931
hart_detect_features, 931
hart_features_offset, 944
hart_mhpm_get_allowed_bits, 932
hart_pmp_get_allowed_addr, 932
mstatus_init, 932
sbi_hart_delegation_dump, 933
sbi_hart_expected_trap, 944
sbi_hart_ext, 945
sbi_hart_get_extensions_str, 933
sbi_hart_get_priv_version_str, 934
sbi_hart_get_smepmp_flags, 934
sbi_hart_has_extension, 935
sbi_hart_init, 936
sbi_hart_map_saddr, 937
sbi_hart_mhpm_bits, 937
sbi_hart_mhpm_mask, 938
sbi_hart_oldpmp_configure, 938
sbi_hart_pmp_addrbits, 939
sbi_hart_pmp_configure, 939
sbi_hart_pmp_count, 940
sbi_hart_pmp_log2gran, 940
sbi_hart_priv_version, 941
sbi_hart_reinit, 941

sbi_hart_smepmp_configure, 942
sbi_hart_smepmp_set, 942
sbi_hart_unmap_saddr, 943
sbi_hart_update_extension, 944
sbi_hart.h
__attribute__, 430
arg1, 439
next_addr, 439
next_mode, 440
next_virt, 440
sbi_hart_delegation_dump, 430
sbi_hart_expected_trap, 440
sbi_hart_expected_trap_addr, 431
sbi_hart_ext, 440
SBI_HART_EXT_MAX, 430
SBI_HART_EXT_SDTRIG, 429
SBI_HART_EXT_SMAIA, 429
SBI_HART_EXT_SMCDELEG, 430
SBI_HART_EXT_SMCNTRPMF, 429
SBI_HART_EXT_SMCSRIND, 429
SBI_HART_EXT_SMEPMP, 429
SBI_HART_EXT_SMNPM, 430
SBI_HART_EXT_SMSTATEEN, 429
SBI_HART_EXT_SSCCFG, 430
SBI_HART_EXT_SSCOFPMF, 429
SBI_HART_EXT_SSCSRIND, 430
SBI_HART_EXT_SSDBLTRP, 430
SBI_HART_EXT_SSTC, 429
SBI_HART_EXT_SVADE, 430
SBI_HART_EXT_SVADU, 430
SBI_HART_EXT_SVPBMT, 429
SBI_HART_EXT_XANDESPMU, 429
SBI_HART_EXT_ZICBOM, 429
SBI_HART_EXT_ZICBOZ, 429
SBI_HART_EXT_ZICFILP, 430
SBI_HART_EXT_ZICFISS, 430
SBI_HART_EXT_ZICNTR, 429
SBI_HART_EXT_ZIHPM, 429
SBI_HART_EXT_ZKR, 429
sbi_hart_extensions, 429
sbi_hart_get_extensions_str, 431
sbi_hart_get_priv_version_str, 432
sbi_hart_has_extension, 432
sbi_hart_init, 433
sbi_hart_map_saddr, 434
sbi_hart_mhpm_bits, 435
sbi_hart_mhpm_mask, 435
sbi_hart_pmp_addrbits, 435
sbi_hart_pmp_configure, 436
sbi_hart_pmp_count, 436
sbi_hart_pmp_log2gran, 437
SBI_HART_PRIV_VER_1_10, 430
SBI_HART_PRIV_VER_1_11, 430
SBI_HART_PRIV_VER_1_12, 430
SBI_HART_PRIV_VER_UNKNOWN, 430
sbi_hart_priv_version, 437
sbi_hart_priv_versions, 430
sbi_hart_reinit, 438
sbi_hart_unmap_saddr, 438
sbi_hart_update_extension, 439
SBI_SMEPMP_RESV_ENTRY, 429
sbi_hart_delegation_dump
sbi_hart.c, 933
sbi_hart.h, 430
sbi_hart_expected_trap
sbi_hart.c, 944
sbi_hart.h, 440
sbi_hart_expected_trap_addr
sbi_hart.h, 431
sbi_hart_ext
sbi_hart.c, 945
sbi_hart.h, 440
sbi_hart_ext_data, 133
id, 134
name, 134
SBI_HART_EXT_MAX
sbi_hart.h, 430
SBI_HART_EXT_SDTRIG
sbi_hart.h, 429
SBI_HART_EXT_SMAIA
sbi_hart.h, 429
SBI_HART_EXT_SMCDELEG
sbi_hart.h, 430
SBI_HART_EXT_SMCNTRPMF
sbi_hart.h, 429
SBI_HART_EXT_SMCSRIND
sbi_hart.h, 429
SBI_HART_EXT_SMEPMP
sbi_hart.h, 429
SBI_HART_EXT_SMNPM
sbi_hart.h, 430
SBI_HART_EXT_SMSTATEEN
sbi_hart.h, 429
SBI_HART_EXT_SSCCFG
sbi_hart.h, 430
SBI_HART_EXT_SSCOFPMF
sbi_hart.h, 429
SBI_HART_EXT_SSCSRIND
sbi_hart.h, 430
SBI_HART_EXT_SSDBLTRP
sbi_hart.h, 430
SBI_HART_EXT_SSTC
sbi_hart.h, 429
SBI_HART_EXT_SVADE
sbi_hart.h, 430
SBI_HART_EXT_SVADU
sbi_hart.h, 430
SBI_HART_EXT_SVPBMT
sbi_hart.h, 429
SBI_HART_EXT_XANDESPMU
sbi_hart.h, 429
SBI_HART_EXT_ZICBOM
sbi_hart.h, 429
SBI_HART_EXT_ZICBOZ
sbi_hart.h, 429
SBI_HART_EXT_ZICFILP

sbi_hart.h, 430
SBI_HART_EXT_ZICFISS
sbi_hart.h, 430
SBI_HART_EXT_ZICNTR
sbi_hart.h, 429
SBI_HART_EXT_ZIHPM
sbi_hart.h, 429
SBI_HART_EXT_ZKR
sbi_hart.h, 429
sbi_hart_extensions
sbi_hart.h, 429
sbi_hart_features, 134
detected, 134
extensions, 134
mhpm_bits, 134
mhpm_mask, 134
pmp_addr_bits, 134
pmp_count, 134
pmp_log2gran, 134
priv_version, 135
sbi_hart_get_extensions_str
sbi_hart.c, 933
sbi_hart.h, 431
sbi_hart_get_priv_version_str
sbi_hart.c, 934
sbi_hart.h, 432
sbi_hart_get_smepmp_flags
sbi_hart.c, 934
sbi_hart_has_extension
sbi_hart.c, 935
sbi_hart.h, 432
sbi_hart_init
sbi_hart.c, 936
sbi_hart.h, 433
sbi_hart_map_saddr
sbi_hart.c, 937
sbi_hart.h, 434
sbi_hart_mhpm_bits
sbi_hart.c, 937
sbi_hart.h, 435
sbi_hart_mhpm_mask
sbi_hart.c, 938
sbi_hart.h, 435
sbi_hart_oldpmp_configure
sbi_hart.c, 938
sbi_hart_pmp_addrbits
sbi_hart.c, 939
sbi_hart.h, 435
sbi_hart_pmp_configure
sbi_hart.c, 939
sbi_hart.h, 436
sbi_hart_pmp_count
sbi_hart.c, 940
sbi_hart.h, 436
sbi_hart_pmp_log2gran
sbi_hart.c, 940
sbi_hart.h, 437
SBI_HART_PRIV_VER_1_10
sbi_hart.h, 430
SBI_HART_PRIV_VER_1_11
sbi_hart.h, 430
SBI_HART_PRIV_VER_1_12
sbi_hart.h, 430
SBI_HART_PRIV_VER_UNKNOWN
sbi_hart.h, 430
sbi_hart_priv_version
sbi_hart.c, 941
sbi_hart.h, 437
sbi_hart_priv_versions
sbi_hart.h, 430
sbi_hart_reinit
sbi_hart.c, 941
sbi_hart.h, 438
sbi_hart_smepmp_configure
sbi_hart.c, 942
sbi_hart_smepmp_set
sbi_hart.c, 942
sbi_hart_unmap_saddr
sbi_hart.c, 943
sbi_hart.h, 438
sbi_hart_update_extension
sbi_hart.c, 944
sbi_hart.h, 439
sbi_hartid_to_hartindex
sbi_scratch.c, 1029
sbi_scratch.h, 560
sbi_hartid_to_scratch
sbi_scratch.h, 556
sbi_hartid_valid
sbi_scratch.h, 556
sbi_hartindex_to_domain
sbi_domain.c, 854
sbi_domain.h, 366
sbi_hartindex_to_hartid
sbi_scratch.h, 556
sbi_hartindex_to_scratch
sbi_scratch.h, 557
sbi_hartindex_valid
sbi_scratch.h, 557
sbi_hartmask, 135
DECLARE_BITMAP, 135
sbi_hartmask.h
sbi_hartmask_and, 444
sbi_hartmask_bits, 443
sbi_hartmask_clear_all, 444
sbi_hartmask_clear_hartid, 445
sbi_hartmask_clear_hartindex, 445
sbi_hartmask_copy, 446
sbi_hartmask_for_each_hartindex, 443
SBI_HARTMASK_INIT, 443
SBI_HARTMASK_INIT_EXCEPT, 443
SBI_HARTMASK_MAX_BITS, 443
sbi_hartmask_or, 447
sbi_hartmask_set_all, 447
sbi_hartmask_set_hartid, 448
sbi_hartmask_set_hartindex, 448

sbi_hartmask_test_hartid, 449
sbi_hartmask_test_hartindex, 449
sbi_hartmask_xor, 450
sbi_hartmask_and
sbi_hartmask.h, 444
sbi_hartmask_bits
sbi_hartmask.h, 443
sbi_hartmask_clear_all
sbi_hartmask.h, 444
sbi_hartmask_clear_hartid
sbi_hartmask.h, 445
sbi_hartmask_clear_hartindex
sbi_hartmask.h, 445
sbi_hartmask_copy
sbi_hartmask.h, 446
sbi_hartmask_for_each_hartindex
sbi_hartmask.h, 443
SBI_HARTMASK_INIT
sbi_hartmask.h, 443
SBI_HARTMASK_INIT_EXCEPT
sbi_hartmask.h, 443
SBI_HARTMASK_MAX_BITS
sbi_hartmask.h, 443
sbi_hartmask_or
sbi_hartmask.h, 447
sbi_hartmask_set_all
sbi_hartmask.h, 447
sbi_hartmask_set_hartid
sbi_hartmask.h, 448
sbi_hartmask_set_hartindex
sbi_hartmask.h, 448
sbi_hartmask_test_hartid
sbi_hartmask.h, 449
sbi_hartmask_test_hartindex
sbi_hartmask.h, 449
sbi_hartmask_xor
sbi_hartmask.h, 450
sbi_heap.c
alloc_with_align, 946
global_hpctrl, 953
HEAP_ALLOC_ALIGN, 946
HEAP_HOUSEKEEPING_FACTOR, 946
sbi_aligned_alloc_from, 947
sbi_free_from, 948
sbi_heap_alloc_new, 949
sbi_heap_free_space_from, 949
sbi_heap_init, 949
sbi_heap_init_new, 950
sbi_heap_reserved_space_from, 950
sbi_heap_used_space_from, 950
sbi_malloc_from, 951
sbi_zalloc_from, 952
sbi_heap.h
global_hpctrl, 464
HEAP_BASE_ALIGN, 453
sbi_aligned_alloc, 453
sbi_aligned_alloc_from, 453
sbi_calloc, 454
sbi_calloc_from, 455
sbi_free, 455
sbi_free_from, 456
sbi_heap_alloc_new, 457
sbi_heap_free_space, 457
sbi_heap_free_space_from, 458
sbi_heap_init, 458
sbi_heap_init_new, 459
sbi_heap_reserved_space, 459
sbi_heap_reserved_space_from, 460
sbi_heap_used_space, 460
sbi_heap_used_space_from, 460
sbi_malloc, 461
sbi_malloc_from, 461
sbi_zalloc, 462
sbi_zalloc_from, 463
sbi_heap_alloc_new
sbi_heap.c, 949
sbi_heap.h, 457
sbi_heap_free_space
sbi_heap.h, 457
sbi_heap_free_space_from
sbi_heap.c, 949
sbi_heap.h, 458
sbi_heap_init
sbi_heap.c, 949
sbi_heap.h, 458
sbi_heap_init_new
sbi_heap.c, 950
sbi_heap.h, 459
sbi_heap_reserved_space
sbi_heap.h, 459
sbi_heap_reserved_space_from
sbi_heap.c, 950
sbi_heap.h, 460
sbi_heap_used_space
sbi_heap.h, 460
sbi_heap_used_space_from
sbi_heap.c, 950
sbi_heap.h, 460
sbi_hfence.h
__sbi_hfence_gvma_all, 466
__sbi_hfence_gvma_gpa, 466
__sbi_hfence_gvma_vmid, 466
__sbi_hfence_gvma_vmid_gpa, 467
__sbi_hfence_vvma_all, 467
__sbi_hfence_vvma_asid, 467
__sbi_hfence_vvma_asid_va, 467
__sbi_hfence_vvma_va, 468
sbi_hsm.c
__sbi_hsm_hart_change_state, 955
__sbi_hsm_hart_get_state, 955
__sbi_hsm_suspend_default, 955
__sbi_hsm_suspend_non_ret_restore, 956
__sbi_hsm_suspend_non_ret_save, 956
hart_data_offset, 966
hsm_dev, 966
hsm_device_hart_resume, 956

hsm_device_hart_start, 957
hsm_device_hart_stop, 957
hsm_device_hart_suspend, 957
hsm_device_has_hart_hotplug, 957
hsm_device_has_hart_secondary_boot, 958
hsm_start_ticket_acquire, 958
hsm_start_ticket_release, 958
sbi_hsm_exit, 959
sbi_hsm_get_device, 959
sbi_hsm_hart_change_state, 960
sbi_hsm_hart_get_state, 960
sbi_hsm_hart_interruptible_mask, 960
sbi_hsm_hart_resume_finish, 961
sbi_hsm_hart_resume_start, 962
sbi_hsm_hart_start, 962
sbi_hsm_hart_start_finish, 963
sbi_hsm_hart_stop, 964
sbi_hsm_hart_suspend, 964
sbi_hsm_hart_wait, 965
sbi_hsm_init, 965
sbi_hsm_set_device, 966
sbi_hsm.h
__sbi_hsm_hart_get_state, 470
__sbi_hsm_suspend_non_ret_save, 470
sbi_hsm_exit, 471
sbi_hsm_get_device, 471
sbi_hsm_hart_change_state, 471
sbi_hsm_hart_get_state, 472
sbi_hsm_hart_interruptible_mask, 472
sbi_hsm_hart_resume_finish, 473
sbi_hsm_hart_resume_start, 473
sbi_hsm_hart_start, 474
sbi_hsm_hart_start_finish, 475
sbi_hsm_hart_stop, 476
sbi_hsm_hart_suspend, 476
sbi_hsm_init, 477
sbi_hsm_set_device, 478
sbi_hsm_device, 135
hart_resume, 135
hart_start, 135
hart_stop, 136
hart_suspend, 136
name, 136
sbi_hsm_exit
sbi_hsm.c, 959
sbi_hsm.h, 471
sbi_hsm_get_device
sbi_hsm.c, 959
sbi_hsm.h, 471
sbi_hsm_hart_change_state
sbi_hsm.c, 960
sbi_hsm.h, 471
sbi_hsm_hart_get_state
sbi_hsm.c, 960
sbi_hsm.h, 472
sbi_hsm_hart_interruptible_mask
sbi_hsm.c, 960
sbi_hsm.h, 472
sbi_hsm_hart_resume_finish
sbi_hsm.c, 961
sbi_hsm.h, 473
sbi_hsm_hart_resume_start
sbi_hsm.c, 962
sbi_hsm.h, 473
sbi_hsm_hart_start
sbi_hsm.c, 962
sbi_hsm.h, 474
sbi_hsm_hart_start_finish
sbi_hsm.c, 963
sbi_hsm.h, 475
sbi_hsm_hart_stop
sbi_hsm.c, 964
sbi_hsm.h, 476
sbi_hsm_hart_suspend
sbi_hsm.c, 964
sbi_hsm.h, 476
sbi_hsm_hart_wait
sbi_hsm.c, 965
sbi_hsm_init
sbi_hsm.c, 965
sbi_hsm.h, 477
sbi_hsm_rpmi
fdt_hsm_rpmi.c, 1198
sbi_hsm_set_device
sbi_hsm.c, 966
sbi_hsm.h, 478
SBI_HSM_STATE_RESUME_PENDING
sbi_ecall_interface.h, 397
SBI_HSM_STATE_START_PENDING
sbi_ecall_interface.h, 397
SBI_HSM_STATE_STARTED
sbi_ecall_interface.h, 397
SBI_HSM_STATE_STOP_PENDING
sbi_ecall_interface.h, 397
SBI_HSM_STATE_STOPPED
sbi_ecall_interface.h, 397
SBI_HSM_STATE_SUSPEND_PENDING
sbi_ecall_interface.h, 397
SBI_HSM_STATE_SUSPENDED
sbi_ecall_interface.h, 397
SBI_HSM_SUSP_BASE_MASK
sbi_ecall_interface.h, 398
SBI_HSM_SUSP_NON_RET_BIT
sbi_ecall_interface.h, 398
SBI_HSM_SUSP_PLAT_BASE
sbi_ecall_interface.h, 398
SBI_HSM_SUSPEND_NON_RET_DEFAULT
sbi_ecall_interface.h, 398
SBI_HSM_SUSPEND_NON_RET_LAST
sbi_ecall_interface.h, 398
SBI_HSM_SUSPEND_NON_RET_PLATFORM
sbi_ecall_interface.h, 398
SBI_HSM_SUSPEND_RET_DEFAULT
sbi_ecall_interface.h, 398
SBI_HSM_SUSPEND_RET_LAST
sbi_ecall_interface.h, 398

SBI_HSM_SUSPEND_RET_PLATFORM
sbi_ecall_interface.h, 398
sbi_illegal_insn.c
illegal_insn_func, 967
illegal_insn_table, 969
misc_mem_opcode_insn, 968
sbi_illegal_insn_handler, 968
system_opcode_insn, 968
truly_illegal_insn, 969
sbi_illegal_insn.h
sbi_illegal_insn_handler, 480
sbi_illegal_insn_handler
sbi_illegal_insn.c, 968
sbi_illegal_insn.h, 480
sbi_init
sbi_init.c, 979
sbi_init.h, 482
sbi_init.c
BANNER, 971
coldboot_done, 981
coldboot_lottery, 981
entry_count_offset, 981
init_coldboot, 971
init_count_offset, 981
init_warm_resume, 973
init_warm_startup, 973
init_warmboot, 975
sbi_boot_print_banner, 976
sbi_boot_print_domains, 976
sbi_boot_print_general, 977
sbi_boot_print_hart, 977
sbi_entry_count, 978
sbi_exit, 978
sbi_init, 979
sbi_init_count, 980
wait_for_coldboot, 980
wake_coldboot_harts, 981
sbi_init.h
sbi_entry_count, 481
sbi_exit, 481
sbi_init, 482
sbi_init_count, 484
sbi_init_count
sbi_init.c, 980
sbi_init.h, 484
SBI_INIT_LIST_HEAD
sbi_list.h, 498
sbi_ipi.c
ipi_data_off, 990
ipi_dev, 990
ipi_halt_event, 990
ipi_halt_ops, 990
ipi_ops_array, 990
ipi_smode_event, 990
ipi_smode_ops, 990
sbi_ipi_clear_smode, 983
sbi_ipi_event_create, 983
sbi_ipi_event_destroy, 983
sbi_ipi_exit, 983
sbi_ipi_get_device, 984
sbi_ipi_init, 984
sbi_ipi_process, 985
sbi_ipi_process_halt, 985
sbi_ipi_process_smode, 986
sbi_ipi_raw_clear, 986
sbi_ipi_raw_send, 986
sbi_ipi_send, 986
sbi_ipi_send_halt, 987
sbi_ipi_send_many, 988
sbi_ipi_send_smode, 988
sbi_ipi_set_device, 989
sbi_ipi_sync, 989
sbi_ipi_update_halt, 990
sbi_ipi.h
sbi_ipi_clear_smode, 486
sbi_ipi_event_create, 486
sbi_ipi_event_destroy, 487
SBI_IPI_EVENT_MAX, 486
sbi_ipi_exit, 487
sbi_ipi_get_device, 487
sbi_ipi_init, 488
sbi_ipi_process, 488
sbi_ipi_raw_clear, 489
sbi_ipi_raw_send, 489
sbi_ipi_send_halt, 490
sbi_ipi_send_many, 490
sbi_ipi_send_smode, 491
sbi_ipi_set_device, 492
SBI_IPI_UPDATE_BREAK, 486
SBI_IPI_UPDATE_RETRY, 486
SBI_IPI_UPDATE_SUCCESS, 486
sbi_ipi_update_type, 486
sbi_ipi_clear_smode
sbi_ipi.c, 983
sbi_ipi.h, 486
sbi_ipi_device, 136
ipi_clear, 136
ipi_send, 136
name, 136
sbi_ipi_event_create
sbi_ipi.c, 983
sbi_ipi.h, 486
sbi_ipi_event_destroy
sbi_ipi.c, 983
sbi_ipi.h, 487
SBI_IPI_EVENT_MAX
sbi_ipi.h, 486
sbi_ipi_event_ops, 137
name, 137
process, 137
sync, 137
update, 137
sbi_ipi_exit
sbi_ipi.c, 983
sbi_ipi.h, 487
sbi_ipi_get_device

sbi_ipi.c, 984
sbi_ipi.h, 487
sbi_ipi_init
sbi_ipi.c, 984
sbi_ipi.h, 488
sbi_ipi_process
sbi_ipi.c, 985
sbi_ipi.h, 488
sbi_ipi_process_halt
sbi_ipi.c, 985
sbi_ipi_process_smode
sbi_ipi.c, 986
sbi_ipi_raw_clear
sbi_ipi.c, 986
sbi_ipi.h, 489
sbi_ipi_raw_send
sbi_ipi.c, 986
sbi_ipi.h, 489
sbi_ipi_send
sbi_ipi.c, 986
sbi_ipi_send_halt
sbi_ipi.c, 987
sbi_ipi.h, 490
sbi_ipi_send_many
sbi_ipi.c, 988
sbi_ipi.h, 490
sbi_ipi_send_smode
sbi_ipi.c, 988
sbi_ipi.h, 491
sbi_ipi_set_device
sbi_ipi.c, 989
sbi_ipi.h, 492
sbi_ipi_sync
sbi_ipi.c, 989
SBI_IPI_UPDATE_BREAK
sbi_ipi.h, 486
sbi_ipi_update_halt
sbi_ipi.c, 990
SBI_IPI_UPDATE_RETRY
sbi_ipi.h, 486
SBI_IPI_UPDATE_SUCCESS
sbi_ipi.h, 486
sbi_ipi_update_type
sbi_ipi.h, 486
sbi_irqchip.c
default_irqfn, 991
ext_irqfn, 993
sbi_irqchip_add_device, 992
sbi_irqchip_exit, 992
sbi_irqchip_init, 993
SBI_LIST_HEAD, 993
sbi_irqchip.h
sbi_irqchip_add_device, 494
sbi_irqchip_exit, 494
sbi_irqchip_init, 495
sbi_irqchip_process, 495
sbi_irqchip_add_device
sbi_irqchip.c, 992
sbi_irqchip.h, 494
sbi_irqchip_device, 138
irq_handle, 138
node, 138
warm_init, 138
sbi_irqchip_exit
sbi_irqchip.c, 992
sbi_irqchip.h, 494
sbi_irqchip_init
sbi_irqchip.c, 993
sbi_irqchip.h, 495
sbi_irqchip_process
sbi_irqchip.h, 495
sbi_isprintable
sbi_console.c, 815
sbi_console.h, 330
SBI_LAST_ERR
sbi_ecall_interface.h, 398
sbi_ld_access_emulator
sbi_trap_ldst.c, 1101
sbi_ldst_data, 138
data_bytes, 138
data_u32, 138
data_u64, 138
data_ulong, 138
sbi_list.h
__sbi_list_add, 499
__sbi_list_del, 500
__sbi_list_del_entry, 501
SBI_INIT_LIST_HEAD, 498
sbi_list_add, 501
sbi_list_add_tail, 503
sbi_list_del, 505
sbi_list_del_init, 506
sbi_list_empty, 507
sbi_list_entry, 498
sbi_list_first_entry, 498
sbi_list_for_each, 498
sbi_list_for_each_entry, 498
SBI_LIST_HEAD, 499
SBI_LIST_HEAD_INIT, 499
sbi_list_last_entry, 499
SBI_LIST_POISON_NEXT, 499
SBI_LIST_POISON_PREV, 499
sbi_list_add
sbi_list.h, 501
sbi_list_add_tail
sbi_list.h, 503
sbi_list_del
sbi_list.h, 505
sbi_list_del_init
sbi_list.h, 506
sbi_list_empty
sbi_list.h, 507
sbi_list_entry
sbi_list.h, 498
sbi_list_first_entry
sbi_list.h, 498

sbi_list_for_each
sbi_list.h, 498
sbi_list_for_each_entry
sbi_list.h, 498
SBI_LIST_HEAD
fdt_fixup.c, 1154
fdt_timer_mtimer.c, 1605
gpio.c, 1191
i2c.c, 1220
mailbox.c, 1479
regmap.c, 1505
sbi_domain.c, 855
sbi_domain_data.c, 866
sbi_ecall.c, 872
sbi_irqchip.c, 993
sbi_list.h, 499
sbi_mpxy.c, 1001
sbi_system.c, 1069
SBI_LIST_HEAD_INIT
sbi_list.h, 499
sbi_list_last_entry
sbi_list.h, 499
SBI_LIST_POISON_NEXT
sbi_list.h, 499
SBI_LIST_POISON_PREV
sbi_list.h, 499
sbi_load_access_handler
sbi_trap_ldst.c, 1101
sbi_trap_ldst.h, 614
sbi_load_hart_mask_unpriv
sbi_ecall_legacy.c, 886
sbi_malloc
sbi_heap.h, 461
sbi_malloc_from
sbi_heap.c, 951
sbi_heap.h, 461
sbi_math.c
log2roundup, 994
sbi_math.h
log2roundup, 509
sbi_math_test.c
log2roundup_test, 1125
math_test_cases, 1126
SBIUNIT_TEST_SUITE, 1125
sbi_memchr
sbi_string.c, 1065
sbi_string.h, 577
sbi_memcmp
sbi_string.c, 1065
sbi_string.h, 577
sbi_memcpy
sbi_string.c, 1065
sbi_string.h, 577
sbi_memmove
sbi_string.c, 1065
sbi_string.h, 578
sbi_memset
sbi_string.c, 1066
sbi_string.h, 578
sbi_misaligned_ld_emulator
sbi_trap_ldst.c, 1102
sbi_misaligned_load_handler
sbi_trap_ldst.c, 1102
sbi_trap_ldst.h, 615
sbi_misaligned_st_emulator
sbi_trap_ldst.c, 1103
sbi_misaligned_store_handler
sbi_trap_ldst.c, 1103
sbi_trap_ldst.h, 615
sbi_misaligned_tinst_fixup
sbi_trap_ldst.c, 1104
sbi_trap_ldst.h, 616
sbi_misaligned_v_ld_emulator
sbi_trap_ldst.h, 616
sbi_trap_v_ldst.c, 1107
sbi_misaligned_v_st_emulator
sbi_trap_ldst.h, 616
sbi_trap_v_ldst.c, 1107
sbi_mpxy.c
assert_field_offset, 996, 998, 999
ATTR_SIZE, 996
CAP_DISABLE, 996
CAP_ENABLE, 996
CAP_EVENTSSTATE_MASK, 996
CAP_EVENTSSTATE_POS, 996
CAP_GET, 997
CAP_GET_NOTIFICATIONS_MASK, 997
CAP_GET_NOTIFICATIONS_POS, 997
CAP_MSI_MASK, 997
CAP_MSI_POS, 997
CAP_SEND_MSG_WITH_RESP_MASK, 997
CAP_SEND_MSG_WITH_RESP_POS, 997
CAP_SEND_MSG_WITHOUT_RESP_MASK, 997
CAP_SEND_MSG_WITHOUT_RESP_POS, 997
CAP_SSE_MASK, 997
CAP_SSE_POS, 997
hart_shmem_base, 999
INVALID_ADDR, 997
mpxy_check_write_std_attr, 999
mpxy_copy_std_attrs, 999
mpxy_find_channel, 1000
mpxy_is_std_attr, 1000
mpxy_shmem_enabled, 1000
mpxy_state_offset, 1007
mpxy_std_attrs_init, 1001
mpxy_write_std_attr, 1001
SBI_LIST_HEAD, 1001
sbi_mpxy_channel_available, 1001
sbi_mpxy_get_channel_ids, 1002
sbi_mpxy_get_notification_events, 1002
sbi_mpxy_init, 1003
sbi_mpxy_read_attrs, 1003
sbi_mpxy_register_channel, 1004
sbi_mpxy_send_message, 1005
sbi_mpxy_set_shmem, 1005
sbi_mpxy_shmem_disable, 1006

sbi_mpxy_write_attrs, 1006
sbi_mpxy.h
SBI_EXT_MPXY_SHMEM_FLAG_MAX_IDX, 512
SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE, 512
SBI_EXT_MPXY_SHMEM_FLAG_OVERWRITE_RETURN, 512
SBI_EXT_MPXY_SHMEM_FLAGS, 512
SBI_MPXY_ATTR_CHANNEL_CAPABILITY, 512
SBI_MPXY_ATTR_EVENTS_STATE_CONTROL, 513
sbi_mpxy_attr_id, 512
SBI_MPXY_ATTR_MSG_COMPLETION_TIMEOUT, 512
SBI_MPXY_ATTR_MSG_MAX_LEN, 512
SBI_MPXY_ATTR_MSG_PROT_ID, 512
SBI_MPXY_ATTR_MSG_PROT_VER, 512
SBI_MPXY_ATTR_MSG_SEND_TIMEOUT, 512
SBI_MPXY_ATTR_MSGPROTO_ATTR_END, 513
SBI_MPXY_ATTR_MSGPROTO_ATTR_START, 513
SBI_MPXY_ATTR_MSI_ADDR_HI, 513
SBI_MPXY_ATTR_MSI_ADDR_LO, 513
SBI_MPXY_ATTR_MSI_CONTROL, 512
SBI_MPXY_ATTR_MSI_DATA, 513
SBI_MPXY_ATTR_SSE_EVENT_ID, 512
SBI_MPXY_ATTR_STD_ATTR_MAX_IDX, 513
sbi_mpxy_channel_available, 513
sbi_mpxy_get_channel_ids, 513
sbi_mpxy_get_notification_events, 514
sbi_mpxy_init, 514
sbi_mpxy_msgproto_id, 513
SBI_MPXY_MSGPROTO_MAX_IDX, 513
SBI_MPXY_MSGPROTO_RPMI_ID, 513
SBI_MPXY_MSGPROTO_VENDOR_END, 513
SBI_MPXY_MSGPROTO_VENDOR_START, 513
SBI_MPXY_MSGPROTO_VERSION, 512
sbi_mpxy_read_attrs, 515
sbi_mpxy_register_channel, 516
sbi_mpxy_send_message, 516
sbi_mpxy_set_shmem, 517
sbi_mpxy_write_attrs, 517
SBI_MPXY_ATTR_CHANNEL_CAPABILITY
sbi_mpxy.h, 512
SBI_MPXY_ATTR_EVENTS_STATE_CONTROL
sbi_mpxy.h, 513
sbi_mpxy_attr_id
sbi_mpxy.h, 512
SBI_MPXY_ATTR_MSG_COMPLETION_TIMEOUT
sbi_mpxy.h, 512
SBI_MPXY_ATTR_MSG_MAX_LEN
sbi_mpxy.h, 512
SBI_MPXY_ATTR_MSG_PROT_ID
sbi_mpxy.h, 512
SBI_MPXY_ATTR_MSG_PROT_VER
sbi_mpxy.h, 512
SBI_MPXY_ATTR_MSG_SEND_TIMEOUT
sbi_mpxy.h, 512
SBI_MPXY_ATTR_MSGPROTO_ATTR_END
sbi_mpxy.h, 513
SBI_MPXY_ATTR_MSGPROTO_ATTR_START
sbi_mpxy.h, 513
SBI_MPXY_ATTR_MSI_ADDR_HI
sbi_mpxy.h, 513
SBI_MPXY_ATTR_MSI_ADDR_LO
sbi_mpxy.h, 513
SBI_MPXY_ATTR_MSI_CONTROL
sbi_mpxy.h, 512
SBI_MPXY_ATTR_MSI_DATA
sbi_mpxy.h, 513
SBI_MPXY_ATTR_SSE_EVENT_ID
sbi_mpxy.h, 512
SBI_MPXY_ATTR_STD_ATTR_MAX_IDX
sbi_mpxy.h, 513
sbi_mpxy_channel, 139
attrs, 139
channel_id, 139
get_notification_events, 139
head, 139
read_attributes, 139
send_message_with_response, 139
send_message_without_response, 140
switch_eventsstate, 140
write_attributes, 140
sbi_mpxy_channel_attrs, 140
capability, 140
eventsstate_ctrl, 140
msg_completion_timeout, 140
msg_data_maxlen, 141
msg_proto_id, 141
msg_proto_version, 141
msg_send_timeout, 141
msi_control, 141
msi_info, 141
sse_event_id, 141
sbi_mpxy_channel_available
sbi_mpxy.c, 1001
sbi_mpxy.h, 513
sbi_mpxy_get_channel_ids
sbi_mpxy.c, 1002
sbi_mpxy.h, 513
sbi_mpxy_get_notification_events
sbi_mpxy.c, 1002
sbi_mpxy.h, 514
sbi_mpxy_init
sbi_mpxy.c, 1003
sbi_mpxy.h, 514
sbi_mpxy_msgproto_id
sbi_mpxy.h, 513
SBI_MPXY_MSGPROTO_MAX_IDX
sbi_mpxy.h, 513
SBI_MPXY_MSGPROTO_RPMI_ID
sbi_mpxy.h, 513
SBI_MPXY_MSGPROTO_VENDOR_END
sbi_mpxy.h, 513
SBI_MPXY_MSGPROTO_VENDOR_START

sbi_mpxy.h, 513
SBI_MPXY_MSGPROTO_VERSION
sbi_mpxy.h, 512
sbi_mpxy_msi_info, 141
msi_addr_hi, 141
msi_addr_lo, 141
msi_data, 141
sbi_mpxy_read_attrs
sbi_mpxy.c, 1003
sbi_mpxy.h, 515
sbi_mpxy_register_channel
sbi_mpxy.c, 1004
sbi_mpxy.h, 516
sbi_mpxy_send_message
sbi_mpxy.c, 1005
sbi_mpxy.h, 516
sbi_mpxy_set_shmem
sbi_mpxy.c, 1005
sbi_mpxy.h, 517
sbi_mpxy_shmem_disable
sbi_mpxy.c, 1006
sbi_mpxy_write_attrs
sbi_mpxy.c, 1006
sbi_mpxy.h, 517
sbi_mstatus_prev_mode
sbi_trap.h, 607
sbi_ngets
sbi_console.c, 815
sbi_console.h, 330
sbi_nputs
sbi_console.c, 815
sbi_console.h, 330
SBI_OK
sbi_error.h, 417
SBI_OPENSBI_IMPID
sbi_ecall.h, 379
sbi_panic
sbi_console.c, 816
sbi_platform, 142
features, 142
firmware_context, 142
hart_count, 142
hart_index2id, 142
hart_stack_size, 142
heap_size, 142
name, 143
opensbi_version, 143
platform_ops_addr, 143
platform_version, 143
reserved, 143
sbi_platform.c
sbi_platform_feature_id2string, 1008
sbi_platform_get_features_str, 1008
sbi_platform.h
sbi_platform_cold_boot_allowed, 524
SBI_PLATFORM_DEFAULT_FEATURES, 522
SBI_PLATFORM_DEFAULT_HART_STACK_SIZE, 522
SBI_PLATFORM_DEFAULT_HEAP_SIZE, 522
sbi_platform_domains_init, 524
sbi_platform_early_exit, 525
sbi_platform_early_init, 525
sbi_platform_emulate_load, 526
sbi_platform_emulate_store, 526
sbi_platform_extensions_init, 527
sbi_platform_features, 524
SBI_PLATFORM_FEATURES_OFFSET, 522
sbi_platform_final_exit, 527
sbi_platform_final_init, 528
SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET, 522
sbi_platform_get_features, 528
sbi_platform_get_features_str, 529
sbi_platform_hart_count, 529
SBI_PLATFORM_HART_COUNT_OFFSET, 522
SBI_PLATFORM_HART_INDEX2ID_OFFSET, 522
sbi_platform_hart_stack_size, 530
SBI_PLATFORM_HART_STACK_SIZE_OFFSET, 522
SBI_PLATFORM_HAS_LAST_FEATURE, 524
SBI_PLATFORM_HAS_MFAULTS_DELEGATION, 524
sbi_platform_has_mfaults_delegation, 522
SBI_PLATFORM_HEAP_SIZE_OFFSET, 522
sbi_platform_ipi_init, 530
sbi_platform_irqchip_init, 530
sbi_platform_misa_extension, 532
sbi_platform_misa_xlen, 532
sbi_platform_mpxy_init, 533
sbi_platform_name, 533
SBI_PLATFORM_NAME_OFFSET, 522
sbi_platform_nascent_init, 533
SBI_PLATFORM_OPENSBI_VERSION_OFFSET, 523
sbi_platform_ops, 523
SBI_PLATFORM_OPS_OFFSET, 523
sbi_platform_pmu_init, 534
sbi_platform_pmu_xlate_to_mhpmevent, 534
sbi_platform_ptr, 523
SBI_PLATFORM_RESERVED_OFFSET, 523
sbi_platform_thishart_ptr, 523
sbi_platform_timer_init, 536
sbi_platform_tlb_fifo_num_entries, 536
SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT, 523
sbi_platform_tlbr_flush_limit, 537
sbi_platform_vendor_ext_check, 537
sbi_platform_vendor_ext_provider, 538
SBI_PLATFORM_VERSION, 523
SBI_PLATFORM_VERSION_OFFSET, 523
sbi_platform_cold_boot_allowed
sbi_platform.h, 524
SBI_PLATFORM_DEFAULT_FEATURES
sbi_platform.h, 522
SBI_PLATFORM_DEFAULT_HART_STACK_SIZE

sbi_platform.h, 522
SBI_PLATFORM_DEFAULT_HEAP_SIZE
sbi_platform.h, 522
sbi_platform_domains_init
sbi_platform.h, 524
sbi_platform_early_exit
sbi_platform.h, 525
sbi_platform_early_init
sbi_platform.h, 525
sbi_platform_emulate_load
sbi_platform.h, 526
sbi_platform_emulate_store
sbi_platform.h, 526
sbi_platform_extensions_init
sbi_platform.h, 527
sbi_platform_feature_id2string
sbi_platform.c, 1008
sbi_platform_features
sbi_platform.h, 524
SBI_PLATFORM_FEATURES_OFFSET
sbi_platform.h, 522
sbi_platform_final_exit
sbi_platform.h, 527
sbi_platform_final_init
sbi_platform.h, 528
SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET
sbi_platform.h, 522
sbi_platform_get_features
sbi_platform.h, 528
sbi_platform_get_features_str
sbi_platform.c, 1008
sbi_platform.h, 529
sbi_platform_hart_count
sbi_platform.h, 529
SBI_PLATFORM_HART_COUNT_OFFSET
sbi_platform.h, 522
SBI_PLATFORM_HART_INDEX2ID_OFFSET
sbi_platform.h, 522
sbi_platform_hart_stack_size
sbi_platform.h, 530
SBI_PLATFORM_HART_STACK_SIZE_OFFSET
sbi_platform.h, 522
SBI_PLATFORM_HAS_LAST_FEATURE
sbi_platform.h, 524
SBI_PLATFORM_HAS_MFAULTS_DELEGATION
sbi_platform.h, 524
sbi_platform_has_mfaults_delegation
sbi_platform.h, 522
SBI_PLATFORM_HEAP_SIZE_OFFSET
sbi_platform.h, 522
sbi_platform_ipi_init
sbi_platform.h, 530
sbi_platform_irqchip_init
sbi_platform.h, 530
sbi_platform_misa_extension
sbi_platform.h, 532
sbi_platform_misa_xlen
sbi_platform.h, 532
sbi_platform_mpxy_init
sbi_platform.h, 533
sbi_platform_name
sbi_platform.h, 533
SBI_PLATFORM_NAME_OFFSET
sbi_platform.h, 522
sbi_platform_nascent_init
sbi_platform.h, 533
SBI_PLATFORM_OPENSBI_VERSION_OFFSET
sbi_platform.h, 523
sbi_platform_operations, 143
cold_boot_allowed, 144
domains_init, 144
early_exit, 144
early_init, 144
emulate_load, 144
emulate_store, 144
extensions_init, 144
final_exit, 144
final_init, 144
get_tlb_num_entries, 144
get_tlbr_flush_limit, 145
ipi_init, 145
irqchip_init, 145
misa_check_extension, 145
misa_get_xlen, 145
mpxy_init, 145
nascent_init, 145
pmu_init, 145
pmu_xlate_to_mhpmevent, 145
timer_init, 145
vendor_ext_check, 145
vendor_ext_provider, 145
sbi_platform_ops
sbi_platform.h, 523
SBI_PLATFORM_OPS_OFFSET
sbi_platform.h, 523
sbi_platform_pmu_init
sbi_platform.h, 534
sbi_platform_pmu_xlate_to_mhpmevent
sbi_platform.h, 534
sbi_platform_ptr
sbi_platform.h, 523
SBI_PLATFORM_RESERVED_OFFSET
sbi_platform.h, 523
sbi_platform_thishart_ptr
sbi_platform.h, 523
sbi_platform_timer_init
sbi_platform.h, 536
sbi_platform_tlb_fifo_num_entries
sbi_platform.h, 536
SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT
sbi_platform.h, 523
sbi_platform_tlbr_flush_limit
sbi_platform.h, 537
sbi_platform_vendor_ext_check
sbi_platform.h, 537
sbi_platform_vendor_ext_provider

sbi_platform.h, 538
SBI_PLATFORM_VERSION
sbi_platform.h, 523
SBI_PLATFORM_VERSION_OFFSET
sbi_platform.h, 523
sbi_pmu.c
get_cidx_code, 1011
get_cidx_type, 1011
hw_event_map, 1027
num_hw_ctrs, 1027
num_hw_events, 1027
phs_ptr_offset, 1028
pmu_add_hw_event_map, 1011
pmu_ctr_enable_irq_hw, 1011
pmu_ctr_find_fixed_hw, 1012
pmu_ctr_find_fw, 1012
pmu_ctr_find_hw, 1012
pmu_ctr_start_fw, 1013
pmu_ctr_start_hw, 1013
pmu_ctr_stop_fw, 1014
pmu_ctr_stop_hw, 1014
pmu_ctr_validate, 1015
pmu_ctr_write_hw, 1015
pmu_dev, 1028
pmu_event_range_overlap, 1016
pmu_event_select_overlap, 1016
pmu_event_validate, 1017
pmu_fixed_ctr_update_inhibit_bits, 1017
pmu_get_hart_state_ptr, 1011
pmu_reset_event_map, 1017
pmu_reset_hw_mhpmevent, 1018
pmu_set_hart_state_ptr, 1011
pmu_sse_cb_ops, 1028
pmu_sse_complete, 1018
pmu_sse_disable, 1018
pmu_sse_enable, 1018
pmu_thishart_state_ptr, 1011
pmu_update_hw_mhpmevent, 1019
pmu_update_inhibit_flags, 1019
sbi_pmu_add_hw_event_counter_map, 1019
sbi_pmu_add_raw_event_counter_map, 1020
sbi_pmu_ctr_cfg_match, 1021
sbi_pmu_ctr_fw_read, 1021
sbi_pmu_ctr_get_info, 1022
sbi_pmu_ctr_incr_fw, 1023
sbi_pmu_ctr_start, 1023
sbi_pmu_ctr_stop, 1023
sbi_pmu_event_get_info, 1024
sbi_pmu_exit, 1024
sbi_pmu_get_device, 1025
sbi_pmu_init, 1025
sbi_pmu_irq_bit, 1026
sbi_pmu_num_ctr, 1026
sbi_pmu_ovf_irq, 1027
sbi_pmu_set_device, 1027
total_ctrs, 1028
sbi_pmu.h
sbi_pmu_add_hw_event_counter_map, 545
sbi_pmu_add_raw_event_counter_map, 546
sbi_pmu_ctr_cfg_match, 547
sbi_pmu_ctr_fw_read, 547
sbi_pmu_ctr_get_info, 548
sbi_pmu_ctr_incr_fw, 549
SBI_PMU_CTR_MAX, 545
sbi_pmu_ctr_start, 549
sbi_pmu_ctr_stop, 550
SBI_PMU_CY_IR_MASK, 545
sbi_pmu_event_get_info, 550
sbi_pmu_exit, 551
SBI_PMU_FIXED_CTR_MASK, 545
SBI_PMU_FW_CTR_MAX, 545
sbi_pmu_get_device, 551
SBI_PMU_HW_CTR_MAX, 545
SBI_PMU_HW_EVENT_MAX, 545
sbi_pmu_init, 551
sbi_pmu_irq_bit, 552
sbi_pmu_num_ctr, 552
sbi_pmu_ovf_irq, 553
sbi_pmu_set_device, 553
sbi_pmu_add_hw_event_counter_map
sbi_pmu.c, 1019
sbi_pmu.h, 545
sbi_pmu_add_raw_event_counter_map
sbi_pmu.c, 1020
sbi_pmu.h, 546
SBI_PMU_CFG_FLAG_AUTO_START
sbi_ecall_interface.h, 398
SBI_PMU_CFG_FLAG_CLEAR_VALUE
sbi_ecall_interface.h, 398
SBI_PMU_CFG_FLAG_SET_MINH
sbi_ecall_interface.h, 398
SBI_PMU_CFG_FLAG_SET_SINH
sbi_ecall_interface.h, 398
SBI_PMU_CFG_FLAG_SET_UINH
sbi_ecall_interface.h, 399
SBI_PMU_CFG_FLAG_SET_VSINH
sbi_ecall_interface.h, 399
SBI_PMU_CFG_FLAG_SET_VUINH
sbi_ecall_interface.h, 399
SBI_PMU_CFG_FLAG_SKIP_MATCH
sbi_ecall_interface.h, 399
sbi_pmu_ctr_cfg_match
sbi_pmu.c, 1021
sbi_pmu.h, 547
sbi_pmu_ctr_fw_read
sbi_pmu.c, 1021
sbi_pmu.h, 547
sbi_pmu_ctr_get_info
sbi_pmu.c, 1022
sbi_pmu.h, 548
sbi_pmu_ctr_incr_fw
sbi_pmu.c, 1023
sbi_pmu.h, 549
SBI_PMU_CTR_MAX
sbi_pmu.h, 545
sbi_pmu_ctr_start

sbi_pmu.c, 1023
sbi_pmu.h, 549
sbi_pmu_ctr_stop
sbi_pmu.c, 1023
sbi_pmu.h, 550
sbi_pmu_ctr_type
sbi_ecall_interface.h, 404
SBI_PMU_CTR_TYPE_FW
sbi_ecall_interface.h, 404
SBI_PMU_CTR_TYPE_HW
sbi_ecall_interface.h, 404
SBI_PMU_CY_IR_MASK
sbi_pmu.h, 545
sbi_pmu_device, 146
fw_counter_match_encoding, 146
fw_counter_read_value, 146
fw_counter_start, 146
fw_counter_stop, 146
fw_counter_width, 146
fw_counter_write_value, 146
fw_event_validate_encoding, 147
hw_counter_disable_irq, 147
hw_counter_enable_irq, 147
hw_counter_filter_mode, 147
hw_counter_irq_bit, 147
name, 147
sbi_pmu_event_get_info
sbi_pmu.c, 1024
sbi_pmu.h, 550
SBI_PMU_EVENT_HW_CACHE_ID_MASK
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_HW_CACHE_ID_OFFSET
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_HW_CACHE_OPS_RESULT
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_IDX_CODE_MASK
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_IDX_INVALID
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_IDX_MASK
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_IDX_TYPE_MASK
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_IDX_TYPE_OFFSET
sbi_ecall_interface.h, 399
sbi_pmu_event_info, 147
event_data, 147
event_idx, 147
output, 147
SBI_PMU_EVENT_RAW_IDX
sbi_ecall_interface.h, 399
SBI_PMU_EVENT_RAW_V2_IDX
sbi_ecall_interface.h, 400
SBI_PMU_EVENT_TYPE_FW
sbi_ecall_interface.h, 404
SBI_PMU_EVENT_TYPE_HW
sbi_ecall_interface.h, 404
SBI_PMU_EVENT_TYPE_HW_CACHE
sbi_ecall_interface.h, 404
SBI_PMU_EVENT_TYPE_HW_RAW
sbi_ecall_interface.h, 404
SBI_PMU_EVENT_TYPE_HW_RAW_V2
sbi_ecall_interface.h, 404
sbi_pmu_event_type_id
sbi_ecall_interface.h, 404
SBI_PMU_EVENT_TYPE_MAX
sbi_ecall_interface.h, 404
sbi_pmu_exit
sbi_pmu.c, 1024
sbi_pmu.h, 551
SBI_PMU_FIXED_CTR_MASK
sbi_pmu.h, 545
SBI_PMU_FW_ACCESS_LOAD
sbi_ecall_interface.h, 405
SBI_PMU_FW_ACCESS_STORE
sbi_ecall_interface.h, 405
SBI_PMU_FW_CTR_MAX
sbi_pmu.h, 545
sbi_pmu_fw_event_code_id
sbi_ecall_interface.h, 404
SBI_PMU_FW_FENCE_I_RECVD
sbi_ecall_interface.h, 405
SBI_PMU_FW_FENCE_I_SENT
sbi_ecall_interface.h, 405
SBI_PMU_FW_HFENCE_GVMA_RCVD
sbi_ecall_interface.h, 405
SBI_PMU_FW_HFENCE_GVMA_SENT
sbi_ecall_interface.h, 405
SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD
sbi_ecall_interface.h, 405
SBI_PMU_FW_HFENCE_GVMA_VMID_SENT
sbi_ecall_interface.h, 405
SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD
sbi_ecall_interface.h, 405
SBI_PMU_FW_HFENCE_VVMA_ASID_SENT
sbi_ecall_interface.h, 405
SBI_PMU_FW_HFENCE_VVMA_RCVD
sbi_ecall_interface.h, 405
SBI_PMU_FW_HFENCE_VVMA_SENT
sbi_ecall_interface.h, 405
SBI_PMU_FW_ILLEGAL_INSN
sbi_ecall_interface.h, 405
SBI_PMU_FW_IPI_RECVD
sbi_ecall_interface.h, 405
SBI_PMU_FW_IPI_SENT
sbi_ecall_interface.h, 405
SBI_PMU_FW_MAX
sbi_ecall_interface.h, 405
SBI_PMU_FW_MISALIGNED_LOAD
sbi_ecall_interface.h, 405
SBI_PMU_FW_MISALIGNED_STORE
sbi_ecall_interface.h, 405

SBI_PMU_FW_PLATFORM
sbi_ecall_interface.h, 405
SBI_PMU_FW_RESERVED_MAX
sbi_ecall_interface.h, 405
SBI_PMU_FW_SET_TIMER
sbi_ecall_interface.h, 405
SBI_PMU_FW_SFENCE_VMA_ASID_RCVD
sbi_ecall_interface.h, 405
SBI_PMU_FW_SFENCE_VMA_ASID_SENT
sbi_ecall_interface.h, 405
SBI_PMU_FW_SFENCE_VMA_RCVD
sbi_ecall_interface.h, 405
SBI_PMU_FW_SFENCE_VMA_SENT
sbi_ecall_interface.h, 405
sbi_pmu_get_device
sbi_pmu.c, 1025
sbi_pmu.h, 551
SBI_PMU_HW_BRANCH_INSTRUCTIONS
sbi_ecall_interface.h, 406
SBI_PMU_HW_BRANCH_MISSES
sbi_ecall_interface.h, 406
SBI_PMU_HW_BUS_CYCLES
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_BPU
sbi_ecall_interface.h, 405
SBI_PMU_HW_CACHE_DTLB
sbi_ecall_interface.h, 405
sbi_pmu_hw_cache_id
sbi_ecall_interface.h, 405
SBI_PMU_HW_CACHE_ITLB
sbi_ecall_interface.h, 405
SBI_PMU_HW_CACHE_L1D
sbi_ecall_interface.h, 405
SBI_PMU_HW_CACHE_L1I
sbi_ecall_interface.h, 405
SBI_PMU_HW_CACHE_LL
sbi_ecall_interface.h, 405
SBI_PMU_HW_CACHE_MAX
sbi_ecall_interface.h, 405
SBI_PMU_HW_CACHE_MISSES
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_NODE
sbi_ecall_interface.h, 405
sbi_pmu_hw_cache_op_id
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_OP_MAX
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_OP_PREFETCH
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_OP_READ
sbi_ecall_interface.h, 406
sbi_pmu_hw_cache_op_result_id
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_OP_WRITE
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_REFERENCES
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_RESULT_ACCESS
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_RESULT_MAX
sbi_ecall_interface.h, 406
SBI_PMU_HW_CACHE_RESULT_MISS
sbi_ecall_interface.h, 406
SBI_PMU_HW_CPU_CYCLES
sbi_ecall_interface.h, 406
SBI_PMU_HW_CTR_MAX
sbi_pmu.h, 545
SBI_PMU_HW_EVENT_MAX
sbi_pmu.h, 545
SBI_PMU_HW_GENERAL_MAX
sbi_ecall_interface.h, 406
sbi_pmu_hw_generic_events_t
sbi_ecall_interface.h, 406
SBI_PMU_HW_INSTRUCTIONS
sbi_ecall_interface.h, 406
SBI_PMU_HW_NO_EVENT
sbi_ecall_interface.h, 406
SBI_PMU_HW_REF_CPU_CYCLES
sbi_ecall_interface.h, 406
SBI_PMU_HW_STALLED_CYCLES_BACKEND
sbi_ecall_interface.h, 406
SBI_PMU_HW_STALLED_CYCLES_FRONTEND
sbi_ecall_interface.h, 406
sbi_pmu_init
sbi_pmu.c, 1025
sbi_pmu.h, 551
sbi_pmu_irq_bit
sbi_pmu.c, 1026
sbi_pmu.h, 552
sbi_pmu_num_ctr
sbi_pmu.c, 1026
sbi_pmu.h, 552
sbi_pmu_ovf_irq
sbi_pmu.c, 1027
sbi_pmu.h, 553
sbi_pmu_set_device
sbi_pmu.c, 1027
sbi_pmu.h, 553
SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT
sbi_ecall_interface.h, 400
SBI_PMU_START_FLAG_SET_INIT_VALUE
sbi_ecall_interface.h, 400
SBI_PMU_STOP_FLAG_RESET
sbi_ecall_interface.h, 400
SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT
sbi_ecall_interface.h, 400
sbi_popcount
sbi_bitops.h, 320
sbi_printf
sbi_console.c, 817
sbi_putc
sbi_console.c, 818
sbi_console.h, 331
sbi_puts
sbi_console.c, 819
sbi_console.h, 332

sbi_regs_from_virt
sbi_trap.h, 607
sbi_regs_gva
sbi_trap.h, 607
sbi_rpmi_cppc
fdt_cppc_rpmi.c, 1133
sbi_scratch, 148
fw_heap_offset, 148
fw_heap_size, 148
fw_rw_offset, 148
fw_size, 148
fw_start, 148
hartid_to_scratch, 148
hartindex, 149
next_addr, 149
next_arg1, 149
next_mode, 149
options, 149
platform_addr, 149
tmp0, 149
trap_context, 149
warmboot_addr, 149
sbi_scratch.c
extra_lock, 1031
extra_offset, 1031
hartid2scratch, 1029
hartindex_to_hartid_table, 1031
hartindex_to_scratch_table, 1031
last_hartindex_having_scratch, 1032
sbi_hartid_to_hartindex, 1029
sbi_scratch_alloc_offset, 1029
sbi_scratch_free_offset, 1030
sbi_scratch_init, 1030
sbi_scratch_used_space, 1031
sbi_scratch.h
current_hartindex, 556
hartindex_to_hartid_table, 562
hartindex_to_scratch_table, 562
last_hartindex_having_scratch, 562
sbi_hartid_to_hartindex, 560
sbi_hartid_to_scratch, 556
sbi_hartid_valid, 556
sbi_hartindex_to_hartid, 556
sbi_hartindex_to_scratch, 557
sbi_hartindex_valid, 557
sbi_scratch_alloc_offset, 560
sbi_scratch_alloc_type_offset, 557
SBI_SCRATCH_DEBUG_PRINTS, 559
SBI_SCRATCH_EXTRA_SPACE_OFFSET, 557
sbi_scratch_free_offset, 561
SBI_SCRATCH_FW_HEAP_OFFSET, 557
SBI_SCRATCH_FW_HEAP_SIZE_OFFSET, 557
SBI_SCRATCH_FW_RW_OFFSET, 557
SBI_SCRATCH_FW_SIZE_OFFSET, 557
SBI_SCRATCH_FW_START_OFFSET, 557
SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET, 557
SBI_SCRATCH_HARTINDEX_OFFSET, 558
sbi_scratch_init, 561
sbi_scratch_last_hartindex, 558
SBI_SCRATCH_NEXT_ADDR_OFFSET, 558
SBI_SCRATCH_NEXT_ARG1_OFFSET, 558
SBI_SCRATCH_NEXT_MODE_OFFSET, 558
SBI_SCRATCH_NO_BOOT_PRINTS, 559
sbi_scratch_offset_ptr, 558
sbi_scratch_options, 559
SBI_SCRATCH_OPTIONS_OFFSET, 558
SBI_SCRATCH_PLATFORM_ADDR_OFFSET, 558
sbi_scratch_read_type, 558
SBI_SCRATCH_SIZE, 558
sbi_scratch_thishart_arg1_ptr, 559
sbi_scratch_thishart_offset_ptr, 559
sbi_scratch_thishart_ptr, 559
SBI_SCRATCH_TMP0_OFFSET, 559
SBI_SCRATCH_TRAP_CONTEXT_OFFSET, 559
sbi_scratch_used_space, 561
SBI_SCRATCH_WARMBOOT_ADDR_OFFSET, 559
sbi_scratch_write_type, 559
sbi_scratch_alloc_offset
sbi_scratch.c, 1029
sbi_scratch.h, 560
sbi_scratch_alloc_type_offset
sbi_scratch.h, 557
SBI_SCRATCH_DEBUG_PRINTS
sbi_scratch.h, 559
SBI_SCRATCH_EXTRA_SPACE_OFFSET
sbi_scratch.h, 557
sbi_scratch_free_offset
sbi_scratch.c, 1030
sbi_scratch.h, 561
SBI_SCRATCH_FW_HEAP_OFFSET
sbi_scratch.h, 557
SBI_SCRATCH_FW_HEAP_SIZE_OFFSET
sbi_scratch.h, 557
SBI_SCRATCH_FW_RW_OFFSET
sbi_scratch.h, 557
SBI_SCRATCH_FW_SIZE_OFFSET
sbi_scratch.h, 557
SBI_SCRATCH_FW_START_OFFSET
sbi_scratch.h, 557
SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET
sbi_scratch.h, 557
SBI_SCRATCH_HARTINDEX_OFFSET
sbi_scratch.h, 558
sbi_scratch_init
sbi_scratch.c, 1030
sbi_scratch.h, 561
sbi_scratch_last_hartindex
sbi_scratch.h, 558
SBI_SCRATCH_NEXT_ADDR_OFFSET
sbi_scratch.h, 558
SBI_SCRATCH_NEXT_ARG1_OFFSET
sbi_scratch.h, 558
SBI_SCRATCH_NEXT_MODE_OFFSET

sbi_scratch.h, 558
SBI_SCRATCH_NO_BOOT_PRINTS
sbi_scratch.h, 559
sbi_scratch_offset_ptr
sbi_scratch.h, 558
sbi_scratch_options
sbi_scratch.h, 559
SBI_SCRATCH_OPTIONS_OFFSET
sbi_scratch.h, 558
SBI_SCRATCH_PLATFORM_ADDR_OFFSET
sbi_scratch.h, 558
sbi_scratch_read_type
sbi_scratch.h, 558
SBI_SCRATCH_SIZE
sbi_scratch.h, 558
sbi_scratch_thishart_arg1_ptr
sbi_scratch.h, 559
sbi_scratch_thishart_offset_ptr
sbi_scratch.h, 559
sbi_scratch_thishart_ptr
sbi_scratch.h, 559
SBI_SCRATCH_TMP0_OFFSET
sbi_scratch.h, 559
SBI_SCRATCH_TRAP_CONTEXT_OFFSET
sbi_scratch.h, 559
sbi_scratch_used_space
sbi_scratch.c, 1031
sbi_scratch.h, 561
SBI_SCRATCH_WARMBOOT_ADDR_OFFSET
sbi_scratch.h, 559
sbi_scratch_write_type
sbi_scratch.h, 559
SBI_SMEPMP_RESV_ENTRY
sbi_hart.h, 429
sbi_snprintf
sbi_console.c, 820
SBI_SPEC_VERSION_MAJOR_MASK
sbi_ecall_interface.h, 400
SBI_SPEC_VERSION_MAJOR_OFFSET
sbi_ecall_interface.h, 400
SBI_SPEC_VERSION_MINOR_MASK
sbi_ecall_interface.h, 400
sbi_sprintf
sbi_console.c, 820
SBI_SRST_RESET_REASON_NONE
sbi_ecall_interface.h, 400
SBI_SRST_RESET_REASON_SYSFAIL
sbi_ecall_interface.h, 400
SBI_SRST_RESET_TYPE_COLD_REBOOT
sbi_ecall_interface.h, 400
SBI_SRST_RESET_TYPE_LAST
sbi_ecall_interface.h, 400
SBI_SRST_RESET_TYPE_SHUTDOWN
sbi_ecall_interface.h, 400
SBI_SRST_RESET_TYPE_WARM_REBOOT
sbi_ecall_interface.h, 400
sbi_sse.c
assert_field_offset, 1034–1036
copy_attrs, 1036
EVENT_COUNT, 1034
EVENT_IS_GLOBAL, 1034
global_event_count, 1063
global_events, 1063
local_event_count, 1063
sbi_sse_attr_check, 1036
sbi_sse_complete, 1037
sbi_sse_disable, 1037
sbi_sse_enable, 1038
sbi_sse_exit, 1039
sbi_sse_hart_mask, 1039
sbi_sse_hart_unmask, 1040
sbi_sse_init, 1040
sbi_sse_inject_event, 1041
sbi_sse_inject_from_ecall, 1041
sbi_sse_process_pending_events, 1042
sbi_sse_read_attrs, 1043
sbi_sse_register, 1043
sbi_sse_set_cb_ops, 1044
sbi_sse_unregister, 1045
sbi_sse_write_attrs, 1045
shs_ptr_off, 1063
sse_enabled_event_lock, 1046
sse_enabled_event_unlock, 1046
sse_event_add_to_list, 1047
sse_event_check_inject, 1047
sse_event_complete, 1048
sse_event_count_init, 1048
sse_event_disable, 1049
sse_event_enable, 1049
sse_event_get, 1050
sse_event_init, 1051
sse_event_inject, 1051
sse_event_invoke_cb, 1034
sse_event_is_global, 1052
sse_event_is_local, 1052
sse_event_is_ready, 1053
sse_event_pending, 1053
sse_event_put, 1054
sse_event_register, 1054
sse_event_remove_from_list, 1055
sse_event_resume, 1055
sse_event_set_attr, 1056
sse_event_set_attr_check, 1056
sse_event_set_hart_id_check, 1057
sse_event_set_pending, 1057
sse_event_set_state, 1058
sse_event_state, 1058
sse_event_unregister, 1058
sse_get_global_event, 1059
sse_get_hart_state, 1059
sse_get_hart_state_ptr, 1034
sse_global_init, 1060
sse_inject_event, 1060
sse_inject_fifo_mem_off, 1063
sse_inject_fifo_off, 1063
sse_interrupted_flags, 1061

sse_ipi_inject_event, 1063
sse_ipi_inject_ops, 1063
sse_ipi_inject_process, 1061
sse_ipi_inject_send, 1061
sse_local_init, 1062
sse_set_hart_state_ptr, 1035
sse_thishart_state_ptr, 1035
sse_write_attrs, 1062
supported_events, 1064
sbi_sse.h
EXC_MODE_PP, 566
EXC_MODE_PP_SHIFT, 566
EXC_MODE_PV, 566
EXC_MODE_PV_SHIFT, 566
EXC_MODE_SSTATUS_SPIE, 566
EXC_MODE_SSTATUS_SPIE_SHIFT, 566
sbi_sse_complete, 566
sbi_sse_disable, 567
sbi_sse_enable, 567
sbi_sse_exit, 568
sbi_sse_hart_mask, 569
sbi_sse_hart_unmask, 569
sbi_sse_init, 569
sbi_sse_inject_event, 570
sbi_sse_inject_from_ecall, 570
sbi_sse_process_pending_events, 571
sbi_sse_read_attrs, 572
sbi_sse_register, 572
sbi_sse_set_cb_ops, 573
sbi_sse_unregister, 574
sbi_sse_write_attrs, 574
sbi_sse_attr_check
sbi_sse.c, 1036
SBI_SSE_ATTR_CONFIG
sbi_ecall_interface.h, 406
SBI_SSE_ATTR_CONFIG_ONESHOT
sbi_ecall_interface.h, 400
SBI_SSE_ATTR_ENTRY_ARG
sbi_ecall_interface.h, 406
SBI_SSE_ATTR_ENTRY_PC
sbi_ecall_interface.h, 406
sbi_sse_attr_id
sbi_ecall_interface.h, 406
SBI_SSE_ATTR_INTERRUPTED_A6
sbi_ecall_interface.h, 407
SBI_SSE_ATTR_INTERRUPTED_A7
sbi_ecall_interface.h, 407
SBI_SSE_ATTR_INTERRUPTED_FLAGS
sbi_ecall_interface.h, 407
SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV
sbi_ecall_interface.h, 401
SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP
sbi_ecall_interface.h, 401
SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE
sbi_ecall_interface.h, 401
SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP
sbi_ecall_interface.h, 401
SBI_SSE_ATTR_INTERRUPTED_SEPC
sbi_ecall_interface.h, 407
SBI_SSE_ATTR_MAX
sbi_ecall_interface.h, 407
SBI_SSE_ATTR_PREFERRED_HART
sbi_ecall_interface.h, 406
SBI_SSE_ATTR_PRIO
sbi_ecall_interface.h, 406
SBI_SSE_ATTR_STATUS
sbi_ecall_interface.h, 406
SBI_SSE_ATTR_STATUS_INJECT_OFFSET
sbi_ecall_interface.h, 401
SBI_SSE_ATTR_STATUS_PENDING_OFFSET
sbi_ecall_interface.h, 401
SBI_SSE_ATTR_STATUS_STATE_MASK
sbi_ecall_interface.h, 401
SBI_SSE_ATTR_STATUS_STATE_OFFSET
sbi_ecall_interface.h, 401
sbi_sse_cb_ops, 149
complete_cb, 150
disable_cb, 150
enable_cb, 150
register_cb, 150
set_hartid_cb, 150
unregister_cb, 150
sbi_sse_complete
sbi_sse.c, 1037
sbi_sse.h, 566
sbi_sse_disable
sbi_sse.c, 1037
sbi_sse.h, 567
sbi_sse_enable
sbi_sse.c, 1038
sbi_sse.h, 567
SBI_SSE_EVENT_GLOBAL_BIT
sbi_ecall_interface.h, 401
SBI_SSE_EVENT_GLOBAL_PLAT_0_END
sbi_ecall_interface.h, 401
SBI_SSE_EVENT_GLOBAL_PLAT_0_START
sbi_ecall_interface.h, 401
SBI_SSE_EVENT_GLOBAL_PLAT_1_END
sbi_ecall_interface.h, 401
SBI_SSE_EVENT_GLOBAL_PLAT_1_START
sbi_ecall_interface.h, 401
SBI_SSE_EVENT_GLOBAL_PLAT_2_END
sbi_ecall_interface.h, 401
SBI_SSE_EVENT_GLOBAL_PLAT_2_START
sbi_ecall_interface.h, 401
SBI_SSE_EVENT_GLOBAL_PLAT_3_END
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_GLOBAL_PLAT_3_START
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_GLOBAL_RAS
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_GLOBAL_SOFTWARE
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_DOUBLE_TRAP
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PLAT_0_END

sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PLAT_0_START
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PLAT_1_END
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PLAT_1_START
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PLAT_2_END
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PLAT_2_START
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PLAT_3_END
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PLAT_3_START
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_PMU
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_RAS
sbi_ecall_interface.h, 402
SBI_SSE_EVENT_LOCAL_SOFTWARE
sbi_ecall_interface.h, 403
SBI_SSE_EVENT_PLATFORM_BIT
sbi_ecall_interface.h, 403
sbi_sse_exit
sbi_sse.c, 1039
sbi_sse.h, 568
sbi_sse_hart_mask
sbi_sse.c, 1039
sbi_sse.h, 569
sbi_sse_hart_unmask
sbi_sse.c, 1040
sbi_sse.h, 569
sbi_sse_init
sbi_sse.c, 1040
sbi_sse.h, 569
sbi_sse_inject_event
sbi_sse.c, 1041
sbi_sse.h, 570
sbi_sse_inject_from_ecall
sbi_sse.c, 1041
sbi_sse.h, 570
sbi_sse_process_pending_events
sbi_sse.c, 1042
sbi_sse.h, 571
sbi_sse_read_attrs
sbi_sse.c, 1043
sbi_sse.h, 572
sbi_sse_register
sbi_sse.c, 1043
sbi_sse.h, 572
sbi_sse_set_cb_ops
sbi_sse.c, 1044
sbi_sse.h, 573
sbi_sse_state
sbi_ecall_interface.h, 407
SBI_SSE_STATE_ENABLED
sbi_ecall_interface.h, 407
SBI_SSE_STATE_REGISTERED
sbi_ecall_interface.h, 407
SBI_SSE_STATE_RUNNING
sbi_ecall_interface.h, 407
SBI_SSE_STATE_UNUSED
sbi_ecall_interface.h, 407
sbi_sse_unregister
sbi_sse.c, 1045
sbi_sse.h, 574
sbi_sse_write_attrs
sbi_sse.c, 1045
sbi_sse.h, 574
sbi_st_access_emulator
sbi_trap_ldst.c, 1104
sbi_store_access_handler
sbi_trap_ldst.c, 1104
sbi_trap_ldst.h, 617
sbi_strchr
sbi_string.c, 1066
sbi_string.h, 579
sbi_strcmp
sbi_string.c, 1066
sbi_string.h, 579
sbi_strcpy
sbi_string.c, 1067
sbi_string.h, 579
sbi_string.c
sbi_memchr, 1065
sbi_memcmp, 1065
sbi_memcpy, 1065
sbi_memmove, 1065
sbi_memset, 1066
sbi_strchr, 1066
sbi_strcmp, 1066
sbi_strcpy, 1067
sbi_strlen, 1067
sbi_strncmp, 1067
sbi_strncpy, 1068
sbi_strnlen, 1068
sbi_strrchr, 1068
sbi_string.h
sbi_memchr, 577
sbi_memcmp, 577
sbi_memcpy, 577
sbi_memmove, 578
sbi_memset, 578
sbi_strchr, 579
sbi_strcmp, 579
sbi_strcpy, 579
sbi_strlen, 579
sbi_strncmp, 580
sbi_strncpy, 580
sbi_strnlen, 581
sbi_strrchr, 581
sbi_strlen
sbi_string.c, 1067
sbi_string.h, 579
sbi_strncmp
sbi_string.c, 1067

sbi_string.h, 580
sbi_strncpy
sbi_string.c, 1068
sbi_string.h, 580
sbi_strnlen
sbi_string.c, 1068
sbi_string.h, 581
sbi_strrchr
sbi_string.c, 1068
sbi_string.h, 581
SBI_SUCCESS
sbi_ecall_interface.h, 403
SBI_SUSP_PLATFORM_SLEEP_START
sbi_ecall_interface.h, 403
SBI_SUSP_SLEEP_TYPE_LAST
sbi_ecall_interface.h, 403
SBI_SUSP_SLEEP_TYPE_SUSPEND
sbi_ecall_interface.h, 403
sbi_system.c
SBI_LIST_HEAD, 1069
sbi_system_reset, 1069
sbi_system_reset_add_device, 1070
sbi_system_reset_get_device, 1071
sbi_system_reset_supported, 1071
sbi_system_suspend, 1072
sbi_system_suspend_get_device, 1073
sbi_system_suspend_set_device, 1073
sbi_system_suspend_supported, 1073
sbi_system_suspend_test, 1074
sbi_system_suspend_test_check, 1074
sbi_system_suspend_test_enable, 1074
sbi_system_suspend_test_suspend, 1074
suspend_dev, 1074
sbi_system.h
sbi_system_reset, 583
sbi_system_reset_add_device, 583
sbi_system_reset_get_device, 584
sbi_system_reset_supported, 585
sbi_system_suspend, 585
sbi_system_suspend_get_device, 586
sbi_system_suspend_set_device, 586
sbi_system_suspend_supported, 587
sbi_system_suspend_test_enable, 587
to_system_reset_device, 587
sbi_system_reset
sbi_system.c, 1069
sbi_system.h, 583
sbi_system_reset_add_device
sbi_system.c, 1070
sbi_system.h, 583
sbi_system_reset_device, 150
name, 151
node, 151
system_reset, 151
system_reset_check, 151
sbi_system_reset_get_device
sbi_system.c, 1071
sbi_system.h, 584
sbi_system_reset_supported
sbi_system.c, 1071
sbi_system.h, 585
sbi_system_suspend
sbi_system.c, 1072
sbi_system.h, 585
sbi_system_suspend_device, 151
name, 151
system_suspend, 151
system_suspend_check, 151
sbi_system_suspend_get_device
sbi_system.c, 1073
sbi_system.h, 586
sbi_system_suspend_set_device
sbi_system.c, 1073
sbi_system.h, 586
sbi_system_suspend_supported
sbi_system.c, 1073
sbi_system.h, 587
sbi_system_suspend_test
sbi_system.c, 1074
sbi_system_suspend_test_check
sbi_system.c, 1074
sbi_system_suspend_test_enable
sbi_system.c, 1074
sbi_system.h, 587
sbi_system_suspend_test_suspend
sbi_system.c, 1074
sbi_timer.c
get_ticks, 1076
get_time_val, 1082
nop_delay_fn, 1076
sbi_timer_delay_loop, 1076
sbi_timer_event_start, 1077
sbi_timer_exit, 1077
sbi_timer_get_delta, 1078
sbi_timer_get_device, 1078
sbi_timer_init, 1078
sbi_timer_process, 1079
sbi_timer_set_delta, 1079
sbi_timer_set_device, 1080
sbi_timer_value, 1080
sbi_timer_virt_value, 1080
sbi_timer_waitms_until, 1081
time_delta_off, 1082
timer_dev, 1082
sbi_timer.h
sbi_timer_delay_loop, 589
sbi_timer_event_start, 590
sbi_timer_exit, 590
sbi_timer_get_delta, 591
sbi_timer_get_device, 591
sbi_timer_init, 591
sbi_timer_mdelay, 592
sbi_timer_process, 593
sbi_timer_set_delta, 593
sbi_timer_set_device, 593
sbi_timer_udelay, 593

sbi_timer_value, 594
sbi_timer_virt_value, 594
sbi_timer_waitms_until, 595
sbi_timer_delay_loop
sbi_timer.c, 1076
sbi_timer.h, 589
sbi_timer_device, 152
name, 152
timer_event_start, 152
timer_event_stop, 152
timer_freq, 152
timer_value, 152
warm_init, 152
sbi_timer_event_start
sbi_timer.c, 1077
sbi_timer.h, 590
sbi_timer_exit
sbi_timer.c, 1077
sbi_timer.h, 590
sbi_timer_get_delta
sbi_timer.c, 1078
sbi_timer.h, 591
sbi_timer_get_device
sbi_timer.c, 1078
sbi_timer.h, 591
sbi_timer_init
sbi_timer.c, 1078
sbi_timer.h, 591
sbi_timer_mdelay
sbi_timer.h, 592
sbi_timer_process
sbi_timer.c, 1079
sbi_timer.h, 593
sbi_timer_set_delta
sbi_timer.c, 1079
sbi_timer.h, 593
sbi_timer_set_device
sbi_timer.c, 1080
sbi_timer.h, 593
sbi_timer_udelay
sbi_timer.h, 593
sbi_timer_value
sbi_timer.c, 1080
sbi_timer.h, 594
sbi_timer_virt_value
sbi_timer.c, 1080
sbi_timer.h, 594
sbi_timer_waitms_until
sbi_timer.c, 1081
sbi_timer.h, 595
sbi_tlb.c
sbi_tlb_init, 1083
sbi_tlb_local_fence_i, 1084
sbi_tlb_local_hfence_gvma, 1084
sbi_tlb_local_hfence_gvma_vmid, 1085
sbi_tlb_local_hfence_vvma, 1085
sbi_tlb_local_hfence_vvma_asid, 1086
sbi_tlb_local_sfence_vma, 1086
sbi_tlb_local_sfence_vma_asid, 1087
sbi_tlb_request, 1087
tlb_entry_local_process, 1088
tlb_entry_process, 1089
tlb_event, 1094
tlb_fifo_mem_off, 1094
tlb_fifo_off, 1094
tlb_flush_all, 1090
tlb_ops, 1094
tlb_process, 1090
tlb_process_once, 1091
tlb_range_check, 1091
tlb_range_flush_limit, 1094
tlb_sync, 1092
tlb_sync_off, 1094
tlb_type_to_pmu_fw_event, 1094
tlb_update, 1092
tlb_update_cb, 1093
sbi_tlb.h
SBI_TLB_FENCE_I, 598
SBI_TLB_FLUSH_ALL, 598
SBI_TLB_HFENCE_GVMA, 598
SBI_TLB_HFENCE_GVMA_VMID, 598
SBI_TLB_HFENCE_VVMA, 598
SBI_TLB_HFENCE_VVMA_ASID, 598
SBI_TLB_INFO_INIT, 598
SBI_TLB_INFO_SIZE, 598
sbi_tlb_init, 598
sbi_tlb_request, 599
SBI_TLB_SFENCE_VMA, 598
SBI_TLB_SFENCE_VMA_ASID, 598
sbi_tlb_type, 598
SBI_TLB_TYPE_MAX, 598
SBI_TLB_FENCE_I
sbi_tlb.h, 598
SBI_TLB_FLUSH_ALL
sbi_tlb.h, 598
SBI_TLB_HFENCE_GVMA
sbi_tlb.h, 598
SBI_TLB_HFENCE_GVMA_VMID
sbi_tlb.h, 598
SBI_TLB_HFENCE_VVMA
sbi_tlb.h, 598
SBI_TLB_HFENCE_VVMA_ASID
sbi_tlb.h, 598
sbi_tlb_info, 153
asid, 153
size, 153
smask, 153
start, 153
type, 153
vmid, 153
SBI_TLB_INFO_INIT
sbi_tlb.h, 598
SBI_TLB_INFO_SIZE
sbi_tlb.h, 598
sbi_tlb_init
sbi_tlb.c, 1083

sbi_tlb.h, 598
sbi_tlb_local_fence_i
sbi_tlb.c, 1084
sbi_tlb_local_hfence_gvma
sbi_tlb.c, 1084
sbi_tlb_local_hfence_gvma_vmid
sbi_tlb.c, 1085
sbi_tlb_local_hfence_vvma
sbi_tlb.c, 1085
sbi_tlb_local_hfence_vvma_asid
sbi_tlb.c, 1086
sbi_tlb_local_sfence_vma
sbi_tlb.c, 1086
sbi_tlb_local_sfence_vma_asid
sbi_tlb.c, 1087
sbi_tlb_request
sbi_tlb.c, 1087
sbi_tlb.h, 599
SBI_TLB_SFENCE_VMA
sbi_tlb.h, 598
SBI_TLB_SFENCE_VMA_ASID
sbi_tlb.h, 598
sbi_tlb_type
sbi_tlb.h, 598
SBI_TLB_TYPE_MAX
sbi_tlb.h, 598
sbi_trap.c
sbi_trap_aia_irq, 1095
sbi_trap_error, 1095
sbi_trap_error_one, 1096
sbi_trap_handler, 1096
sbi_trap_nonaia_irq, 1097
sbi_trap_redirect, 1098
sbi_trap.h
sbi_mstatus_prev_mode, 607
sbi_regs_from_virt, 607
sbi_regs_gva, 607
SBI_TRAP_CONTEXT_SIZE, 603
sbi_trap_get_context, 608
sbi_trap_handler, 608
SBI_TRAP_INFO_cause, 603
SBI_TRAP_INFO_gva, 603
SBI_TRAP_INFO_last, 603
SBI_TRAP_INFO_OFFSET, 603
SBI_TRAP_INFO_SIZE, 603
SBI_TRAP_INFO_tinst, 603
SBI_TRAP_INFO_tval, 603
SBI_TRAP_INFO_tval2, 603
sbi_trap_redirect, 609
SBI_TRAP_REGS_a0, 603
SBI_TRAP_REGS_a1, 603
SBI_TRAP_REGS_a2, 603
SBI_TRAP_REGS_a3, 604
SBI_TRAP_REGS_a4, 604
SBI_TRAP_REGS_a5, 604
SBI_TRAP_REGS_a6, 604
SBI_TRAP_REGS_a7, 604
SBI_TRAP_REGS_gp, 604
SBI_TRAP_REGS_last, 604
SBI_TRAP_REGS_mepc, 604
SBI_TRAP_REGS_mstatus, 604
SBI_TRAP_REGS_mstatusH, 604
SBI_TRAP_REGS_OFFSET, 604
SBI_TRAP_REGS_ra, 604
SBI_TRAP_REGS_s0, 605
SBI_TRAP_REGS_s1, 605
SBI_TRAP_REGS_s10, 605
SBI_TRAP_REGS_s11, 605
SBI_TRAP_REGS_s2, 605
SBI_TRAP_REGS_s3, 605
SBI_TRAP_REGS_s4, 605
SBI_TRAP_REGS_s5, 605
SBI_TRAP_REGS_s6, 605
SBI_TRAP_REGS_s7, 605
SBI_TRAP_REGS_s8, 605
SBI_TRAP_REGS_s9, 605
SBI_TRAP_REGS_SIZE, 606
SBI_TRAP_REGS_sp, 606
SBI_TRAP_REGS_t0, 606
SBI_TRAP_REGS_t1, 606
SBI_TRAP_REGS_t2, 606
SBI_TRAP_REGS_t3, 606
SBI_TRAP_REGS_t4, 606
SBI_TRAP_REGS_t5, 606
SBI_TRAP_REGS_t6, 606
SBI_TRAP_REGS_tp, 606
SBI_TRAP_REGS_zero, 606
sbi_trap_set_context, 610
sbi_trap_aia_irq
sbi_trap.c, 1095
sbi_trap_context, 153
prev_context, 154
regs, 154
trap, 154
SBI_TRAP_CONTEXT_SIZE
sbi_trap.h, 603
sbi_trap_emulate_load
sbi_trap_ldst.c, 1105
sbi_trap_emulate_store
sbi_trap_ldst.c, 1106
sbi_trap_error
sbi_trap.c, 1095
sbi_trap_error_one
sbi_trap.c, 1096
sbi_trap_get_context
sbi_trap.h, 608
sbi_trap_handler
sbi_trap.c, 1096
sbi_trap.h, 608
sbi_trap_info, 154
cause, 154
gva, 154
tinst, 154
tval, 154
tval2, 154
SBI_TRAP_INFO_cause

sbi_trap.h, 603
SBI_TRAP_INFO_gva
sbi_trap.h, 603
SBI_TRAP_INFO_last
sbi_trap.h, 603
SBI_TRAP_INFO_OFFSET
sbi_trap.h, 603
SBI_TRAP_INFO_SIZE
sbi_trap.h, 603
SBI_TRAP_INFO_tinst
sbi_trap.h, 603
SBI_TRAP_INFO_tval
sbi_trap.h, 603
SBI_TRAP_INFO_tval2
sbi_trap.h, 603
sbi_trap_ld_emulator
sbi_trap_ldst.c, 1100
sbi_trap_ldst.c
sbi_ld_access_emulator, 1101
sbi_load_access_handler, 1101
sbi_misaligned_ld_emulator, 1102
sbi_misaligned_load_handler, 1102
sbi_misaligned_st_emulator, 1103
sbi_misaligned_store_handler, 1103
sbi_misaligned_tinst_fixup, 1104
sbi_st_access_emulator, 1104
sbi_store_access_handler, 1104
sbi_trap_emulate_load, 1105
sbi_trap_emulate_store, 1106
sbi_trap_ld_emulator, 1100
sbi_trap_st_emulator, 1100
sbi_trap_ldst.h
sbi_double_trap_handler, 614
sbi_load_access_handler, 614
sbi_misaligned_load_handler, 615
sbi_misaligned_store_handler, 615
sbi_misaligned_tinst_fixup, 616
sbi_misaligned_v_ld_emulator, 616
sbi_misaligned_v_st_emulator, 616
sbi_store_access_handler, 617
sbi_trap_nonaia_irq
sbi_trap.c, 1097
sbi_trap_redirect
sbi_trap.c, 1098
sbi_trap.h, 609
sbi_trap_regs, 155
a0, 156
a1, 156
a2, 156
a3, 156
a4, 156
a5, 156
a6, 156
a7, 156
gp, 156
mepc, 156
mstatus, 156
mstatusH, 156
ra, 156
s0, 157
s1, 157
s10, 157
s11, 157
s2, 157
s3, 157
s4, 157
s5, 157
s6, 157
s7, 157
s8, 157
s9, 157
sp, 158
t0, 158
t1, 158
t2, 158
t3, 158
t4, 158
t5, 158
t6, 158
tp, 158
zero, 158
SBI_TRAP_REGS_a0
sbi_trap.h, 603
SBI_TRAP_REGS_a1
sbi_trap.h, 603
SBI_TRAP_REGS_a2
sbi_trap.h, 603
SBI_TRAP_REGS_a3
sbi_trap.h, 604
SBI_TRAP_REGS_a4
sbi_trap.h, 604
SBI_TRAP_REGS_a5
sbi_trap.h, 604
SBI_TRAP_REGS_a6
sbi_trap.h, 604
SBI_TRAP_REGS_a7
sbi_trap.h, 604
SBI_TRAP_REGS_gp
sbi_trap.h, 604
SBI_TRAP_REGS_last
sbi_trap.h, 604
SBI_TRAP_REGS_mepc
sbi_trap.h, 604
SBI_TRAP_REGS_mstatus
sbi_trap.h, 604
SBI_TRAP_REGS_mstatusH
sbi_trap.h, 604
SBI_TRAP_REGS_OFFSET
sbi_trap.h, 604
SBI_TRAP_REGS_ra
sbi_trap.h, 604
SBI_TRAP_REGS_s0
sbi_trap.h, 605
SBI_TRAP_REGS_s1
sbi_trap.h, 605
SBI_TRAP_REGS_s10

sbi_trap.h, 605
SBI_TRAP_REGS_s11
sbi_trap.h, 605
SBI_TRAP_REGS_s2
sbi_trap.h, 605
SBI_TRAP_REGS_s3
sbi_trap.h, 605
SBI_TRAP_REGS_s4
sbi_trap.h, 605
SBI_TRAP_REGS_s5
sbi_trap.h, 605
SBI_TRAP_REGS_s6
sbi_trap.h, 605
SBI_TRAP_REGS_s7
sbi_trap.h, 605
SBI_TRAP_REGS_s8
sbi_trap.h, 605
SBI_TRAP_REGS_s9
sbi_trap.h, 605
SBI_TRAP_REGS_SIZE
sbi_trap.h, 606
SBI_TRAP_REGS_sp
sbi_trap.h, 606
SBI_TRAP_REGS_t0
sbi_trap.h, 606
SBI_TRAP_REGS_t1
sbi_trap.h, 606
SBI_TRAP_REGS_t2
sbi_trap.h, 606
SBI_TRAP_REGS_t3
sbi_trap.h, 606
SBI_TRAP_REGS_t4
sbi_trap.h, 606
SBI_TRAP_REGS_t5
sbi_trap.h, 606
SBI_TRAP_REGS_t6
sbi_trap.h, 606
SBI_TRAP_REGS_tp
sbi_trap.h, 606
SBI_TRAP_REGS_zero
sbi_trap.h, 606
sbi_trap_set_context
sbi_trap.h, 610
sbi_trap_st_emulator
sbi_trap_ldst.c, 1100
sbi_trap_v_ldst.c
sbi_misaligned_v_ld_emulator, 1107
sbi_misaligned_v_st_emulator, 1107
sbi_trigger_init
sbi_dbtr.c, 837
sbi_types.h
__aligned, 619
__always_inline, 619
__has_builtin, 619
__noreturn, 619
__packed, 619
array_size, 619
be16_t, 621
be32_t, 621
be64_t, 621
bool, 621
CLAMP, 619
container_of, 619
false, 619
int16_t, 621
int32_t, 621
le16_t, 621
le32_t, 621
le64_t, 621
likely, 620
MAX, 620
MIN, 620
NULL, 620
offsetof, 620
physical_addr_t, 621
physical_size_t, 621
ROUNDDOWN, 620
ROUNDUP, 620
s16, 621
s32, 621
s8, 621
size_t, 621
ssize_t, 621
STR, 620
true, 620
u16, 622
u32, 622
u8, 622
uint16_t, 622
uint32_t, 622
uint8_t, 622
uintptr_t, 622
ulong, 622
unlikely, 620
virtual_addr_t, 622
virtual_size_t, 622
XSTR, 620
sbi_unit_test.c
ANSI_COLOR_GREEN, 1126
ANSI_COLOR_RED, 1126
ANSI_COLOR_RESET, 1127
run_all_tests, 1127
run_test_suite, 1127
sbi_unit_tests, 1127
sbi_unit_test.h
run_all_tests, 624
sbi_unit_tests
sbi_unit_test.c, 1127
sbi_unpriv.c
DEFINE_UNPRIVILEGED_LOAD_FUNCTION, 1108
DEFINE_UNPRIVILEGED_STORE_FUNCTION, 1109
sbi_get_insn, 1109
sbi_unpriv.h

DECLARE_UNPRIVILEGED_LOAD_FUNCTION, 626
DECLARE_UNPRIVILEGED_STORE_FUNCTION, 626
sbi_get_insn, 626
sbi_update_hartindex_to_domain
sbi_domain.c, 855
sbi_domain.h, 367
sbi_version.h
OPENSBI_VERSION, 628
OPENSBI_VERSION_MAJOR, 628
OPENSBI_VERSION_MINOR, 628
sbi_zalloc
sbi_heap.h, 462
sbi_zalloc_from
sbi_heap.c, 952
sbi_heap.h, 463
SBIUNIT_TEST_SUITE
riscv_locks_test.c, 1116
sbi_bitmap_test.c, 1120
sbi_console_test.c, 1123
sbi_math_test.c, 1125
SCBRR_VALUE
renesas_scif.c, 1558
scif_base
renesas_scif.c, 1561
SCIF_FCR_RFRST
renesas_scif.c, 1558
SCIF_FCR_RST_ASSRT_RFTF
renesas_scif.c, 1558
SCIF_FCR_RST_NGATE_RFTF
renesas_scif.c, 1558
SCIF_FCR_TFRST
renesas_scif.c, 1558
SCIF_FSR_BRK
renesas_scif.c, 1558
SCIF_FSR_DR
renesas_scif.c, 1558
SCIF_FSR_ER
renesas_scif.c, 1558
SCIF_FSR_TDFE
renesas_scif.c, 1558
SCIF_FSR_TEND
renesas_scif.c, 1558
SCIF_FSR_TXD_CHK
renesas_scif.c, 1558
SCIF_REG_8BIT
renesas_scif.c, 1558
SCIF_REG_BRR
renesas_scif.c, 1558
SCIF_REG_FCR
renesas_scif.c, 1558
SCIF_REG_FSR
renesas_scif.c, 1559
SCIF_REG_FTDR
renesas_scif.c, 1559
SCIF_REG_LSR
renesas_scif.c, 1559
SCIF_REG_SCR
renesas_scif.c, 1559
SCIF_REG_SEMR
renesas_scif.c, 1559
SCIF_REG_SMR
renesas_scif.c, 1559
SCIF_SCR_RCV_TRN_DIS
renesas_scif.c, 1559
SCIF_SCR_RCV_TRN_EN
renesas_scif.c, 1559
SCIF_SCR_RE
renesas_scif.c, 1559
SCIF_SCR_TE
renesas_scif.c, 1559
SCIF_SEMR_MDDRS
renesas_scif.c, 1559
select
fdt_pmu_hw_event_select_map, 87
fdt_pmu_raw_event_counter_map, 87
select_mask
fdt_pmu_raw_event_counter_map, 87
semihosting.c
_semihosting_enabled, 1566
semihosting_console, 1566
semihosting_enabled, 1563
semihosting_errno, 1563
semihosting_getc, 1563
semihosting_infd, 1566
semihosting_init, 1564
semihosting_open, 1564
semihosting_outfd, 1567
semihosting_puts, 1565
semihosting_read, 1565
semihosting_trap, 1565
semihosting_write, 1566
SYSERRNO, 1562
SYSOPEN, 1562
SYSREAD, 1562
SYSREADC, 1563
SYSWRITE, 1563
SYSWRITEC, 1563
try_semihosting, 1567
semihosting.h
MODE_APPEND, 752
MODE_BINARY, 752
MODE_PLUS, 752
MODE_READ, 752
MODE_WRITE, 752
semihosting_enabled, 752
semihosting_init, 752
semihosting_open_mode, 752
semihosting_console
semihosting.c, 1566
semihosting_enabled
semihosting.c, 1563
semihosting.h, 752
semihosting_errno
semihosting.c, 1563

semihosting_getc
semihosting.c, 1563
semihosting_infd
semihosting.c, 1566
semihosting_init
semihosting.c, 1564
semihosting.h, 752
semihosting_open
semihosting.c, 1564
semihosting_open_mode
semihosting.h, 752
semihosting_outfd
semihosting.c, 1567
semihosting_puts
semihosting.c, 1565
semihosting_read
semihosting.c, 1565
semihosting_trap
semihosting.c, 1565
semihosting_write
semihosting.c, 1566
send_message_with_response
sbi_mpxy_channel, 139
send_message_without_response
sbi_mpxy_channel, 140
seq
mbox_xfer, 96
serial_cadence_init
fdt_serial_cadence.c, 1536
serial_cadence_match
fdt_serial_cadence.c, 1537
serial_gaisler_init
fdt_serial_gaisler.c, 1538
serial_gaisler_match
fdt_serial_gaisler.c, 1538
serial_htif_init
fdt_serial_htif.c, 1539
serial_htif_match
fdt_serial_htif.c, 1540
serial_litex_init
fdt_serial_litex.c, 1541
serial_litex_match
fdt_serial_litex.c, 1541
serial_renesas_scif_init
fdt_serial_renesas_scif.c, 1542
serial_renesas_scif_match
fdt_serial_renesas_scif.c, 1543
serial_shakti_init
fdt_serial_shakti.c, 1544
serial_shakti_match
fdt_serial_shakti.c, 1544
serial_sifive_init
fdt_serial_sifive.c, 1545
serial_sifive_match
fdt_serial_sifive.c, 1546
serial_uart8250_init
fdt_serial_uart8250.c, 1547
serial_uart8250_match
fdt_serial_uart8250.c, 1547
serial_xlnx_uartlite_init
fdt_serial_xlnx_uartlite.c, 1548
serial_xlnx_uartlite_match
fdt_serial_xlnx_uartlite.c, 1549
service_id
rpmi_message_args, 119
rpmi_message_header, 120
servicegroup_id
rpmi_message_header, 120
set
gpio_chip, 90
set_attribute
mbox_controller, 95
set_custom_addr
htif.c, 1596
SET_DBTR_BIT
riscv_dbtr.h, 191
set_hartid_cb
sbi_sse_cb_ops, 150
set_multi_letter_ext
fdt_helper.c, 1156
SET_RD
riscv_encoding.h, 271
set_reg
cadence-uart.c, 1533
gaisler-uart.c, 1552
litex-uart.c, 1556
renesas_scif.c, 1561
sifive-uart.c, 1572
uart8250.c, 1577
SET_TRIG_HW_INDEX
sbi_dbtr.h, 343
sg2042_mcu_reset
fdt_reset_sg2042_hwmon_mcu.c, 1521
sg2042_mcu_reset_check
fdt_reset_sg2042_hwmon_mcu.c, 1521
sg2042_mcu_reset_check_board
fdt_reset_sg2042_hwmon_mcu.c, 1521
sg2042_mcu_reset_device
fdt_reset_sg2042_hwmon_mcu.c, 1522
sg2042_mcu_reset_init
fdt_reset_sg2042_hwmon_mcu.c, 1521
sg2042_mcu_reset_match
fdt_reset_sg2042_hwmon_mcu.c, 1522
SH_MEW
riscv_encoding.h, 271
SH_RD
riscv_encoding.h, 271
SH_RS1
riscv_encoding.h, 271
SH_RS2
riscv_encoding.h, 271
SH_RS2C
riscv_encoding.h, 271
SH_VD
riscv_encoding.h, 272
SH_VIEW

riscv_encoding.h, 272
SH_VM
riscv_encoding.h, 272
SH_VS2
riscv_encoding.h, 272
SH_VSEW
riscv_encoding.h, 272
Shakti C-class SoC Platform, 57
shakti-uart.c
REG_BAUD, 1568
REG_CONTROL, 1568
REG_DELAY, 1568
REG_INT_EN, 1568
REG_IQ_CYCLES, 1568
REG_RX, 1568
REG_RX_THRES, 1568
REG_STATUS, 1568
REG_TX, 1568
shakti_console, 1569
shakti_uart_getc, 1568
shakti_uart_init, 1568
shakti_uart_putc, 1569
uart_base, 1569
UART_RX_FULL, 1568
UART_RX_NOT_EMPTY, 1568
UART_TX_FULL, 1568
shakti-uart.h
shakti_uart_init, 753
shakti_console
shakti-uart.c, 1569
shakti_uart_getc
shakti-uart.c, 1568
shakti_uart_init
shakti-uart.c, 1568
shakti-uart.h, 753
shakti_uart_putc
shakti-uart.c, 1569
SHIFT_CSR
riscv_encoding.h, 272
SHIFT_FUNCT3
riscv_encoding.h, 272
SHIFT_RIGHT
riscv_encoding.h, 272
shl
qdivrem.c, 1452
shmem
sbi_dbtr_hart_triggers_state, 125
shs_ptr_off
sbi_sse.c, 1063
SiFive FU540 SoC Platform, 59
sifive-uart.c
get_reg, 1571
set_reg, 1572
sifive_console, 1574
sifive_uart_getc, 1572
sifive_uart_init, 1572
sifive_uart_putc, 1573
uart_base, 1574
uart_baudrate, 1574
uart_in_freq, 1574
uart_min_clk_divisor, 1573
UART_REG_DIV, 1570
UART_REG_IE, 1570
UART_REG_IP, 1571
UART_REG_RXCTRL, 1571
UART_REG_RXFIFO, 1571
UART_REG_TXCTRL, 1571
UART_REG_TXFIFO, 1571
UART_RXCTRL_RXEN, 1571
UART_RXFIFO_DATA, 1571
UART_RXFIFO_EMPTY, 1571
UART_TXCTRL_TXEN, 1571
UART_TXFIFO_FULL, 1571
sifive-uart.h
sifive_uart_init, 755
sifive_clint_quirks
fdt_timer_mtimer.c, 1606
sifive_console
sifive-uart.c, 1574
SIFIVE_GPIO_BIT
fdt_gpio_sifive.c, 1183
sifive_gpio_direction_output
fdt_gpio_sifive.c, 1183
sifive_gpio_init
fdt_gpio_sifive.c, 1184
sifive_gpio_match
fdt_gpio_sifive.c, 1185
SIFIVE_GPIO_OUTEN
fdt_gpio_sifive.c, 1183
SIFIVE_GPIO_OUTVAL
fdt_gpio_sifive.c, 1183
SIFIVE_GPIO_PINS_DEF
fdt_gpio_sifive.c, 1183
SIFIVE_GPIO_PINS_MAX
fdt_gpio_sifive.c, 1183
SIFIVE_GPIO_PINS_MIN
fdt_gpio_sifive.c, 1183
sifive_gpio_set
fdt_gpio_sifive.c, 1184
sifive_i2c_adapter_poll
fdt_i2c_sifive.c, 1213
sifive_i2c_adapter_poll_busy
fdt_i2c_sifive.c, 1211
sifive_i2c_adapter_poll_tip
fdt_i2c_sifive.c, 1211
sifive_i2c_adapter_read
fdt_i2c_sifive.c, 1213
sifive_i2c_adapter_rxack
fdt_i2c_sifive.c, 1214
sifive_i2c_adapter_start
fdt_i2c_sifive.c, 1214
sifive_i2c_adapter_write
fdt_i2c_sifive.c, 1215
SIFIVE_I2C_CMD_ACK
fdt_i2c_sifive.c, 1211
SIFIVE_I2C_CMD_IACK

fdt_i2c_sifive.c, 1211
SIFIVE_I2C_CMD_RD
fdt_i2c_sifive.c, 1211
SIFIVE_I2C_CMD_STA
fdt_i2c_sifive.c, 1211
SIFIVE_I2C_CMD_STO
fdt_i2c_sifive.c, 1211
SIFIVE_I2C_CMD_WR
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_CR
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_CTR
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_CTR_EN
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_CTR_IEN
fdt_i2c_sifive.c, 1212
sifive_i2c_getreg
fdt_i2c_sifive.c, 1216
sifive_i2c_init
fdt_i2c_sifive.c, 1216
sifive_i2c_match
fdt_i2c_sifive.c, 1217
SIFIVE_I2C_PREHI
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_PRELO
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_READ_BIT
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_RXR
fdt_i2c_sifive.c, 1212
sifive_i2c_setreg
fdt_i2c_sifive.c, 1216
SIFIVE_I2C_SR
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_STATUS_AL
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_STATUS_BUSY
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_STATUS_IF
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_STATUS_RXACK
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_STATUS_TIP
fdt_i2c_sifive.c, 1212
SIFIVE_I2C_TXR
fdt_i2c_sifive.c, 1213
SIFIVE_I2C_WRITE_BIT
fdt_i2c_sifive.c, 1213
sifive_uart_getc
sifive-uart.c, 1572
sifive_uart_init
sifive-uart.c, 1572
sifive-uart.h, 755
sifive_uart_putc
sifive-uart.c, 1573
SIP_SSIP
riscv_encoding.h, 272
SIP_STIP
riscv_encoding.h, 272
size
aclint_mswi_data, 78
aplic_data, 80
fdt_reserve_entry, 88
imsic_regs, 93
plic_data, 98
plicsw_data, 98
plmt_data, 99
sbi_tlb_info, 153
size_dt_strings
fdt_header, 84
size_dt_struct
fdt_header, 84
size_t
sbi_types.h, 621
skip_regs_update
sbi_ecall_return, 133
sl
uu, 159
SLEEP_CMD
atcsmu.h, 764
small_const_nbits
sbi_bitmap.h, 305
smask
sbi_tlb_info, 153
smp_mb
riscv_barrier.h, 186
smp_rmb
riscv_barrier.h, 186
smp_wmb
riscv_barrier.h, 186
smq_base_get_two_u32
fdt_mailbox_rpmi_shmem.c, 1472
smq_rx
fdt_mailbox_rpmi_shmem.c, 1472
smq_tx
fdt_mailbox_rpmi_shmem.c, 1473
SMSTATEEN0_AIA
riscv_encoding.h, 272
SMSTATEEN0_AIA_SHIFT
riscv_encoding.h, 272
SMSTATEEN0_CONTEXT
riscv_encoding.h, 272
SMSTATEEN0_CONTEXT_SHIFT
riscv_encoding.h, 272
SMSTATEEN0_CS
riscv_encoding.h, 272
SMSTATEEN0_CS_SHIFT
riscv_encoding.h, 273
SMSTATEEN0_FCSR
riscv_encoding.h, 273
SMSTATEEN0_FCSR_SHIFT
riscv_encoding.h, 273
SMSTATEEN0_HSENVCFG
riscv_encoding.h, 273
SMSTATEEN0_HSENVCFG_SHIFT

riscv_encoding.h, 273
SMSTATEEN0_IMSIC
riscv_encoding.h, 273
SMSTATEEN0_IMSIC_SHIFT
riscv_encoding.h, 273
SMSTATEEN0_SVSLCT
riscv_encoding.h, 273
SMSTATEEN0_SVSLCT_SHIFT
riscv_encoding.h, 273
SMSTATEEN_MAX_COUNT
riscv_encoding.h, 273
SMSTATEEN_STATEN
riscv_encoding.h, 273
SMSTATEEN_STATEN_SHIFT
riscv_encoding.h, 273
smu
fdt_reset_atcwdt200.c, 1510
smu_data, 158
addr, 159
smu_set_command
atcsmu.c, 1587
atcsmu.h, 764
smu_set_reset_vector
atcsmu.c, 1587
atcsmu.h, 764
smu_set_wakeup_events
atcsmu.c, 1588
atcsmu.h, 765
smu_support_sleep_mode
atcsmu.c, 1588
atcsmu.h, 765
sp
sbi_trap_regs, 158
Spike Simulator Platform, 63
spin_lock
riscv_locks.c, 791
riscv_locks.h, 300
spin_lock_check
riscv_locks.c, 792
riscv_locks.h, 301
SPIN_LOCK_INIT
riscv_locks.h, 300
SPIN_LOCK_INITIALIZER
riscv_locks.h, 300
spin_lock_test
riscv_locks_test.c, 1116
spin_lock_unlocked
riscv_locks.c, 793
spin_trylock
riscv_locks.c, 793
riscv_locks.h, 302
spin_trylock_fail
riscv_locks_test.c, 1117
spin_trylock_success
riscv_locks_test.c, 1117
spin_unlock
riscv_locks.c, 794
riscv_locks.h, 302
srst_available
sbi_ecall_srst.c, 893
sse_enabled_event_lock
sbi_sse.c, 1046
sse_enabled_event_unlock
sbi_sse.c, 1046
sse_event_add_to_list
sbi_sse.c, 1047
sse_event_check_inject
sbi_sse.c, 1047
sse_event_complete
sbi_sse.c, 1048
sse_event_count_init
sbi_sse.c, 1048
sse_event_disable
sbi_sse.c, 1049
sse_event_enable
sbi_sse.c, 1049
sse_event_get
sbi_sse.c, 1050
sse_event_id
sbi_mpxy_channel_attrs, 141
sse_event_init
sbi_sse.c, 1051
sse_event_inject
sbi_sse.c, 1051
sse_event_invoke_cb
sbi_sse.c, 1034
sse_event_is_global
sbi_sse.c, 1052
sse_event_is_local
sbi_sse.c, 1052
sse_event_is_ready
sbi_sse.c, 1053
sse_event_pending
sbi_sse.c, 1053
sse_event_put
sbi_sse.c, 1054
sse_event_register
sbi_sse.c, 1054
sse_event_remove_from_list
sbi_sse.c, 1055
sse_event_resume
sbi_sse.c, 1055
sse_event_set_attr
sbi_sse.c, 1056
sse_event_set_attr_check
sbi_sse.c, 1056
sse_event_set_hart_id_check
sbi_sse.c, 1057
sse_event_set_pending
sbi_sse.c, 1057
sse_event_set_state
sbi_sse.c, 1058
sse_event_state
sbi_sse.c, 1058
sse_event_unregister
sbi_sse.c, 1058

sse_get_global_event
sbi_sse.c, 1059
sse_get_hart_state
sbi_sse.c, 1059
sse_get_hart_state_ptr
sbi_sse.c, 1034
sse_global_init
sbi_sse.c, 1060
sse_inject_event
sbi_sse.c, 1060
sse_inject_fifo_mem_off
sbi_sse.c, 1063
sse_inject_fifo_off
sbi_sse.c, 1063
sse_interrupted_flags
sbi_sse.c, 1061
sse_ipi_inject_event
sbi_sse.c, 1063
sse_ipi_inject_ops
sbi_sse.c, 1063
sse_ipi_inject_process
sbi_sse.c, 1061
sse_ipi_inject_send
sbi_sse.c, 1061
sse_local_init
sbi_sse.c, 1062
sse_set_hart_state_ptr
sbi_sse.c, 1035
sse_thishart_state_ptr
sbi_sse.c, 1035
sse_write_attrs
sbi_sse.c, 1062
ssize_t
sbi_types.h, 621
SSTATUS32_SD
riscv_encoding.h, 273
SSTATUS64_SD
riscv_encoding.h, 273
SSTATUS64_UXL
riscv_encoding.h, 273
SSTATUS_FS
riscv_encoding.h, 274
SSTATUS_MXR
riscv_encoding.h, 274
SSTATUS_SD
riscv_encoding.h, 274
SSTATUS_SIE
riscv_encoding.h, 274
SSTATUS_SPIE
riscv_encoding.h, 274
SSTATUS_SPIE_SHIFT
riscv_encoding.h, 274
SSTATUS_SPP
riscv_encoding.h, 274
SSTATUS_SPP_SHIFT
riscv_encoding.h, 274
SSTATUS_SUM
riscv_encoding.h, 274
SSTATUS_VS
riscv_encoding.h, 274
SSTATUS_XS
riscv_encoding.h, 274
starfive_gpio_direction_output
fdt_gpio_starfive.c, 1186
starfive_gpio_init
fdt_gpio_starfive.c, 1186
STARFIVE_GPIO_MASK
fdt_gpio_starfive.c, 1186
starfive_gpio_match
fdt_gpio_starfive.c, 1187
STARFIVE_GPIO_OUTVAL
fdt_gpio_starfive.c, 1186
STARFIVE_GPIO_PINS_DEF
fdt_gpio_starfive.c, 1186
STARFIVE_GPIO_REG_SHIFT_MASK
fdt_gpio_starfive.c, 1186
starfive_gpio_set
fdt_gpio_starfive.c, 1187
STARFIVE_GPIO_SHIFT_BITS
fdt_gpio_starfive.c, 1186
start
sbi_tlb_info, 153
start_addr_hi
rpmi_hsm_hart_start_req, 116
start_addr_lo
rpmi_hsm_hart_start_req, 116
start_index
rpmi_cppc_hart_list_req, 109
rpmi_hsm_get_hart_list_req, 113
rpmi_hsm_get_susp_types_req, 115
state
sbi_dbtr_trigger, 127
status
rpmi_base_get_platform_info_resp, 101
rpmi_clock_get_attributes_resp, 102
rpmi_clock_get_config_resp, 103
rpmi_clock_get_num_clocks_resp, 103
rpmi_clock_get_rate_resp, 104
rpmi_clock_get_supported_rates_resp, 105
rpmi_clock_set_config_resp, 106
rpmi_clock_set_rate_resp, 106
rpmi_cppc_get_fastchan_offset_resp, 107
rpmi_cppc_get_fastchan_region_resp, 108
rpmi_cppc_hart_list_resp, 109
rpmi_cppc_probe_resp, 110
rpmi_cppc_read_reg_resp, 111
rpmi_cppc_write_reg_resp, 112
rpmi_enable_notification_resp, 112
rpmi_hsm_get_hart_list_resp, 113
rpmi_hsm_get_hart_status_resp, 114
rpmi_hsm_get_susp_info_resp, 115
rpmi_hsm_get_susp_types_resp, 116
rpmi_hsm_hart_start_resp, 117
rpmi_hsm_hart_stop_resp, 117
rpmi_hsm_hart_susp_resp, 118
rpmi_sysrst_get_reset_attributes_resp, 120

rpmi_syssusp_get_attr_resp, 121
rpmi_syssusp_suspend_resp, 122
status_code
rpmi_base_get_attributes_resp, 101
STR
sbi_types.h, 620
strchr
libfdt_env.h, 1429
strcmp
libfdt_env.h, 1430
strcpy
libfdt_env.h, 1430
STRINGIFY
sbi_const.h, 335
strlen
libfdt_env.h, 1430
strncmp
libfdt_env.h, 1430
strncpy
libfdt_env.h, 1430
strnlen
libfdt_env.h, 1430
strrchr
libfdt_env.h, 1430
sub_return_test
riscv_atomic_test.c, 1114
sunxi_wdt_base
fdt_reset_sunxi_wdt.c, 1524
sunxi_wdt_reset
fdt_reset_sunxi_wdt.c, 1524
sunxi_wdt_reset_init
fdt_reset_sunxi_wdt.c, 1524
sunxi_wdt_reset_match
fdt_reset_sunxi_wdt.c, 1524
sunxi_wdt_system_reset
fdt_reset_sunxi_wdt.c, 1524
sunxi_wdt_system_reset_check
fdt_reset_sunxi_wdt.c, 1524
supported_events
sbi_sse.c, 1064
susp_available
sbi_ecall_susp.c, 897
susp_type
rpmi_syssusp_get_attr_req, 121
suspend_dev
sbi_system.c, 1074
suspend_param
sbi_cpu_idle_state, 124
suspend_type
rpmi_hsm_get_susp_info_req, 114
rpmi_hsm_hart_susp_req, 118
rpmi_syssusp_suspend_req, 122
sw_flags
fdt_sw.c, 1347
swap_region
sbi_domain.c, 855
switch_eventsstate
sbi_mpxy_channel, 140
switch_to_next_domain_context
sbi_domain_context.c, 860
switchcase_csr_read
riscv_asm.c, 776
switchcase_csr_read_16
riscv_asm.c, 776
switchcase_csr_read_2
riscv_asm.c, 776
switchcase_csr_read_32
riscv_asm.c, 776
switchcase_csr_read_4
riscv_asm.c, 777
switchcase_csr_read_64
riscv_asm.c, 777
switchcase_csr_read_8
riscv_asm.c, 777
switchcase_csr_write
riscv_asm.c, 777
switchcase_csr_write_16
riscv_asm.c, 777
switchcase_csr_write_2
riscv_asm.c, 777
switchcase_csr_write_32
riscv_asm.c, 777
switchcase_csr_write_4
riscv_asm.c, 778
switchcase_csr_write_64
riscv_asm.c, 778
switchcase_csr_write_8
riscv_asm.c, 778
switchcase_hpm
sbi_emulate_csr.c, 902
switchcase_hpm_16
sbi_emulate_csr.c, 902
switchcase_hpm_2
sbi_emulate_csr.c, 903
switchcase_hpm_4
sbi_emulate_csr.c, 903
switchcase_hpm_8
sbi_emulate_csr.c, 903
sync
sbi_ipi_event_ops, 137
syscon_do_poweroff
fdt_reset_syscon.c, 1526
syscon_do_reboot
fdt_reset_syscon.c, 1526
SYSCON_ENDIAN_BIG
fdt_regmap_syscon.c, 1495
SYSCON_ENDIAN_LITTLE
fdt_regmap_syscon.c, 1495
SYSCON_ENDIAN_MAX
fdt_regmap_syscon.c, 1495
SYSCON_ENDIAN_NATIVE
fdt_regmap_syscon.c, 1495
syscon_poweroff
fdt_reset_syscon.c, 1529
syscon_poweroff_check
fdt_reset_syscon.c, 1526

syscon_poweroff_match
fdt_reset_syscon.c, 1529
syscon_reboot
fdt_reset_syscon.c, 1529
syscon_reboot_check
fdt_reset_syscon.c, 1526
syscon_reboot_match
fdt_reset_syscon.c, 1529
syscon_regmap_endian
fdt_regmap_syscon.c, 1495
syscon_reset_exec
fdt_reset_syscon.c, 1527
syscon_reset_get
fdt_reset_syscon.c, 1527
syscon_reset_init
fdt_reset_syscon.c, 1528
SYSERRNO
semihosting.c, 1562
SYSOPEN
semihosting.c, 1562
SYSREAD
semihosting.c, 1562
SYSREADC
semihosting.c, 1563
sysreset_ctx
fdt_reset_rpmi.c, 1519
syssusp_ctx
fdt_suspend_rpmi.c, 1586
system_opcode_insn
sbi_illegal_insn.c, 968
system_reset
sbi_system_reset_device, 151
system_reset_allowed
sbi_domain, 129
system_reset_check
sbi_system_reset_device, 151
system_suspend
sbi_system_suspend_device, 151
system_suspend_allowed
sbi_domain, 129
system_suspend_check
sbi_system_suspend_device, 151
SYSWRITE
semihosting.c, 1563
SYSWRITEC
semihosting.c, 1563
SZREG
riscv_asm.h, 169

T-HEAD C9xx Series Processors, 65
t0
sbi_trap_regs, 158
t1
sbi_trap_regs, 158
t2
sbi_trap_regs, 158
t3
sbi_trap_regs, 158
t4
sbi_trap_regs, 158
t5
sbi_trap_regs, 158
t6
sbi_trap_regs, 158
tag
fdt_node_header, 85
fdt_property, 88
tail
sbi_fifo, 133
targets_mmode
aplic_data, 80
imsic_data, 93
tdata1
sbi_dbtr_data_msg, 125
sbi_dbtr_trigger, 127
TDATA1_GET_TYPE
sbi_dbtr.h, 343
tdata2
sbi_dbtr_data_msg, 125
sbi_dbtr_trigger, 127
tdata3
sbi_dbtr_data_msg, 125
sbi_dbtr_trigger, 127
test_atomic
riscv_atomic_test.c, 1115
test_console_begin
sbi_console_test.c, 1123
test_console_buf
sbi_console_test.c, 1124
TEST_CONSOLE_BUF_LEN
sbi_console_test.c, 1122
test_console_buf_pos
sbi_console_test.c, 1124
test_console_dev
sbi_console_test.c, 1124
test_console_end
sbi_console_test.c, 1123
test_console_lock
sbi_console_test.c, 1124
test_console_putc
sbi_console_test.c, 1124
test_lock
riscv_locks_test.c, 1118
thead_aclint_quirks
fdt_timer_mtimer.c, 1606
thead_clint_quirks
fdt_timer_mtimer.c, 1606
THEAD_PLIC_CTRL_REG
plic.c, 1251
TICKET_SHIFT
riscv_locks.h, 300
time_cmp
plmt_data, 99
time_delta_computed
aclint_mtimer_data, 79
time_delta_off
sbi_timer.c, 1082

time_delta_reference
aclint_mtimer_data, 79
time_rd
aclint_mtimer_data, 79
time_val
plmt_data, 99
time_wr
aclint_mtimer_data, 79
timer_dev
sbi_timer.c, 1082
timer_event_start
sbi_timer_device, 152
timer_event_stop
sbi_timer_device, 152
timer_freq
plmt_data, 99
sbi_timer_device, 152
timer_init
sbi_platform_operations, 145
timer_mtimer_cold_init
fdt_timer_mtimer.c, 1605
timer_mtimer_match
fdt_timer_mtimer.c, 1606
timer_plmt_match
fdt_timer_plmt.c, 1608
timer_value
sbi_timer_device, 152
tinst
sbi_trap_info, 154
tlb_entry_local_process
sbi_tlb.c, 1088
tlb_entry_process
sbi_tlb.c, 1089
tlb_event
sbi_tlb.c, 1094
tlb_fifo_mem_off
sbi_tlb.c, 1094
tlb_fifo_off
sbi_tlb.c, 1094
tlb_flush_all
sbi_tlb.c, 1090
tlb_ops
sbi_tlb.c, 1094
tlb_process
sbi_tlb.c, 1090
tlb_process_once
sbi_tlb.c, 1091
tlb_range_check
sbi_tlb.c, 1091
tlb_range_flush_limit
sbi_tlb.c, 1094
tlb_sync
sbi_tlb.c, 1092
tlb_sync_off
sbi_tlb.c, 1094
tlb_type_to_pmu_fw_event
sbi_tlb.c, 1094
tlb_update
sbi_tlb.c, 1092
tlb_update_cb
sbi_tlb.c, 1093
tmp0
sbi_scratch, 149
to_gpio_chip
gpio.h, 671
to_i2c_adapter
i2c.h, 682
to_mbox_chan
mailbox.h, 704
to_mbox_controller
mailbox.h, 704
to_regmap
regmap.h, 740
to_srvgrp_chan
fdt_mailbox_rpmi_shmem.c, 1466
to_syscon_regmap
fdt_regmap_syscon.c, 1494
to_system_reset_device
sbi_system.h, 587
TOHOST_CMD
htif.c, 1591
token
rpmi_message_header, 120
TOPI_IID_MASK
riscv_encoding.h, 274
TOPI_IID_SHIFT
riscv_encoding.h, 274
TOPI_IPRIO_MASK
riscv_encoding.h, 274
total_ctrs
sbi_pmu.c, 1028
total_trigs
sbi_dbtr_hart_triggers_state, 125
totalsize
fdt_header, 84
tp
sbi_trap_regs, 158
transition_latency
rpmi_clock_get_attributes_resp, 102
trap
sbi_trap_context, 154
trap_context
sbi_scratch, 149
triggers
sbi_dbtr_hart_triggers_state, 126
true
sbi_types.h, 620
truly_illegal_insn
sbi_illegal_insn.c, 969
try_semihosting
semihosting.c, 1567
tstate
sbi_dbtr_data_msg, 125
tval
sbi_trap_info, 154
tval2

sbi_trap_info, 154
tx
mbox_xfer, 96
tx_endian_words
rpmi_message_args, 119
tx_len
mbox_xfer, 96
tx_timeout
mbox_xfer, 96
type
rpmi_message_args, 119
sbi_tlb_info, 153
type_mask
sbi_dbtr_trigger, 127
types
rpmi_hsm_get_susp_types_resp, 116
types.h
_QUAD_HIGHWORD, 1449
_QUAD_LOWWORD, 1449
quad_t, 1449
u_long, 1449
u_quad_t, 1449

U-Boot as a payload to OpenSBI, 35
u16
sbi_types.h, 622
u32
sbi_types.h, 622
u8
sbi_types.h, 622
u_long
types.h, 1449
u_quad_t
types.h, 1449
uart8250.c
get_reg, 1577
set_reg, 1577
uart8250_base, 1579
uart8250_baudrate, 1579
uart8250_console, 1579
uart8250_getc, 1577
uart8250_in_freq, 1579
uart8250_init, 1578
uart8250_putc, 1578
uart8250_reg_shift, 1579
uart8250_reg_width, 1579
UART_DLL_OFFSET, 1575
UART_DLM_OFFSET, 1575
UART_FCR_OFFSET, 1575
UART_IER_OFFSET, 1575
UART_IIR_OFFSET, 1576
UART_LCR_OFFSET, 1576
UART_LSR_BI, 1576
UART_LSR_BRK_ERROR_BITS, 1576
UART_LSR_DR, 1576
UART_LSR_FE, 1576
UART_LSR_FIFOE, 1576
UART_LSR_OE, 1576
UART_LSR_OFFSET, 1576
UART_LSR_PE, 1576
UART_LSR_TEMT, 1576
UART_LSR_THRE, 1576
UART_MCR_OFFSET, 1576
UART_MDR1_OFFSET, 1576
UART_MSR_OFFSET, 1576
UART_RBR_OFFSET, 1577
UART_SCR_OFFSET, 1577
UART_THR_OFFSET, 1577
uart8250.h
uart8250_init, 757
uart8250_base
uart8250.c, 1579
uart8250_baudrate
uart8250.c, 1579
uart8250_console
uart8250.c, 1579
uart8250_getc
uart8250.c, 1577
uart8250_in_freq
uart8250.c, 1579
uart8250_init
uart8250.c, 1578
uart8250.h, 757
uart8250_putc
uart8250.c, 1578
uart8250_reg_shift
uart8250.c, 1579
uart8250_reg_width
uart8250.c, 1579
uart_base
cadence-uart.c, 1534
gaisler-uart.c, 1553
litex-uart.c, 1556
shakti-uart.c, 1569
sifive-uart.c, 1574
uart_baudrate
cadence-uart.c, 1534
sifive-uart.c, 1574
UART_BRGR_CD_CLKDIVISOR
cadence-uart.c, 1530
UART_CSR_REMPTY
cadence-uart.c, 1530
UART_CSR_TFUL
cadence-uart.c, 1531
UART_CTRL_DB
gaisler-uart.c, 1550
UART_CTRL_FL
gaisler-uart.c, 1550
UART_CTRL_IE
xlnx-uartlite.c, 1580
UART_CTRL_OFFSET
xlnx-uartlite.c, 1580
UART_CTRL_RE
gaisler-uart.c, 1550
UART_CTRL_RST_RX
xlnx-uartlite.c, 1580
UART_CTRL_RST_TX

xlnx-uartlite.c, 1581
UART_CTRL_RXDIS
cadence-uart.c, 1531
UART_CTRL_RXEN
cadence-uart.c, 1531
UART_CTRL_RXRES
cadence-uart.c, 1531
UART_CTRL_TE
gaisler-uart.c, 1550
UART_CTRL_TXDIS
cadence-uart.c, 1531
UART_CTRL_TXEN
cadence-uart.c, 1531
UART_CTRL_TXRES
cadence-uart.c, 1531
UART_DATA_DATA
gaisler-uart.c, 1550
UART_DLL_OFFSET
uart8250.c, 1575
UART_DLM_OFFSET
uart8250.c, 1575
UART_EV_RX
litex-uart.c, 1554
UART_FCR_OFFSET
uart8250.c, 1575
UART_IER_OFFSET
uart8250.c, 1575
UART_IIR_OFFSET
uart8250.c, 1576
uart_in_freq
cadence-uart.c, 1534
sifive-uart.c, 1574
UART_LCR_OFFSET
uart8250.c, 1576
UART_LSR_BI
uart8250.c, 1576
UART_LSR_BRK_ERROR_BITS
uart8250.c, 1576
UART_LSR_DR
uart8250.c, 1576
UART_LSR_FE
uart8250.c, 1576
UART_LSR_FIFOE
uart8250.c, 1576
UART_LSR_OE
uart8250.c, 1576
UART_LSR_OFFSET
uart8250.c, 1576
UART_LSR_PE
uart8250.c, 1576
UART_LSR_TEMT
uart8250.c, 1576
UART_LSR_THRE
uart8250.c, 1576
UART_MCR_OFFSET
uart8250.c, 1576
UART_MDR1_OFFSET
uart8250.c, 1576
uart_min_clk_divisor
cadence-uart.c, 1533
sifive-uart.c, 1573
UART_MODE_PAR_NONE
cadence-uart.c, 1531
UART_MSR_OFFSET
uart8250.c, 1576
UART_RBR_OFFSET
uart8250.c, 1577
UART_REG_BDIVR
cadence-uart.c, 1531
UART_REG_BRGR
cadence-uart.c, 1531
UART_REG_CSR
cadence-uart.c, 1531
UART_REG_CTRL
cadence-uart.c, 1531
gaisler-uart.c, 1550
UART_REG_DATA
gaisler-uart.c, 1550
UART_REG_DIV
sifive-uart.c, 1570
UART_REG_EV_ENABLE
litex-uart.c, 1554
UART_REG_EV_PENDING
litex-uart.c, 1554
UART_REG_EV_STATUS
litex-uart.c, 1554
UART_REG_IDR
cadence-uart.c, 1531
UART_REG_IE
sifive-uart.c, 1570
UART_REG_IP
sifive-uart.c, 1571
UART_REG_MODE
cadence-uart.c, 1531
UART_REG_RFIFO_TFIFO
cadence-uart.c, 1531
UART_REG_RXCTRL
sifive-uart.c, 1571
UART_REG_RXEMPTY
litex-uart.c, 1554
UART_REG_RXFIFO
sifive-uart.c, 1571
UART_REG_RXTX
litex-uart.c, 1554
UART_REG_SCALER
gaisler-uart.c, 1550
UART_REG_STATUS
gaisler-uart.c, 1550
UART_REG_TXCTRL
sifive-uart.c, 1571
UART_REG_TXFIFO
sifive-uart.c, 1571
UART_REG_TXFULL
litex-uart.c, 1554
UART_RX_FULL
shakti-uart.c, 1568

UART_RX_NOT_EMPTY
shakti-uart.c, 1568
UART_RX_OFFSET
xlnx-uartlite.c, 1581
UART_RXCTRL_RXEN
sifive-uart.c, 1571
UART_RXFIFO_DATA
sifive-uart.c, 1571
UART_RXFIFO_EMPTY
sifive-uart.c, 1571
UART_SCR_OFFSET
uart8250.c, 1577
UART_STATUS_DATAREADY
gaisler-uart.c, 1550
UART_STATUS_FIFOFULL
gaisler-uart.c, 1550
UART_STATUS_FRAME
xlnx-uartlite.c, 1581
UART_STATUS_IE
xlnx-uartlite.c, 1581
UART_STATUS_OFFSET
xlnx-uartlite.c, 1581
UART_STATUS_OVERRUN
xlnx-uartlite.c, 1581
UART_STATUS_PARITY
xlnx-uartlite.c, 1581
UART_STATUS_RXFULL
xlnx-uartlite.c, 1581
UART_STATUS_RXVALID
xlnx-uartlite.c, 1581
UART_STATUS_TXEMPTY
xlnx-uartlite.c, 1581
UART_STATUS_TXFULL
xlnx-uartlite.c, 1581
UART_THR_OFFSET
uart8250.c, 1577
UART_TX_FULL
shakti-uart.c, 1568
UART_TX_OFFSET
xlnx-uartlite.c, 1581
UART_TXCTRL_TXEN
sifive-uart.c, 1571
UART_TXFIFO_FULL
sifive-uart.c, 1571
udivdi3.c
__FBSDID, 1459
__udivdi3, 1459
uint16_t
sbi_types.h, 622
UINT32_MAX
libfdt_env.h, 1430
uint32_t
sbi_types.h, 622
uint8_t
sbi_types.h, 622
UINT_MAX
libfdt_env.h, 1430
uintptr_t
sbi_types.h, 622
UL
sbi_const.h, 335
ul
uu, 159
ULL
sbi_const.h, 335
ulong
sbi_types.h, 622
umoddi3.c
__FBSDID, 1460
__umoddi3, 1460
unlikely
sbi_types.h, 620
unregister_cb
sbi_sse_cb_ops, 150
update
sbi_ipi_event_ops, 137
update_bit
sbi_dbtr.c, 837
uq
uu, 159
USE_TBUF
sbi_console.c, 801
uu, 159
q, 159
sl, 159
ul, 159
uq, 159

va_arg
sbi_console.c, 801
va_end
sbi_console.c, 801
va_list
sbi_console.c, 802
va_start
sbi_console.c, 801
value
sbi_ecall_return, 133
VD_MASK
riscv_encoding.h, 274
vendor_ext_check
sbi_platform_operations, 145
vendor_ext_provider
sbi_platform_operations, 145
version
fdt_header, 84
fw_dynamic_info, 89
VIEW_MASK
riscv_encoding.h, 275
virtual_addr_t
sbi_types.h, 622
virtual_size_t
sbi_types.h, 622
VLMUL_MASK
riscv_encoding.h, 275
VM_MASK
riscv_encoding.h, 275

vmid
sbi_tlb_info, 153
VS2_MASK
riscv_encoding.h, 275
VSEW_MASK
riscv_encoding.h, 275

wait_for_coldboot
sbi_init.c, 980
wake_coldboot_harts
sbi_init.c, 981
WAKEUP_CMD
atcsmu.h, 764
wakeup_latency_us
rpmi_hsm_get_susp_info_resp, 115
sbi_cpu_idle_state, 124
warm_init
sbi_irqchip_device, 138
sbi_timer_device, 152
warmboot_addr
sbi_scratch, 149
wdt_addr
fdt_reset_atcwdt200.c, 1510
WDT_EN
fdt_reset_atcwdt200.c, 1508
WDT_KEY_VAL
fdt_reset_sunxi_wdt.c, 1523
WDT_MODE_REG
fdt_reset_sunxi_wdt.c, 1523
WDT_SOFT_RST_EN
fdt_reset_sunxi_wdt.c, 1523
WDT_SOFT_RST_REG
fdt_reset_sunxi_wdt.c, 1523
wfi
riscv_asm.h, 169
wmb
riscv_barrier.h, 186
WREN_REG
fdt_reset_atcwdt200.c, 1509
write
i2c_adapter, 92
write_attributes
sbi_mpxy_channel, 140
writeb
riscv_io.h, 296
writeb_relaxed
riscv_io.h, 296
writel
riscv_io.h, 296
writel_relaxed
riscv_io.h, 296
writeq
riscv_io.h, 296
writeq_relaxed
riscv_io.h, 296
writew
riscv_io.h, 296
writew_relaxed
riscv_io.h, 297
Writing tests for OpenSBI, 23

xfer
mbox_controller, 95
xfer_next_seq
mbox_controller, 95
xlate
fdt_gpio, 83
fdt_mailbox, 85
xlnx-uartlite.c
UART_CTRL_IE, 1580
UART_CTRL_OFFSET, 1580
UART_CTRL_RST_RX, 1580
UART_CTRL_RST_TX, 1581
UART_RX_OFFSET, 1581
UART_STATUS_FRAME, 1581
UART_STATUS_IE, 1581
UART_STATUS_OFFSET, 1581
UART_STATUS_OVERRUN, 1581
UART_STATUS_PARITY, 1581
UART_STATUS_RXFULL, 1581
UART_STATUS_RXVALID, 1581
UART_STATUS_TXEMPTY, 1581
UART_STATUS_TXFULL, 1581
UART_TX_OFFSET, 1581
xlnx_uartlite_base, 1582
xlnx_uartlite_console, 1582
xlnx_uartlite_getc, 1581
xlnx_uartlite_init, 1581
xlnx_uartlite_putc, 1582
xlnx_uartlite.h
xlnx_uartlite_init, 758
xlnx_uartlite_base
xlnx-uartlite.c, 1582
xlnx_uartlite_console
xlnx-uartlite.c, 1582
xlnx_uartlite_getc
xlnx-uartlite.c, 1581
xlnx_uartlite_init
xlnx-uartlite.c, 1581
xlnx_uartlite.h, 758
xlnx_uartlite_putc
xlnx-uartlite.c, 1582
XSTR
sbi_types.h, 620

zero
sbi_trap_regs, 158