RISC-V OpenSBI

v0.8

Generated by Doxygen 1.8.17

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

2 OpenSBI Contribution Guideline 5

3 OpenSBI Platform Support Guideline 7

4 OpenSBI Platform Requirements 9

5 OpenSBI Library Usage 11

6 OpenSBI Platform Firmwares 13

7 OpenSBI Firmware with Dynamic Information (FW_DYNAMIC) 15

8 OpenSBI Firmware with Jump Address (FW_JUMP) 17

9 OpenSBI Firmware with Payload (FW_PAYLOAD) 19

10 Linux as a direct payload to OpenSBI 21

11 U-Boot as a payload to OpenSBI 23

12 Andes AE350 SoC Platform 25

13 Ariane FPGA SoC Platform 27

14 OpenPiton FPGA SoC Platform 29

15 Generic Platform 31

16 Nuclei UX600 Platform 33

17 OpenSBI Supported Platforms 35

18 QEMU RISC-V Virt Machine Platform 37

19 Shakti C-class SoC Platform 39

20 SiFive FU540 SoC Platform 41

21 Spike Simulator Platform 43

22 T-HEAD C910 Processor 45

23 Data Structure Index 47

23.1 Data Structures 47

24 File Index 49

24.1 File List 49

25 Data Structure Documentation 53

25.1 atomic_t Struct Reference 53

25.1.1 Field Documentation . . . 53
25.1.1.1 counter . . . 53
25.2 clint_data Struct Reference . . . 53
25.2.1 Field Documentation . . . 53
25.2.1.1 addr . . . 54
25.2.1.2 first_hartid . . . 54
25.2.1.3 hart_count . . . 54
25.2.1.4 has_64bit_mmio . . . 54
25.2.1.5 ipi . . . 54
25.2.1.6 time_cmp . . . 54
25.2.1.7 time_delta . . . 54
25.2.1.8 time_delta_computed . . . 54
25.2.1.9 time_delta_reference . . . 54
25.2.1.10 time_rd . . . 54
25.2.1.11 time_val . . . 54
25.2.1.12 time_wr . . . 54
25.3 fdt_header Struct Reference . . . 55
25.3.1 Field Documentation . . . 55
25.3.1.1 boot_cpuid_phys . . . 55
25.3.1.2 last_comp_version . . . 55
25.3.1.3 magic . . . 55
25.3.1.4 off_dt_strings . . . 55
25.3.1.5 off_dt_struct . . . 55
25.3.1.6 off_mem_rsvmap . . . 55
25.3.1.7 size_dt_strings . . . 55
25.3.1.8 size_dt_struct . . . 55
25.3.1.9 totalsize . . . 56
25.3.1.10 version . . . 56
25.4 fdt_ipi Struct Reference . . . 56
25.4.1 Field Documentation . . . 56
25.4.1.1 clear . . . 56
25.4.1.2 cold_init . . . 56
25.4.1.3 exit . . . 56
25.4.1.4 match_table . . . 56
25.4.1.5 send . . . 56
25.4.1.6 warm_init . . . 56
25.5 fdt_irqchip Struct Reference . . . 57
25.5.1 Field Documentation . . . 57
25.5.1.1 cold_init . . . 57
25.5.1.2 exit . . . 57
25.5.1.3 match_table . . . 57
25.5.1.4 warm_init . . . 57

25.6 fdt_match Struct Reference . . . 57
25.6.1 Field Documentation . . . 57
25.6.1.1 compatible . . . 57
25.6.1.2 data . . . 58
25.7 fdt_node_header Struct Reference . . . 58
25.7.1 Field Documentation . . . 58
25.7.1.1 name . . . 58
25.7.1.2 tag . . . 58
25.8 fdt_property Struct Reference . . . 58
25.8.1 Field Documentation . . . 58
25.8.1.1 data . . . 58
25.8.1.2 len . . . 58
25.8.1.3 nameoff . . . 59
25.8.1.4 tag . . . 59
25.9 fdt_reserve_entry Struct Reference . . . 59
25.9.1 Field Documentation . . . 59
25.9.1.1 address . . . 59
25.9.1.2 size . . . 59
25.10 fdt_reset Struct Reference . . . 59
25.10.1 Field Documentation . . . 59
25.10.1.1 init . . . 59
25.10.1.2 match_table . . . 60
25.10.1.3 system_reset . . . 60
25.11 fdt_serial Struct Reference . . . 60
25.11.1 Field Documentation . . . 60
25.11.1.1 getc . . . 60
25.11.1.2 init . . . 60
25.11.1.3 match_table . . . 60
25.11.1.4 putc . . . 60
25.12 fdt_timer Struct Reference . . . 60
25.12.1 Field Documentation . . . 61
25.12.1.1 cold_init . . . 61
25.12.1.2 event_start . . . 61
25.12.1.3 event_stop . . . 61
25.12.1.4 exit . . . 61
25.12.1.5 match_table . . . 61
25.12.1.6 value . . . 61
25.12.1.7 warm_init . . . 61
25.13 fw_dynamic_info Struct Reference . . . 61
25.13.1 Detailed Description . . . 62
25.13.2 Field Documentation . . . 62
25.13.2.1 boot_hart . . . 62

25.13.2.2 magic . . . 62
25.13.2.3 next_addr . . . 62
25.13.2.4 next_mode . . . 62
25.13.2.5 options . . . 62
25.13.2.6 version . . . 62
25.14 platform_uart_data Struct Reference . . . 62
25.14.1 Field Documentation . . . 63
25.14.1.1 addr . . . 63
25.14.1.2 baud . . . 63
25.14.1.3 freq . . . 63
25.14.1.4 reg_io_width . . . 63
25.14.1.5 reg_shift . . . 63
25.15 plic_data Struct Reference . . . 63
25.15.1 Field Documentation . . . 63
25.15.1.1 addr . . . 63
25.15.1.2 num_src . . . 64
25.16 sbi_dlist Struct Reference . . . 64
25.16.1 Field Documentation . . . 64
25.16.1.1 next . . . 64
25.16.1.2 prev . . . 64
25.17 sbi_ecall_extension Struct Reference . . . 64
25.17.1 Field Documentation . . . 64
25.17.1.1 extid_end . . . 64
25.17.1.2 extid_start . . . 64
25.17.1.3 handle . . . 65
25.17.1.4 head . . . 65
25.17.1.5 probe . . . 65
25.18 sbi_fifo Struct Reference . . . 65
25.18.1 Field Documentation . . . 65
25.18.1.1 avail . . . 65
25.18.1.2 entry_size . . . 65
25.18.1.3 num_entries . . . 65
25.18.1.4 qlock . . . 65
25.18.1.5 queue . . . 65
25.18.1.6 tail . . . 66
25.19 sbi_hartmask Struct Reference . . . 66
25.19.1 Detailed Description . . . 66
25.19.2 Member Function Documentation . . . 66
25.19.2.1 DECLARE_BITMAP() . . . 66
25.20 sbi_ipi_event_ops Struct Reference . . . 66
25.20.1 Detailed Description . . . 66
25.20.2 Field Documentation . . . 66

25.20.2.1 name . . . 66
25.20.2.2 process . . . 67
25.20.2.3 sync . . . 67
25.20.2.4 update . . . 67
25.21 sbi_platform Struct Reference . . . 67
25.21.1 Detailed Description . . . 67
25.21.2 Field Documentation . . . 67
25.21.2.1 features . . . 67
25.21.2.2 firmware_context . . . 67
25.21.2.3 hart_count . . . 68
25.21.2.4 hart_index2id . . . 68
25.21.2.5 hart_stack_size . . . 68
25.21.2.6 name . . . 68
25.21.2.7 opensbi_version . . . 68
25.21.2.8 platform_ops_addr . . . 68
25.21.2.9 platform_version . . . 68
25.22 sbi_platform_operations Struct Reference . . . 68
25.22.1 Detailed Description . . . 69
25.22.2 Field Documentation . . . 69
25.22.2.1 console_getc . . . 69
25.22.2.2 console_init . . . 69
25.22.2.3 console_putc . . . 69
25.22.2.4 early_exit . . . 69
25.22.2.5 early_init . . . 69
25.22.2.6 final_exit . . . 70
25.22.2.7 final_init . . . 70
25.22.2.8 get_tlbr_flush_limit . . . 70
25.22.2.9 hart_start . . . 70
25.22.2.10 hart_stop . . . 70
25.22.2.11 ipi_clear . . . 70
25.22.2.12 ipi_exit . . . 70
25.22.2.13 ipi_init . . . 70
25.22.2.14 ipi_send . . . 70
25.22.2.15 irqchip_exit . . . 70
25.22.2.16 irqchip_init . . . 70
25.22.2.17 misa_check_extension . . . 70
25.22.2.18 misa_get_xlen . . . 71
25.22.2.19 pmp_region_count . . . 71
25.22.2.20 pmp_region_info . . . 71
25.22.2.21 system_reset . . . 71
25.22.2.22 timer_event_start . . . 71
25.22.2.23 timer_event_stop . . . 71

25.22.2.24 timer_exit . . . 71
25.22.2.25 timer_init . . . 71
25.22.2.26 timer_value . . . 71
25.22.2.27 vendor_ext_check . . . 71
25.22.2.28 vendor_ext_provider . . . 71
25.23 sbi_scratch Struct Reference . . . 72
25.23.1 Detailed Description . . . 72
25.23.2 Field Documentation . . . 72
25.23.2.1 fw_size . . . 72
25.23.2.2 fw_start . . . 72
25.23.2.3 hartid_to_scratch . . . 72
25.23.2.4 next_addr . . . 72
25.23.2.5 next_arg1 . . . 72
25.23.2.6 next_mode . . . 72
25.23.2.7 options . . . 72
25.23.2.8 platform_addr . . . 73
25.23.2.9 tmp0 . . . 73
25.23.2.10 warmboot_addr . . . 73
25.24 sbi_tlb_info Struct Reference . . . 73
25.24.1 Field Documentation . . . 73
25.24.1.1 asid . . . 73
25.24.1.2 size . . . 73
25.24.1.3 smask . . . 73
25.24.1.4 start . . . 73
25.24.1.5 type . . . 73
25.24.1.6 vmid . . . 74
25.25 sbi_trap_info Struct Reference . . . 74
25.25.1 Detailed Description . . . 74
25.25.2 Field Documentation . . . 74
25.25.2.1 cause . . . 74
25.25.2.2 epc . . . 74
25.25.2.3 tinst . . . 74
25.25.2.4 tval . . . 74
25.25.2.5 tval2 . . . 74
25.26 sbi_trap_regs Struct Reference . . . 74
25.26.1 Detailed Description . . . 75
25.26.2 Field Documentation . . . 75
25.26.2.1 a0 . . . 75
25.26.2.2 a1 . . . 75
25.26.2.3 a2 . . . 76
25.26.2.4 a3 . . . 76
25.26.2.5 a4 . . . 76

25.26.2.6 a5 . . . 76
25.26.2.7 a6 . . . 76
25.26.2.8 a7 . . . 76
25.26.2.9 gp . . . 76
25.26.2.10 mepc . . . 76
25.26.2.11 mstatus . . . 76
25.26.2.12 mstatusH . . . 76
25.26.2.13 ra . . . 76
25.26.2.14 s0 . . . 76
25.26.2.15 s1 . . . 77
25.26.2.16 s10 . . . 77
25.26.2.17 s11 . . . 77
25.26.2.18 s2 . . . 77
25.26.2.19 s3 . . . 77
25.26.2.20 s4 . . . 77
25.26.2.21 s5 . . . 77
25.26.2.22 s6 . . . 77
25.26.2.23 s7 . . . 77
25.26.2.24 s8 . . . 77
25.26.2.25 s9 . . . 77
25.26.2.26 sp . . . 77
25.26.2.27 t0 . . . 78
25.26.2.28 t1 . . . 78
25.26.2.29 t2 . . . 78
25.26.2.30 t3 . . . 78
25.26.2.31 t4 . . . 78
25.26.2.32 t5 . . . 78
25.26.2.33 t6 . . . 78
25.26.2.34 tp . . . 78
25.26.2.35 zero . . . 78
25.27 spinlock_t Struct Reference . . . 78
25.27.1 Field Documentation . . . 78
25.27.1.1 lock . . . 79

26 File Documentation **81**
26.1 docs/contributing.md File Reference . . . 82
26.2 docs/firmware/fw.md File Reference . . . 82
26.3 docs/firmware/fw_dynamic.md File Reference . . . 82
26.4 docs/firmware/fw_jump.md File Reference . . . 82
26.5 docs/firmware/fw_payload.md File Reference . . . 82
26.6 docs/firmware/payload_linux.md File Reference . . . 82
26.7 docs/firmware/payload_uboot.md File Reference . . . 82

26.8 docs/library_usage.md File Reference . . . 82
26.9 docs/platform/andes-ae350.md File Reference . . . 82
26.10 docs/platform/fpga-ariane.md File Reference . . . 82
26.11 docs/platform/fpga-openpiton.md File Reference . . . 82
26.12 docs/platform/generic.md File Reference . . . 82
26.13 docs/platform/nuclei_ux600.md File Reference . . . 82
26.14 docs/platform/platform.md File Reference . . . 82
26.15 docs/platform/qemu_virt.md File Reference . . . 82
26.16 docs/platform/shakti_cclass.md File Reference . . . 82
26.17 docs/platform/sifive_fu540.md File Reference . . . 82
26.18 docs/platform/spike.md File Reference . . . 82
26.19 docs/platform/thead-c910.md File Reference . . . 82
26.20 docs/platform_guide.md File Reference . . . 82
26.21 docs/platform_requirements.md File Reference . . . 82
26.22 include/sbi/fw_dynamic.h File Reference . . . 82
26.22.1 Macro Definition Documentation . . . 83
26.22.1.1 FW_DYNAMIC_INFO_BOOT_HART_OFFSET . . . 83
26.22.1.2 FW_DYNAMIC_INFO_MAGIC_OFFSET . . . 84
26.22.1.3 FW_DYNAMIC_INFO_MAGIC_VALUE . . . 84
26.22.1.4 FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET . . . 84
26.22.1.5 FW_DYNAMIC_INFO_NEXT_MODE_M . . . 84
26.22.1.6 FW_DYNAMIC_INFO_NEXT_MODE_OFFSET . . . 84
26.22.1.7 FW_DYNAMIC_INFO_NEXT_MODE_S . . . 84
26.22.1.8 FW_DYNAMIC_INFO_NEXT_MODE_U . . . 84
26.22.1.9 FW_DYNAMIC_INFO_OPTIONS_OFFSET . . . 84
26.22.1.10 FW_DYNAMIC_INFO_VERSION_MAX . . . 84
26.22.1.11 FW_DYNAMIC_INFO_VERSION_OFFSET . . . 84
26.22.2 Variable Documentation . . . 84
26.22.2.1 __packed . . . 84
26.23 include/sbi/riscv_asm.h File Reference . . . 85
26.23.1 Macro Definition Documentation . . . 86
26.23.1.1 __ASM_STR . . . 86
26.23.1.2 csr_clear . . . 86
26.23.1.3 csr_read . . . 86
26.23.1.4 csr_read_clear . . . 86
26.23.1.5 csr_read_set . . . 86
26.23.1.6 csr_set . . . 87
26.23.1.7 csr_swap . . . 87
26.23.1.8 csr_write . . . 87
26.23.1.9 current_hartid . . . 87
26.23.1.10 LGREG . . . 87
26.23.1.11 misa_extension . . . 88

26.23.1.12 PAGE_MASK . . . 88
26.23.1.13 PAGE_SHIFT . . . 88
26.23.1.14 PAGE_SIZE . . . 88
26.23.1.15 REG_L . . . 88
26.23.1.16 REG_S . . . 88
26.23.1.17 SZREG . . . 88
26.23.1.18 wfi . . . 88
26.23.2 Function Documentation . . . 88
26.23.2.1 csr_read_num() . . . 88
26.23.2.2 csr_write_num() . . . 89
26.23.2.3 misa_extension_imp() . . . 89
26.23.2.4 misa_string() . . . 90
26.23.2.5 misa_xlen() . . . 90
26.23.2.6 pmp_get() . . . 90
26.23.2.7 pmp_set() . . . 91
26.24 include/sbi/riscv_atomic.h File Reference . . . 92
26.24.1 Macro Definition Documentation . . . 92
26.24.1.1 ATOMIC_INIT . . . 92
26.24.1.2 ATOMIC_INITIALIZER . . . 93
26.24.2 Function Documentation . . . 93
26.24.2.1 atomic_add_return() . . . 93
26.24.2.2 atomic_clear_bit() . . . 93
26.24.2.3 atomic_cmpxchg() . . . 93
26.24.2.4 atomic_raw_clear_bit() . . . 94
26.24.2.5 atomic_raw_set_bit() . . . 94
26.24.2.6 atomic_raw_xchg_uint() . . . 94
26.24.2.7 atomic_raw_xchg_ulong() . . . 94
26.24.2.8 atomic_read() . . . 95
26.24.2.9 atomic_set_bit() . . . 95
26.24.2.10 atomic_sub_return() . . . 95
26.24.2.11 atomic_write() . . . 96
26.24.2.12 atomic_xchg() . . . 96
26.25 include/sbi/riscv_barrier.h File Reference . . . 96
26.25.1 Macro Definition Documentation . . . 96
26.25.1.1 __smp_load_acquire . . . 97
26.25.1.2 __smp_store_release . . . 97
26.25.1.3 cpu_relax . . . 97
26.25.1.4 mb . . . 97
26.25.1.5 RISCV_ACQUIRE_BARRIER . . . 97
26.25.1.6 RISCV_FENCE . . . 97
26.25.1.7 RISCV_RELEASE_BARRIER . . . 97
26.25.1.8 rmb . . . 97

26.25.1.9 smp_mb . . . 97
26.25.1.10 smp_rmb . . . 98
26.25.1.11 smp_wmb . . . 98
26.25.1.12 wmb . . . 98
26.26 include/sbi/riscv_encoding.h File Reference . . . 98
26.26.1 Macro Definition Documentation . . . 107
26.26.1.1 CAUSE_BREAKPOINT . . . 107
26.26.1.2 CAUSE_FETCH_ACCESS . . . 107
26.26.1.3 CAUSE_FETCH_GUEST_PAGE_FAULT . . . 107
26.26.1.4 CAUSE_FETCH_PAGE_FAULT . . . 107
26.26.1.5 CAUSE_HYPERVISOR_ECALL . . . 107
26.26.1.6 CAUSE_ILLEGAL_INSTRUCTION . . . 108
26.26.1.7 CAUSE_LOAD_ACCESS . . . 108
26.26.1.8 CAUSE_LOAD_GUEST_PAGE_FAULT . . . 108
26.26.1.9 CAUSE_LOAD_PAGE_FAULT . . . 108
26.26.1.10 CAUSE_MACHINE_ECALL . . . 108
26.26.1.11 CAUSE_MISALIGNED_FETCH . . . 108
26.26.1.12 CAUSE_MISALIGNED_LOAD . . . 108
26.26.1.13 CAUSE_MISALIGNED_STORE . . . 108
26.26.1.14 CAUSE_STORE_ACCESS . . . 108
26.26.1.15 CAUSE_STORE_GUEST_PAGE_FAULT . . . 108
26.26.1.16 CAUSE_STORE_PAGE_FAULT . . . 108
26.26.1.17 CAUSE_SUPERVISOR_ECALL . . . 108
26.26.1.18 CAUSE_USER_ECALL . . . 109
26.26.1.19 CAUSE_VIRTUAL_INST_FAULT . . . 109
26.26.1.20 CSR_CYCLE . . . 109
26.26.1.21 CSR_CYCLEH . . . 109
26.26.1.22 CSR_DCSR . . . 109
26.26.1.23 CSR_DPC . . . 109
26.26.1.24 CSR_DSCRATCH . . . 109
26.26.1.25 CSR_FCSR . . . 109
26.26.1.26 CSR_FFLAGS . . . 109
26.26.1.27 CSR_FRM . . . 109
26.26.1.28 CSR_HCOUNTERNEN . . . 109
26.26.1.29 CSR_HEDELEG . . . 109
26.26.1.30 CSR_HGATP . . . 110
26.26.1.31 CSR_HGEIE . . . 110
26.26.1.32 CSR_HGEIP . . . 110
26.26.1.33 CSR_HIDELEG . . . 110
26.26.1.34 CSR_HIE . . . 110
26.26.1.35 CSR_HIP . . . 110
26.26.1.36 CSR_HPMCOUNTER10 . . . 110

26.26.1.37 CSR_HPMCOUNTER10H 110
26.26.1.38 CSR_HPMCOUNTER11 110
26.26.1.39 CSR_HPMCOUNTER11H 110
26.26.1.40 CSR_HPMCOUNTER12 110
26.26.1.41 CSR_HPMCOUNTER12H 110
26.26.1.42 CSR_HPMCOUNTER13 111
26.26.1.43 CSR_HPMCOUNTER13H 111
26.26.1.44 CSR_HPMCOUNTER14 111
26.26.1.45 CSR_HPMCOUNTER14H 111
26.26.1.46 CSR_HPMCOUNTER15 111
26.26.1.47 CSR_HPMCOUNTER15H 111
26.26.1.48 CSR_HPMCOUNTER16 111
26.26.1.49 CSR_HPMCOUNTER16H 111
26.26.1.50 CSR_HPMCOUNTER17 111
26.26.1.51 CSR_HPMCOUNTER17H 111
26.26.1.52 CSR_HPMCOUNTER18 111
26.26.1.53 CSR_HPMCOUNTER18H 111
26.26.1.54 CSR_HPMCOUNTER19 112
26.26.1.55 CSR_HPMCOUNTER19H 112
26.26.1.56 CSR_HPMCOUNTER20 112
26.26.1.57 CSR_HPMCOUNTER20H 112
26.26.1.58 CSR_HPMCOUNTER21 112
26.26.1.59 CSR_HPMCOUNTER21H 112
26.26.1.60 CSR_HPMCOUNTER22 112
26.26.1.61 CSR_HPMCOUNTER22H 112
26.26.1.62 CSR_HPMCOUNTER23 112
26.26.1.63 CSR_HPMCOUNTER23H 112
26.26.1.64 CSR_HPMCOUNTER24 112
26.26.1.65 CSR_HPMCOUNTER24H 112
26.26.1.66 CSR_HPMCOUNTER25 113
26.26.1.67 CSR_HPMCOUNTER25H 113
26.26.1.68 CSR_HPMCOUNTER26 113
26.26.1.69 CSR_HPMCOUNTER26H 113
26.26.1.70 CSR_HPMCOUNTER27 113
26.26.1.71 CSR_HPMCOUNTER27H 113
26.26.1.72 CSR_HPMCOUNTER28 113
26.26.1.73 CSR_HPMCOUNTER28H 113
26.26.1.74 CSR_HPMCOUNTER29 113
26.26.1.75 CSR_HPMCOUNTER29H 113
26.26.1.76 CSR_HPMCOUNTER3 113
26.26.1.77 CSR_HPMCOUNTER30 113
26.26.1.78 CSR_HPMCOUNTER30H 114

26.26.1.79 CSR_HPMCOUNTER31 114
26.26.1.80 CSR_HPMCOUNTER31H 114
26.26.1.81 CSR_HPMCOUNTER3H 114
26.26.1.82 CSR_HPMCOUNTER4 114
26.26.1.83 CSR_HPMCOUNTER4H 114
26.26.1.84 CSR_HPMCOUNTER5 114
26.26.1.85 CSR_HPMCOUNTER5H 114
26.26.1.86 CSR_HPMCOUNTER6 114
26.26.1.87 CSR_HPMCOUNTER6H 114
26.26.1.88 CSR_HPMCOUNTER7 114
26.26.1.89 CSR_HPMCOUNTER7H 114
26.26.1.90 CSR_HPMCOUNTER8 115
26.26.1.91 CSR_HPMCOUNTER8H 115
26.26.1.92 CSR_HPMCOUNTER9 115
26.26.1.93 CSR_HPMCOUNTER9H 115
26.26.1.94 CSR_HSTATUS 115
26.26.1.95 CSR_HTIMEDELTA 115
26.26.1.96 CSR_HTIMEDELTAH 115
26.26.1.97 CSR_HTINST 115
26.26.1.98 CSR_HTVAL 115
26.26.1.99 CSR_INSTRET 115
26.26.1.100 CSR_INSTRETH 115
26.26.1.101 CSR_MARCHID 115
26.26.1.102 CSR_MCAUSE 116
26.26.1.103 CSR_MCOUNTEREN 116
26.26.1.104 CSR_MCYCLE 116
26.26.1.105 CSR_MCYCLEH 116
26.26.1.106 CSR_MEDELEG 116
26.26.1.107 CSR_MEPC 116
26.26.1.108 CSR_MHARTID 116
26.26.1.109 CSR_MHPMCOUNTER10 116
26.26.1.110 CSR_MHPMCOUNTER10H 116
26.26.1.111 CSR_MHPMCOUNTER11 116
26.26.1.112 CSR_MHPMCOUNTER11H 116
26.26.1.113 CSR_MHPMCOUNTER12 116
26.26.1.114 CSR_MHPMCOUNTER12H 117
26.26.1.115 CSR_MHPMCOUNTER13 117
26.26.1.116 CSR_MHPMCOUNTER13H 117
26.26.1.117 CSR_MHPMCOUNTER14 117
26.26.1.118 CSR_MHPMCOUNTER14H 117
26.26.1.119 CSR_MHPMCOUNTER15 117
26.26.1.120 CSR_MHPMCOUNTER15H 117

26.26.1.121 CSR_MHPMCOUNTER16 117
26.26.1.122 CSR_MHPMCOUNTER16H 117
26.26.1.123 CSR_MHPMCOUNTER17 117
26.26.1.124 CSR_MHPMCOUNTER17H 117
26.26.1.125 CSR_MHPMCOUNTER18 117
26.26.1.126 CSR_MHPMCOUNTER18H 118
26.26.1.127 CSR_MHPMCOUNTER19 118
26.26.1.128 CSR_MHPMCOUNTER19H 118
26.26.1.129 CSR_MHPMCOUNTER20 118
26.26.1.130 CSR_MHPMCOUNTER20H 118
26.26.1.131 CSR_MHPMCOUNTER21 118
26.26.1.132 CSR_MHPMCOUNTER21H 118
26.26.1.133 CSR_MHPMCOUNTER22 118
26.26.1.134 CSR_MHPMCOUNTER22H 118
26.26.1.135 CSR_MHPMCOUNTER23 118
26.26.1.136 CSR_MHPMCOUNTER23H 118
26.26.1.137 CSR_MHPMCOUNTER24 118
26.26.1.138 CSR_MHPMCOUNTER24H 119
26.26.1.139 CSR_MHPMCOUNTER25 119
26.26.1.140 CSR_MHPMCOUNTER25H 119
26.26.1.141 CSR_MHPMCOUNTER26 119
26.26.1.142 CSR_MHPMCOUNTER26H 119
26.26.1.143 CSR_MHPMCOUNTER27 119
26.26.1.144 CSR_MHPMCOUNTER27H 119
26.26.1.145 CSR_MHPMCOUNTER28 119
26.26.1.146 CSR_MHPMCOUNTER28H 119
26.26.1.147 CSR_MHPMCOUNTER29 119
26.26.1.148 CSR_MHPMCOUNTER29H 119
26.26.1.149 CSR_MHPMCOUNTER3 119
26.26.1.150 CSR_MHPMCOUNTER30 120
26.26.1.151 CSR_MHPMCOUNTER30H 120
26.26.1.152 CSR_MHPMCOUNTER31 120
26.26.1.153 CSR_MHPMCOUNTER31H 120
26.26.1.154 CSR_MHPMCOUNTER3H 120
26.26.1.155 CSR_MHPMCOUNTER4 120
26.26.1.156 CSR_MHPMCOUNTER4H 120
26.26.1.157 CSR_MHPMCOUNTER5 120
26.26.1.158 CSR_MHPMCOUNTER5H 120
26.26.1.159 CSR_MHPMCOUNTER6 120
26.26.1.160 CSR_MHPMCOUNTER6H 120
26.26.1.161 CSR_MHPMCOUNTER7 120
26.26.1.162 CSR_MHPMCOUNTER7H 121

26.26.1.163 CSR_MHPMCOUNTER8 121
26.26.1.164 CSR_MHPMCOUNTER8H 121
26.26.1.165 CSR_MHPMCOUNTER9 121
26.26.1.166 CSR_MHPMCOUNTER9H 121
26.26.1.167 CSR_MHPMEVENT10 121
26.26.1.168 CSR_MHPMEVENT11 121
26.26.1.169 CSR_MHPMEVENT12 121
26.26.1.170 CSR_MHPMEVENT13 121
26.26.1.171 CSR_MHPMEVENT14 121
26.26.1.172 CSR_MHPMEVENT15 121
26.26.1.173 CSR_MHPMEVENT16 121
26.26.1.174 CSR_MHPMEVENT17 122
26.26.1.175 CSR_MHPMEVENT18 122
26.26.1.176 CSR_MHPMEVENT19 122
26.26.1.177 CSR_MHPMEVENT20 122
26.26.1.178 CSR_MHPMEVENT21 122
26.26.1.179 CSR_MHPMEVENT22 122
26.26.1.180 CSR_MHPMEVENT23 122
26.26.1.181 CSR_MHPMEVENT24 122
26.26.1.182 CSR_MHPMEVENT25 122
26.26.1.183 CSR_MHPMEVENT26 122
26.26.1.184 CSR_MHPMEVENT27 122
26.26.1.185 CSR_MHPMEVENT28 122
26.26.1.186 CSR_MHPMEVENT29 123
26.26.1.187 CSR_MHPMEVENT3 123
26.26.1.188 CSR_MHPMEVENT30 123
26.26.1.189 CSR_MHPMEVENT31 123
26.26.1.190 CSR_MHPMEVENT4 123
26.26.1.191 CSR_MHPMEVENT5 123
26.26.1.192 CSR_MHPMEVENT6 123
26.26.1.193 CSR_MHPMEVENT7 123
26.26.1.194 CSR_MHPMEVENT8 123
26.26.1.195 CSR_MHPMEVENT9 123
26.26.1.196 CSR_MIDELEG 123
26.26.1.197 CSR_MIE 123
26.26.1.198 CSR_MIMPID 124
26.26.1.199 CSR_MINSTRET 124
26.26.1.200 CSR_MINSTRETH 124
26.26.1.201 CSR_MIP 124
26.26.1.202 CSR_MISA 124
26.26.1.203 CSR_MSCRATCH 124
26.26.1.204 CSR_MSTATUS 124

26.26.1.205 CSR_MSTATUSH . . . 124
26.26.1.206 CSR_MTINST . . . 124
26.26.1.207 CSR_MTVAL . . . 124
26.26.1.208 CSR_MTVAL2 . . . 124
26.26.1.209 CSR_MTVEC . . . 124
26.26.1.210 CSR_MVENDORID . . . 125
26.26.1.211 CSR_PMPADDR0 . . . 125
26.26.1.212 CSR_PMPADDR1 . . . 125
26.26.1.213 CSR_PMPADDR10 . . . 125
26.26.1.214 CSR_PMPADDR11 . . . 125
26.26.1.215 CSR_PMPADDR12 . . . 125
26.26.1.216 CSR_PMPADDR13 . . . 125
26.26.1.217 CSR_PMPADDR14 . . . 125
26.26.1.218 CSR_PMPADDR15 . . . 125
26.26.1.219 CSR_PMPADDR2 . . . 125
26.26.1.220 CSR_PMPADDR3 . . . 125
26.26.1.221 CSR_PMPADDR4 . . . 125
26.26.1.222 CSR_PMPADDR5 . . . 126
26.26.1.223 CSR_PMPADDR6 . . . 126
26.26.1.224 CSR_PMPADDR7 . . . 126
26.26.1.225 CSR_PMPADDR8 . . . 126
26.26.1.226 CSR_PMPADDR9 . . . 126
26.26.1.227 CSR_PMPCFG0 . . . 126
26.26.1.228 CSR_PMPCFG1 . . . 126
26.26.1.229 CSR_PMPCFG2 . . . 126
26.26.1.230 CSR_PMPCFG3 . . . 126
26.26.1.231 CSR_SATP . . . 126
26.26.1.232 CSR_SCAUSE . . . 126
26.26.1.233 CSR_SCOUNTEREN . . . 126
26.26.1.234 CSR_SEPC . . . 127
26.26.1.235 CSR_SIE . . . 127
26.26.1.236 CSR_SIP . . . 127
26.26.1.237 CSR_SSCRATCH . . . 127
26.26.1.238 CSR_SSTATUS . . . 127
26.26.1.239 CSR_STVAL . . . 127
26.26.1.240 CSR_STVEC . . . 127
26.26.1.241 CSR_TDATA1 . . . 127
26.26.1.242 CSR_TDATA2 . . . 127
26.26.1.243 CSR_TDATA3 . . . 127
26.26.1.244 CSR_TIME . . . 127
26.26.1.245 CSR_TIMEH . . . 127
26.26.1.246 CSR_TSELECT . . . 128

26.26.1.247 CSR_UCAUSE . . . 128
26.26.1.248 CSR_UEPC . . . 128
26.26.1.249 CSR_UIE . . . 128
26.26.1.250 CSR_UIP . . . 128
26.26.1.251 CSR_USCRATCH . . . 128
26.26.1.252 CSR_USTATUS . . . 128
26.26.1.253 CSR_UTVAL . . . 128
26.26.1.254 CSR_UTVEC . . . 128
26.26.1.255 CSR_VSATP . . . 128
26.26.1.256 CSR_VSCAUSE . . . 128
26.26.1.257 CSR_VSEPC . . . 128
26.26.1.258 CSR_VSIE . . . 129
26.26.1.259 CSR_VSIP . . . 129
26.26.1.260 CSR_VSSCRATCH . . . 129
26.26.1.261 CSR_VSSTATUS . . . 129
26.26.1.262 CSR_VSTVAL . . . 129
26.26.1.263 CSR_VSTVEC . . . 129
26.26.1.264 GET_RM . . . 129
26.26.1.265 GET_RS1 . . . 129
26.26.1.266 GET_RS1S . . . 129
26.26.1.267 GET_RS2 . . . 129
26.26.1.268 GET_RS2C . . . 130
26.26.1.269 GET_RS2S . . . 130
26.26.1.270 GET_SP . . . 130
26.26.1.271 HGATP32_MODE_SHIFT . . . 130
26.26.1.272 HGATP32_PPN . . . 130
26.26.1.273 HGATP32_VMID_MASK . . . 130
26.26.1.274 HGATP32_VMID_SHIFT . . . 130
26.26.1.275 HGATP64_MODE_SHIFT . . . 130
26.26.1.276 HGATP64_PPN . . . 130
26.26.1.277 HGATP64_VMID_MASK . . . 130
26.26.1.278 HGATP64_VMID_SHIFT . . . 130
26.26.1.279 HGATP_MODE_OFF . . . 131
26.26.1.280 HGATP_MODE_SHIFT . . . 131
26.26.1.281 HGATP_MODE_SV32X4 . . . 131
26.26.1.282 HGATP_MODE_SV39X4 . . . 131
26.26.1.283 HGATP_MODE_SV48X4 . . . 131
26.26.1.284 HGATP_PPN . . . 131
26.26.1.285 HGATP_VMID_MASK . . . 131
26.26.1.286 HGATP_VMID_SHIFT . . . 131
26.26.1.287 HSTATUS_GVA . . . 131
26.26.1.288 HSTATUS_HU . . . 131

26.26.1.289 HSTATUS_SPV . . . 131
26.26.1.290 HSTATUS_SPVP . . . 131
26.26.1.291 HSTATUS_VGEIN . . . 132
26.26.1.292 HSTATUS_VGEIN_SHIFT . . . 132
26.26.1.293 HSTATUS_VSBE . . . 132
26.26.1.294 HSTATUS_VTSR . . . 132
26.26.1.295 HSTATUS_VTVM . . . 132
26.26.1.296 HSTATUS_VTW . . . 132
26.26.1.297 IMM_I . . . 132
26.26.1.298 IMM_S . . . 132
26.26.1.299 INSN_16BIT_MASK . . . 132
26.26.1.300 INSN_32BIT_MASK . . . 132
26.26.1.301 INSN_IS_16BIT . . . 132
26.26.1.302 INSN_IS_32BIT . . . 133
26.26.1.303 INSN_LEN . . . 133
26.26.1.304 INSN_MASK_C_FLD . . . 133
26.26.1.305 INSN_MASK_C_FLDSP . . . 133
26.26.1.306 INSN_MASK_C_FLW . . . 133
26.26.1.307 INSN_MASK_C_FLWSP . . . 133
26.26.1.308 INSN_MASK_C_FSD . . . 133
26.26.1.309 INSN_MASK_C_FSDSP . . . 133
26.26.1.310 INSN_MASK_C_FSW . . . 133
26.26.1.311 INSN_MASK_C_FSWSP . . . 133
26.26.1.312 INSN_MASK_C_LD . . . 133
26.26.1.313 INSN_MASK_C_LDSP . . . 134
26.26.1.314 INSN_MASK_C_LW . . . 134
26.26.1.315 INSN_MASK_C_LWSP . . . 134
26.26.1.316 INSN_MASK_C_SD . . . 134
26.26.1.317 INSN_MASK_C_SDSP . . . 134
26.26.1.318 INSN_MASK_C_SW . . . 134
26.26.1.319 INSN_MASK_C_SWSP . . . 134
26.26.1.320 INSN_MASK_FLD . . . 134
26.26.1.321 INSN_MASK_FLQ . . . 134
26.26.1.322 INSN_MASK_FLW . . . 134
26.26.1.323 INSN_MASK_FSD . . . 134
26.26.1.324 INSN_MASK_FSQ . . . 134
26.26.1.325 INSN_MASK_FSW . . . 135
26.26.1.326 INSN_MASK_LB . . . 135
26.26.1.327 INSN_MASK_LBU . . . 135
26.26.1.328 INSN_MASK_LD . . . 135
26.26.1.329 INSN_MASK_LH . . . 135
26.26.1.330 INSN_MASK_LHU . . . 135

26.26.1.331 INSN_MASK_LW . . . 135
26.26.1.332 INSN_MASK_LWU . . . 135
26.26.1.333 INSN_MASK_SB . . . 135
26.26.1.334 INSN_MASK_SD . . . 135
26.26.1.335 INSN_MASK_SH . . . 135
26.26.1.336 INSN_MASK_SW . . . 135
26.26.1.337 INSN_MASK_WFI . . . 136
26.26.1.338 INSN_MATCH_C_FLD . . . 136
26.26.1.339 INSN_MATCH_C_FLDSP . . . 136
26.26.1.340 INSN_MATCH_C_FLW . . . 136
26.26.1.341 INSN_MATCH_C_FLWSP . . . 136
26.26.1.342 INSN_MATCH_C_FSD . . . 136
26.26.1.343 INSN_MATCH_C_FSDSP . . . 136
26.26.1.344 INSN_MATCH_C_FSW . . . 136
26.26.1.345 INSN_MATCH_C_FSWSP . . . 136
26.26.1.346 INSN_MATCH_C_LD . . . 136
26.26.1.347 INSN_MATCH_C_LDSP . . . 136
26.26.1.348 INSN_MATCH_C_LW . . . 136
26.26.1.349 INSN_MATCH_C_LWSP . . . 137
26.26.1.350 INSN_MATCH_C_SD . . . 137
26.26.1.351 INSN_MATCH_C_SDSP . . . 137
26.26.1.352 INSN_MATCH_C_SW . . . 137
26.26.1.353 INSN_MATCH_C_SWSP . . . 137
26.26.1.354 INSN_MATCH_FLD . . . 137
26.26.1.355 INSN_MATCH_FLQ . . . 137
26.26.1.356 INSN_MATCH_FLW . . . 137
26.26.1.357 INSN_MATCH_FSD . . . 137
26.26.1.358 INSN_MATCH_FSQ . . . 137
26.26.1.359 INSN_MATCH_FSW . . . 137
26.26.1.360 INSN_MATCH_LB . . . 137
26.26.1.361 INSN_MATCH_LBU . . . 138
26.26.1.362 INSN_MATCH_LD . . . 138
26.26.1.363 INSN_MATCH_LH . . . 138
26.26.1.364 INSN_MATCH_LHU . . . 138
26.26.1.365 INSN_MATCH_LW . . . 138
26.26.1.366 INSN_MATCH_LWU . . . 138
26.26.1.367 INSN_MATCH_SB . . . 138
26.26.1.368 INSN_MATCH_SD . . . 138
26.26.1.369 INSN_MATCH_SH . . . 138
26.26.1.370 INSN_MATCH_SW . . . 138
26.26.1.371 INSN_MATCH_WFI . . . 138
26.26.1.372 IRQ_M_EXT . . . 138

26.26.1.373 IRQ_M_SOFT . . . 139
26.26.1.374 IRQ_M_TIMER . . . 139
26.26.1.375 IRQ_S_EXT . . . 139
26.26.1.376 IRQ_S_GEXT . . . 139
26.26.1.377 IRQ_S_SOFT . . . 139
26.26.1.378 IRQ_S_TIMER . . . 139
26.26.1.379 IRQ_VS_EXT . . . 139
26.26.1.380 IRQ_VS_SOFT . . . 139
26.26.1.381 IRQ_VS_TIMER . . . 139
26.26.1.382 LOG_REGBYTES . . . 139
26.26.1.383 MASK_FUNCT3 . . . 139
26.26.1.384 MIP_MEIP . . . 139
26.26.1.385 MIP_MSIP . . . 140
26.26.1.386 MIP_MTIP . . . 140
26.26.1.387 MIP_SEIP . . . 140
26.26.1.388 MIP_SGEIP . . . 140
26.26.1.389 MIP_SSIP . . . 140
26.26.1.390 MIP_STIP . . . 140
26.26.1.391 MIP_VSEIP . . . 140
26.26.1.392 MIP_VSSIP . . . 140
26.26.1.393 MIP_VSTIP . . . 140
26.26.1.394 MSTATUS32_SD [1/2] . . . 140
26.26.1.395 MSTATUS32_SD [2/2] . . . 140
26.26.1.396 MSTATUS64_SD . . . 140
26.26.1.397 MSTATUS_FS . . . 141
26.26.1.398 MSTATUS_MIE . . . 141
26.26.1.399 MSTATUS_MPIE . . . 141
26.26.1.400 MSTATUS_MPP . . . 141
26.26.1.401 MSTATUS_MPP_SHIFT . . . 141
26.26.1.402 MSTATUS_MPRV . . . 141
26.26.1.403 MSTATUS_MXR . . . 141
26.26.1.404 MSTATUS_SD . . . 141
26.26.1.405 MSTATUS_SIE . . . 141
26.26.1.406 MSTATUS_SPIE . . . 141
26.26.1.407 MSTATUS_SPIE_SHIFT . . . 141
26.26.1.408 MSTATUS_SPP . . . 141
26.26.1.409 MSTATUS_SPP_SHIFT . . . 142
26.26.1.410 MSTATUS_SUM . . . 142
26.26.1.411 MSTATUS_TSR . . . 142
26.26.1.412 MSTATUS_TVM . . . 142
26.26.1.413 MSTATUS_TW . . . 142
26.26.1.414 MSTATUS_UBE . . . 142

26.26.1.415 MSTATUS_VS . . . 142
26.26.1.416 MSTATUS_XS . . . 142
26.26.1.417 MSTATUSH_MBE . . . 142
26.26.1.418 MSTATUSH_MPV . . . 142
26.26.1.419 MSTATUSH_SBE . . . 142
26.26.1.420 PMP_A . . . 142
26.26.1.421 PMP_A_NA4 . . . 143
26.26.1.422 PMP_A_NAPOT . . . 143
26.26.1.423 PMP_A_TOR . . . 143
26.26.1.424 PMP_COUNT . . . 143
26.26.1.425 PMP_L . . . 143
26.26.1.426 PMP_R . . . 143
26.26.1.427 PMP_SHIFT . . . 143
26.26.1.428 PMP_W . . . 143
26.26.1.429 PMP_X . . . 143
26.26.1.430 PRV_M . . . 143
26.26.1.431 PRV_S . . . 143
26.26.1.432 PRV_U . . . 143
26.26.1.433 REG_MASK . . . 144
26.26.1.434 REG_OFFSET . . . 144
26.26.1.435 REG_PTR . . . 144
26.26.1.436 REGBYTES . . . 144
26.26.1.437 RV_X . . . 144
26.26.1.438 RVC_LD_IMM . . . 144
26.26.1.439 RVC_LDSP_IMM . . . 144
26.26.1.440 RVC_LW_IMM . . . 144
26.26.1.441 RVC_LWSP_IMM . . . 145
26.26.1.442 RVC_RS1S . . . 145
26.26.1.443 RVC_RS2 . . . 145
26.26.1.444 RVC_RS2S . . . 145
26.26.1.445 RVC_SDSP_IMM . . . 145
26.26.1.446 RVC_SWSP_IMM . . . 145
26.26.1.447 SATP32_ASID . . . 145
26.26.1.448 SATP32_MODE . . . 145
26.26.1.449 SATP32_PPN . . . 145
26.26.1.450 SATP64_ASID . . . 146
26.26.1.451 SATP64_MODE . . . 146
26.26.1.452 SATP64_PPN . . . 146
26.26.1.453 SATP_MODE . . . 146
26.26.1.454 SATP_MODE_OFF . . . 146
26.26.1.455 SATP_MODE_SV32 . . . 146
26.26.1.456 SATP_MODE_SV39 . . . 146

26.26.1.457 SATP_MODE_SV48 . . . 146
26.26.1.458 SATP_MODE_SV57 . . . 146
26.26.1.459 SATP_MODE_SV64 . . . 146
26.26.1.460 SET_RD . . . 146
26.26.1.461 SH_RD . . . 147
26.26.1.462 SH_RS1 . . . 147
26.26.1.463 SH_RS2 . . . 147
26.26.1.464 SH_RS2C . . . 147
26.26.1.465 SHIFT_RIGHT . . . 147
26.26.1.466 SIP_SSIP . . . 147
26.26.1.467 SIP_STIP . . . 147
26.26.1.468 SSTATUS32_SD . . . 147
26.26.1.469 SSTATUS64_SD . . . 147
26.26.1.470 SSTATUS64_UXL . . . 147
26.26.1.471 SSTATUS_FS . . . 147
26.26.1.472 SSTATUS_MXR . . . 147
26.26.1.473 SSTATUS_SD . . . 148
26.26.1.474 SSTATUS_SIE . . . 148
26.26.1.475 SSTATUS_SPIE . . . 148
26.26.1.476 SSTATUS_SPIE_SHIFT . . . 148
26.26.1.477 SSTATUS_SPP . . . 148
26.26.1.478 SSTATUS_SPP_SHIFT . . . 148
26.26.1.479 SSTATUS_SUM . . . 148
26.26.1.480 SSTATUS_VS . . . 148
26.26.1.481 SSTATUS_XS . . . 148
26.27 include/sbi/riscv_fp.h File Reference . . . 148
26.27.1 Macro Definition Documentation . . . 149
26.27.1.1 GET_PRECISION . . . 149
26.27.1.2 GET_RM . . . 150
26.27.1.3 PRECISION_D . . . 150
26.27.1.4 PRECISION_S . . . 150
26.28 include/sbi/riscv_io.h File Reference . . . 150
26.28.1 Macro Definition Documentation . . . 151
26.28.1.1 __io_ar . . . 151
26.28.1.2 __io_aw . . . 151
26.28.1.3 __io_br . . . 151
26.28.1.4 __io_bw . . . 151
26.28.1.5 __io_rar . . . 151
26.28.1.6 __io_raw . . . 152
26.28.1.7 __io_rbr . . . 152
26.28.1.8 __io_rbw . . . 152
26.28.1.9 readb . . . 152

26.28.1.10 readb_relaxed . . . 152
26.28.1.11 readl . . . 152
26.28.1.12 readl_relaxed . . . 152
26.28.1.13 readq . . . 152
26.28.1.14 readq_relaxed . . . 152
26.28.1.15 readw . . . 152
26.28.1.16 readw_relaxed . . . 153
26.28.1.17 writeb . . . 153
26.28.1.18 writeb_relaxed . . . 153
26.28.1.19 writel . . . 153
26.28.1.20 writel_relaxed . . . 153
26.28.1.21 writeq . . . 153
26.28.1.22 writeq_relaxed . . . 153
26.28.1.23 writew . . . 153
26.28.1.24 writew_relaxed . . . 153
26.28.2 Function Documentation . . . 154
26.28.2.1 __raw_readb() . . . 154
26.28.2.2 __raw_readl() . . . 154
26.28.2.3 __raw_readq() . . . 154
26.28.2.4 __raw_readw() . . . 154
26.28.2.5 __raw_writeb() . . . 154
26.28.2.6 __raw_writel() . . . 154
26.28.2.7 __raw_writeq() . . . 154
26.28.2.8 __raw_writew() . . . 154
26.29 include/sbi/riscv_locks.h File Reference . . . 155
26.29.1 Macro Definition Documentation . . . 155
26.29.1.1 __RISCV_SPIN_UNLOCKED . . . 155
26.29.1.2 SPIN_LOCK_INIT . . . 155
26.29.1.3 SPIN_LOCK_INITIALIZER . . . 155
26.29.2 Function Documentation . . . 155
26.29.2.1 spin_lock() . . . 156
26.29.2.2 spin_lock_check() . . . 156
26.29.2.3 spin_trylock() . . . 157
26.29.2.4 spin_unlock() . . . 157
26.30 include/sbi/sbi_bitmap.h File Reference . . . 158
26.30.1 Macro Definition Documentation . . . 159
26.30.1.1 BITMAP_FIRST_WORD_MASK . . . 159
26.30.1.2 BITMAP_LAST_WORD_MASK . . . 159
26.30.1.3 DECLARE_BITMAP . . . 160
26.30.1.4 DEFINE_BITMAP . . . 160
26.30.1.5 small_const_nbits . . . 160
26.30.2 Function Documentation . . . 160

26.30.2.1 __bitmap_and() . . . 160
26.30.2.2 __bitmap_or() . . . 160
26.30.2.3 __bitmap_xor() . . . 161
26.30.2.4 bitmap_and() . . . 161
26.30.2.5 bitmap_clear() . . . 161
26.30.2.6 bitmap_copy() . . . 162
26.30.2.7 bitmap_estimate_size() . . . 162
26.30.2.8 bitmap_fill() . . . 162
26.30.2.9 bitmap_or() . . . 162
26.30.2.10 bitmap_set() . . . 163
26.30.2.11 bitmap_xor() . . . 163
26.30.2.12 bitmap_zero() . . . 163
26.30.2.13 bitmap_zero_except() . . . 164
26.31 include/sbi/sbi_bitops.h File Reference . . . 164
26.31.1 Macro Definition Documentation . . . 165
26.31.1.1 BIT . . . 165
26.31.1.2 BIT_MASK . . . 165
26.31.1.3 BIT_WORD . . . 166
26.31.1.4 BIT_WORD_OFFSET . . . 166
26.31.1.5 BITS_TO_LONGS . . . 166
26.31.1.6 EXTRACT_FIELD . . . 166
26.31.1.7 ffz . . . 166
26.31.1.8 for_each_clear_bit . . . 166
26.31.1.9 for_each_clear_bit_from . . . 166
26.31.1.10 for_each_set_bit . . . 166
26.31.1.11 for_each_set_bit_from . . . 167
26.31.1.12 GENMASK . . . 167
26.31.1.13 INSERT_FIELD . . . 167
26.31.2 Function Documentation . . . 167
26.31.2.1 __change_bit() . . . 167
26.31.2.2 __clear_bit() . . . 167
26.31.2.3 __ffs() . . . 168
26.31.2.4 __fls() . . . 168
26.31.2.5 __set_bit() . . . 168
26.31.2.6 __test_and_clear_bit() . . . 169
26.31.2.7 __test_and_set_bit() . . . 169
26.31.2.8 __test_bit() . . . 169
26.31.2.9 ffs() . . . 169
26.31.2.10 find_first_bit() . . . 169
26.31.2.11 find_first_zero_bit() . . . 170
26.31.2.12 find_last_bit() . . . 170
26.31.2.13 find_next_bit() . . . 170

26.31.2.14 find_next_zero_bit() . . . 170
26.31.2.15 fls() . . . 171
26.32 include/sbi/sbi_console.h File Reference . . . 171
26.32.1 Macro Definition Documentation . . . 172
26.32.1.1 __printf . . . 172
26.32.2 Function Documentation . . . 172
26.32.2.1 __printf() [1/3] . . . 172
26.32.2.2 __printf() [2/3] . . . 172
26.32.2.3 __printf() [3/3] . . . 172
26.32.2.4 sbi_console_init() . . . 172
26.32.2.5 sbi_getc() . . . 173
26.32.2.6 sbi_gets() . . . 173
26.32.2.7 sbi_isprintable() . . . 174
26.32.2.8 sbi_putc() . . . 174
26.32.2.9 sbi_puts() . . . 174
26.32.3 Variable Documentation . . . 175
26.32.3.1 format . . . 175
26.32.3.2 out_sz . . . 175
26.33 include/sbi/sbi_const.h File Reference . . . 175
26.33.1 Macro Definition Documentation . . . 175
26.33.1.1 __AC . . . 176
26.33.1.2 __STR . . . 176
26.33.1.3 _AC . . . 176
26.33.1.4 _AT . . . 176
26.33.1.5 _BITUL . . . 176
26.33.1.6 _BITULL . . . 176
26.33.1.7 _UL . . . 176
26.33.1.8 _ULL . . . 176
26.33.1.9 STRINGIFY . . . 176
26.33.1.10 UL . . . 177
26.33.1.11 ULL . . . 177
26.34 include/sbi/sbi_csr_detect.h File Reference . . . 177
26.34.1 Macro Definition Documentation . . . 178
26.34.1.1 csr_read_allowed . . . 178
26.34.1.2 csr_write_allowed . . . 178
26.35 include/sbi/sbi_ecall.h File Reference . . . 178
26.35.1 Macro Definition Documentation . . . 180
26.35.1.1 SBI_ECALL_VERSION_MAJOR . . . 180
26.35.1.2 SBI_ECALL_VERSION_MINOR . . . 180
26.35.1.3 SBI_OPENSBI_IMPID . . . 180
26.35.2 Function Documentation . . . 180
26.35.2.1 sbi_ecall_find_extension() . . . 180

26.35.2.2 sbi_ecall_get_impid() . . . 180
26.35.2.3 sbi_ecall_handler() . . . 181
26.35.2.4 sbi_ecall_init() . . . 181
26.35.2.5 sbi_ecall_register_extension() . . . 182
26.35.2.6 sbi_ecall_set_impid() . . . 182
26.35.2.7 sbi_ecall_unregister_extension() . . . 182
26.35.2.8 sbi_ecall_version_major() . . . 182
26.35.2.9 sbi_ecall_version_minor() . . . 183
26.35.3 Variable Documentation . . . 183
26.35.3.1 ecall_base . . . 183
26.35.3.2 ecall_hsm . . . 183
26.35.3.3 ecall_ipi . . . 183
26.35.3.4 ecall_legacy . . . 183
26.35.3.5 ecall_rfence . . . 183
26.35.3.6 ecall_time . . . 183
26.35.3.7 ecall_vendor . . . 184
26.36 include/sbi/sbi_ecall_interface.h File Reference . . . 184
26.36.1 Macro Definition Documentation . . . 185
26.36.1.1 SBI_ERR_ALREADY_AVAILABLE . . . 185
26.36.1.2 SBI_ERR_DENIED . . . 185
26.36.1.3 SBI_ERR_FAILED . . . 185
26.36.1.4 SBI_ERR_INVALID_ADDRESS . . . 185
26.36.1.5 SBI_ERR_INVALID_PARAM . . . 185
26.36.1.6 SBI_ERR_NOT_SUPPORTED . . . 185
26.36.1.7 SBI_EXT_0_1_CLEAR_IPI . . . 185
26.36.1.8 SBI_EXT_0_1_CONSOLE_GETCHAR . . . 185
26.36.1.9 SBI_EXT_0_1_CONSOLE_PUTCHAR . . . 186
26.36.1.10 SBI_EXT_0_1_REMOTE_FENCE_I . . . 186
26.36.1.11 SBI_EXT_0_1_REMOTE_SFENCE_VMA . . . 186
26.36.1.12 SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID . . . 186
26.36.1.13 SBI_EXT_0_1_SEND_IPI . . . 186
26.36.1.14 SBI_EXT_0_1_SET_TIMER . . . 186
26.36.1.15 SBI_EXT_0_1_SHUTDOWN . . . 186
26.36.1.16 SBI_EXT_BASE . . . 186
26.36.1.17 SBI_EXT_BASE_GET_IMP_ID . . . 186
26.36.1.18 SBI_EXT_BASE_GET_IMP_VERSION . . . 186
26.36.1.19 SBI_EXT_BASE_GET_MARCHID . . . 186
26.36.1.20 SBI_EXT_BASE_GET_MIMPID . . . 186
26.36.1.21 SBI_EXT_BASE_GET_MVENDORID . . . 187
26.36.1.22 SBI_EXT_BASE_GET_SPEC_VERSION . . . 187
26.36.1.23 SBI_EXT_BASE_PROBE_EXT . . . 187
26.36.1.24 SBI_EXT_FIRMWARE_END . . . 187

26.36.1.25 SBI_EXT_FIRMWARE_START . . . 187
26.36.1.26 SBI_EXT_HSM . . . 187
26.36.1.27 SBI_EXT_HSM_HART_GET_STATUS . . . 187
26.36.1.28 SBI_EXT_HSM_HART_START . . . 187
26.36.1.29 SBI_EXT_HSM_HART_STOP . . . 187
26.36.1.30 SBI_EXT_IPI . . . 187
26.36.1.31 SBI_EXT_IPI_SEND_IPI . . . 187
26.36.1.32 SBI_EXT_RFENCE . . . 187
26.36.1.33 SBI_EXT_RFENCE_REMOTE_FENCE_I . . . 188
26.36.1.34 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA . . . 188
26.36.1.35 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID . . . 188
26.36.1.36 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA . . . 188
26.36.1.37 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID . . . 188
26.36.1.38 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA . . . 188
26.36.1.39 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID . . . 188
26.36.1.40 SBI_EXT_TIME . . . 188
26.36.1.41 SBI_EXT_TIME_SET_TIMER . . . 188
26.36.1.42 SBI_EXT_VENDOR_END . . . 188
26.36.1.43 SBI_EXT_VENDOR_START . . . 188
26.36.1.44 SBI_HSM_HART_STATUS_START_PENDING . . . 188
26.36.1.45 SBI_HSM_HART_STATUS_STARTED . . . 189
26.36.1.46 SBI_HSM_HART_STATUS_STOP_PENDING . . . 189
26.36.1.47 SBI_HSM_HART_STATUS_STOPPED . . . 189
26.36.1.48 SBI_LAST_ERR . . . 189
26.36.1.49 SBI_SPEC_VERSION_MAJOR_MASK . . . 189
26.36.1.50 SBI_SPEC_VERSION_MAJOR_OFFSET . . . 189
26.36.1.51 SBI_SPEC_VERSION_MINOR_MASK . . . 189
26.36.1.52 SBI_SUCCESS . . . 189
26.37 include/sbi/sbi_emulate_csr.h File Reference . . . 189
26.37.1 Function Documentation . . . 190
26.37.1.1 sbi_emulate_csr_read() . . . 190
26.37.1.2 sbi_emulate_csr_write() . . . 191
26.38 include/sbi/sbi_error.h File Reference . . . 191
26.38.1 Macro Definition Documentation . . . 192
26.38.1.1 SBI_EALREADY . . . 192
26.38.1.2 SBI_EDENIED . . . 192
26.38.1.3 SBI_EFAIL . . . 192
26.38.1.4 SBI_EILL . . . 192
26.38.1.5 SBI_EINVAL . . . 192
26.38.1.6 SBI_EINVALID_ADDR . . . 192
26.38.1.7 SBI_EIO . . . 192
26.38.1.8 SBI_ENODEV . . . 193

26.38.1.9 SBI_ENOENT . . . 193
26.38.1.10 SBI_ENOMEM . . . 193
26.38.1.11 SBI_ENOSPC . . . 193
26.38.1.12 SBI_ENOSYS . . . 193
26.38.1.13 SBI_ENOTSUPP . . . 193
26.38.1.14 SBI_ETIMEDOUT . . . 193
26.38.1.15 SBI_ETRAP . . . 193
26.38.1.16 SBI_EUNKNOWN . . . 193
26.38.1.17 SBI_OK . . . 193
26.39 include/sbi/sbi_fifo.h File Reference . . . 193
26.39.1 Enumeration Type Documentation . . . 194
26.39.1.1 sbi_fifo_inplace_update_types . . . 195
26.39.2 Function Documentation . . . 196
26.39.2.1 sbi_fifo_avail() . . . 196
26.39.2.2 sbi_fifo_dequeue() . . . 196
26.39.2.3 sbi_fifo_enqueue() . . . 197
26.39.2.4 sbi_fifo_init() . . . 197
26.39.2.5 sbi_fifo_inplace_update() . . . 198
26.39.2.6 sbi_fifo_is_empty() . . . 199
26.39.2.7 sbi_fifo_is_full() . . . 199
26.40 include/sbi/sbi_hart.h File Reference . . . 199
26.40.1 Enumeration Type Documentation . . . 200
26.40.1.1 sbi_hart_features . . . 201
26.40.2 Function Documentation . . . 201
26.40.2.1 __attribute__() . . . 201
26.40.2.2 sbi_hart_delegation_dump() . . . 201
26.40.2.3 sbi_hart_expected_trap_addr() . . . 201
26.40.2.4 sbi_hart_get_features_str() . . . 202
26.40.2.5 sbi_hart_has_feature() . . . 202
26.40.2.6 sbi_hart_init() . . . 204
26.40.2.7 sbi_hart_pmp_check_addr() . . . 205
26.40.2.8 sbi_hart_pmp_count() . . . 205
26.40.2.9 sbi_hart_pmp_dump() . . . 206
26.40.2.10 sbi_hart_pmp_get() . . . 206
26.40.3 Variable Documentation . . . 207
26.40.3.1 arg1 . . . 207
26.40.3.2 next_addr . . . 207
26.40.3.3 next_mode . . . 207
26.40.3.4 next_virt . . . 207
26.40.3.5 sbi_hart_expected_trap . . . 207
26.41 include/sbi/sbi_hartmask.h File Reference . . . 207
26.41.1 Macro Definition Documentation . . . 209

26.41.1.1 sbi_hartmask_bits . . . 209
26.41.1.2 sbi_hartmask_for_each_hart . . . 209
26.41.1.3 SBI_HARTMASK_INIT . . . 209
26.41.1.4 SBI_HARTMASK_INIT_EXCEPT . . . 209
26.41.1.5 SBI_HARTMASK_MAX_BITS . . . 209
26.41.2 Function Documentation . . . 209
26.41.2.1 sbi_hartmask_and() . . . 209
26.41.2.2 sbi_hartmask_clear_all() . . . 210
26.41.2.3 sbi_hartmask_clear_hart() . . . 210
26.41.2.4 sbi_hartmask_or() . . . 211
26.41.2.5 sbi_hartmask_set_all() . . . 212
26.41.2.6 sbi_hartmask_set_hart() . . . 212
26.41.2.7 sbi_hartmask_test_hart() . . . 213
26.41.2.8 sbi_hartmask_xor() . . . 213
26.42 include/sbi/sbi_hfence.h File Reference . . . 214
26.42.1 Function Documentation . . . 214
26.42.1.1 __sbi_hfence_gvma_all() . . . 214
26.42.1.2 __sbi_hfence_gvma_gpa() . . . 215
26.42.1.3 __sbi_hfence_gvma_vmid() . . . 215
26.42.1.4 __sbi_hfence_gvma_vmid_gpa() . . . 215
26.42.1.5 __sbi_hfence_vvma_all() . . . 215
26.42.1.6 __sbi_hfence_vvma_asid() . . . 216
26.42.1.7 __sbi_hfence_vvma_asid_va() . . . 216
26.42.1.8 __sbi_hfence_vvma_va() . . . 216
26.43 include/sbi/sbi_hsm.h File Reference . . . 217
26.43.1 Macro Definition Documentation . . . 217
26.43.1.1 SBI_HART_STARTED . . . 217
26.43.1.2 SBI_HART_STARTING . . . 218
26.43.1.3 SBI_HART_STOPPED . . . 218
26.43.1.4 SBI_HART_STOPPING . . . 218
26.43.1.5 SBI_HART_UNKNOWN . . . 218
26.43.2 Function Documentation . . . 218
26.43.2.1 sbi_hsm_exit() . . . 218
26.43.2.2 sbi_hsm_hart_get_state() . . . 218
26.43.2.3 sbi_hsm_hart_start() . . . 219
26.43.2.4 sbi_hsm_hart_started() . . . 220
26.43.2.5 sbi_hsm_hart_started_mask() . . . 220
26.43.2.6 sbi_hsm_hart_state_to_status() . . . 221
26.43.2.7 sbi_hsm_hart_stop() . . . 221
26.43.2.8 sbi_hsm_init() . . . 222
26.43.2.9 sbi_hsm_prepare_next_jump() . . . 223
26.44 include/sbi/sbi_illegal_insn.h File Reference . . . 223

26.44.1 Function Documentation . . . 224
26.44.1.1 sbi_illegal_insn_handler() . . . 224
26.45 include/sbi/sbi_init.h File Reference . . . 225
26.45.1 Function Documentation . . . 226
26.45.1.1 sbi_exit() . . . 226
26.45.1.2 sbi_init() . . . 227
26.45.1.3 sbi_init_count() . . . 228
26.46 include/sbi/sbi_ipi.h File Reference . . . 229
26.46.1 Macro Definition Documentation . . . 229
26.46.1.1 SBI_IPI_EVENT_MAX . . . 229
26.46.2 Function Documentation . . . 230
26.46.2.1 sbi_ipi_clear_smode() . . . 230
26.46.2.2 sbi_ipi_event_create() . . . 230
26.46.2.3 sbi_ipi_event_destroy() . . . 230
26.46.2.4 sbi_ipi_exit() . . . 230
26.46.2.5 sbi_ipi_init() . . . 231
26.46.2.6 sbi_ipi_process() . . . 232
26.46.2.7 sbi_ipi_send_halt() . . . 232
26.46.2.8 sbi_ipi_send_many() . . . 233
26.46.2.9 sbi_ipi_send_smode() . . . 234
26.47 include/sbi/sbi_list.h File Reference . . . 234
26.47.1 Macro Definition Documentation . . . 235
26.47.1.1 SBI_INIT_LIST_HEAD . . . 235
26.47.1.2 sbi_list_entry . . . 235
26.47.1.3 sbi_list_first_entry . . . 236
26.47.1.4 sbi_list_for_each . . . 236
26.47.1.5 sbi_list_for_each_entry . . . 236
26.47.1.6 SBI_LIST_HEAD . . . 236
26.47.1.7 SBI_LIST_HEAD_INIT . . . 236
26.47.1.8 sbi_list_last_entry . . . 237
26.47.1.9 SBI_LIST_POISON_NEXT . . . 237
26.47.1.10 SBI_LIST_POISON_PREV . . . 237
26.47.2 Function Documentation . . . 237
26.47.2.1 __sbi_list_add() . . . 237
26.47.2.2 __sbi_list_del() . . . 237
26.47.2.3 __sbi_list_del_entry() . . . 238
26.47.2.4 sbi_list_add() . . . 238
26.47.2.5 sbi_list_add_tail() . . . 239
26.47.2.6 sbi_list_del() . . . 239
26.47.2.7 sbi_list_del_init() . . . 240
26.48 include/sbi/sbi_math.h File Reference . . . 241
26.48.1 Function Documentation . . . 241

26.48.1.1 log2roundup() . . . 241
26.49 include/sbi/sbi_misaligned_ldst.h File Reference . . . 241
26.49.1 Function Documentation . . . 242
26.49.1.1 sbi_misaligned_load_handler() . . . 242
26.49.1.2 sbi_misaligned_store_handler() . . . 243
26.50 include/sbi/sbi_platform.h File Reference . . . 244
26.50.1 Macro Definition Documentation . . . 246
26.50.1.1 SBI_PLATFORM_DEFAULT_FEATURES . . . 246
26.50.1.2 SBI_PLATFORM_DEFAULT_HART_STACK_SIZE . . . 246
26.50.1.3 SBI_PLATFORM_FEATURES_OFFSET . . . 246
26.50.1.4 SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET . . . 246
26.50.1.5 SBI_PLATFORM_HART_COUNT_OFFSET . . . 246
26.50.1.6 SBI_PLATFORM_HART_INDEX2ID_OFFSET . . . 246
26.50.1.7 SBI_PLATFORM_HART_STACK_SIZE_OFFSET . . . 246
26.50.1.8 sbi_platform_has_hart_hotplug . . . 246
26.50.1.9 sbi_platform_has_hart_secondary_boot . . . 246
26.50.1.10 sbi_platform_has_mfaults_delegation . . . 246
26.50.1.11 sbi_platform_has_timer_value . . . 247
26.50.1.12 SBI_PLATFORM_NAME_OFFSET . . . 247
26.50.1.13 SBI_PLATFORM_OPENSBI_VERSION_OFFSET . . . 247
26.50.1.14 sbi_platform_ops . . . 247
26.50.1.15 SBI_PLATFORM_OPS_OFFSET . . . 247
26.50.1.16 sbi_platform_ptr . . . 247
26.50.1.17 SBI_PLATFORM_RESET_COLD . . . 247
26.50.1.18 SBI_PLATFORM_RESET_SHUTDOWN . . . 247
26.50.1.19 SBI_PLATFORM_RESET_WARM . . . 247
26.50.1.20 sbi_platform_thishart_ptr . . . 247
26.50.1.21 SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT . . . 247
26.50.1.22 SBI_PLATFORM_VERSION . . . 248
26.50.1.23 SBI_PLATFORM_VERSION_OFFSET . . . 248
26.50.2 Enumeration Type Documentation . . . 248
26.50.2.1 sbi_platform_features . . . 248
26.50.3 Function Documentation . . . 248
26.50.3.1 sbi_platform_console_getc() . . . 248
26.50.3.2 sbi_platform_console_init() . . . 249
26.50.3.3 sbi_platform_console_putc() . . . 249
26.50.3.4 sbi_platform_early_exit() . . . 250
26.50.3.5 sbi_platform_early_init() . . . 250
26.50.3.6 sbi_platform_final_exit() . . . 251
26.50.3.7 sbi_platform_final_init() . . . 251
26.50.3.8 sbi_platform_get_features() . . . 252
26.50.3.9 sbi_platform_get_features_str() . . . 252

26.50.3.10 sbi_platform_hart_count() . . . 253
26.50.3.11 sbi_platform_hart_index() . . . 253
26.50.3.12 sbi_platform_hart_invalid() . . . 254
26.50.3.13 sbi_platform_hart_stack_size() . . . 255
26.50.3.14 sbi_platform_hart_start() . . . 255
26.50.3.15 sbi_platform_hart_stop() . . . 255
26.50.3.16 sbi_platform_ipi_clear() . . . 256
26.50.3.17 sbi_platform_ipi_exit() . . . 256
26.50.3.18 sbi_platform_ipi_init() . . . 257
26.50.3.19 sbi_platform_ipi_send() . . . 257
26.50.3.20 sbi_platform_irqchip_exit() . . . 258
26.50.3.21 sbi_platform_irqchip_init() . . . 258
26.50.3.22 sbi_platform_misa_extension() . . . 258
26.50.3.23 sbi_platform_misa_xlen() . . . 260
26.50.3.24 sbi_platform_name() . . . 260
26.50.3.25 sbi_platform_pmp_region_count() . . . 261
26.50.3.26 sbi_platform_pmp_region_info() . . . 261
26.50.3.27 sbi_platform_system_reset() . . . 262
26.50.3.28 sbi_platform_timer_event_start() . . . 262
26.50.3.29 sbi_platform_timer_event_stop() . . . 263
26.50.3.30 sbi_platform_timer_exit() . . . 263
26.50.3.31 sbi_platform_timer_init() . . . 264
26.50.3.32 sbi_platform_timer_value() . . . 264
26.50.3.33 sbi_platform_tlbr_flush_limit() . . . 265
26.50.3.34 sbi_platform_vendor_ext_check() . . . 265
26.50.3.35 sbi_platform_vendor_ext_provider() . . . 266
26.50.4 Variable Documentation . . . 266
26.50.4.1 __packed . . . 267
26.51 include/sbi/sbi_scratch.h File Reference . . . 267
26.51.1 Macro Definition Documentation . . . 268
26.51.1.1 sbi_hartid_to_scratch . . . 268
26.51.1.2 SBI_SCRATCH_EXTRA_SPACE_OFFSET . . . 268
26.51.1.3 SBI_SCRATCH_FW_SIZE_OFFSET . . . 268
26.51.1.4 SBI_SCRATCH_FW_START_OFFSET . . . 268
26.51.1.5 SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET . . . 268
26.51.1.6 sbi_scratch_last_hartid . . . 269
26.51.1.7 SBI_SCRATCH_NEXT_ADDR_OFFSET . . . 269
26.51.1.8 SBI_SCRATCH_NEXT_ARG1_OFFSET . . . 269
26.51.1.9 SBI_SCRATCH_NEXT_MODE_OFFSET . . . 269
26.51.1.10 sbi_scratch_offset_ptr . . . 269
26.51.1.11 SBI_SCRATCH_OPTIONS_OFFSET . . . 269
26.51.1.12 SBI_SCRATCH_PLATFORM_ADDR_OFFSET . . . 269

26.51.1.13 SBI_SCRATCH_SIZE . . . 269
26.51.1.14 sbi_scratch_thishart_arg1_ptr . . . 269
26.51.1.15 sbi_scratch_thishart_offset_ptr . . . 269
26.51.1.16 sbi_scratch_thishart_ptr . . . 269
26.51.1.17 SBI_SCRATCH_TMP0_OFFSET . . . 270
26.51.1.18 SBI_SCRATCH_WARMBOOT_ADDR_OFFSET . . . 270
26.51.2 Enumeration Type Documentation . . . 270
26.51.2.1 sbi_scratch_options . . . 270
26.51.3 Function Documentation . . . 270
26.51.3.1 sbi_scratch_alloc_offset() . . . 270
26.51.3.2 sbi_scratch_free_offset() . . . 271
26.51.3.3 sbi_scratch_init() . . . 271
26.51.4 Variable Documentation . . . 272
26.51.4.1 __packed . . . 272
26.51.4.2 hartid_to_scratch_table . . . 272
26.51.4.3 last_hartid_having_scratch . . . 272
26.52 include/sbi/sbi_string.h File Reference . . . 272
26.52.1 Function Documentation . . . 273
26.52.1.1 sbi_memchr() . . . 273
26.52.1.2 sbi_memcmp() . . . 273
26.52.1.3 sbi_memcpy() . . . 273
26.52.1.4 sbi_memmove() . . . 274
26.52.1.5 sbi_memset() . . . 274
26.52.1.6 sbi_strchr() . . . 274
26.52.1.7 sbi_strcmp() . . . 274
26.52.1.8 sbi_strcpy() . . . 275
26.52.1.9 sbi_strlen() . . . 275
26.52.1.10 sbi_strncpy() . . . 275
26.52.1.11 sbi_strnlen() . . . 276
26.52.1.12 sbi_strrchr() . . . 276
26.53 include/sbi/sbi_system.h File Reference . . . 276
26.53.1 Function Documentation . . . 277
26.53.1.1 sbi_system_reset() . . . 277
26.54 include/sbi/sbi_timer.h File Reference . . . 277
26.54.1 Function Documentation . . . 278
26.54.1.1 sbi_timer_event_start() . . . 278
26.54.1.2 sbi_timer_exit() . . . 279
26.54.1.3 sbi_timer_get_delta() . . . 279
26.54.1.4 sbi_timer_init() . . . 279
26.54.1.5 sbi_timer_process() . . . 280
26.54.1.6 sbi_timer_set_delta() . . . 280
26.54.1.7 sbi_timer_set_delta_upper() . . . 281

26.54.1.8 sbi_timer_value() . . . 281
26.54.1.9 sbi_timer_virt_value() . . . 281
26.55 include/sbi/sbi_tlb.h File Reference . . . 282
26.55.1 Macro Definition Documentation . . . 283
26.55.1.1 SBI_TLB_FIFO_NUM_ENTRIES . . . 283
26.55.1.2 SBI_TLB_FLUSH_ALL . . . 283
26.55.1.3 SBI_TLB_INFO_INIT . . . 283
26.55.1.4 SBI_TLB_INFO_SIZE . . . 283
26.55.2 Enumeration Type Documentation . . . 284
26.55.2.1 sbi_tlb_info_types . . . 284
26.55.3 Function Documentation . . . 284
26.55.3.1 sbi_tlb_init() . . . 284
26.55.3.2 sbi_tlb_request() . . . 285
26.56 include/sbi/sbi_trap.h File Reference . . . 285
26.56.1 Macro Definition Documentation . . . 287
26.56.1.1 SBI_TRAP_INFO_cause . . . 287
26.56.1.2 SBI_TRAP_INFO_epc . . . 287
26.56.1.3 SBI_TRAP_INFO_last . . . 287
26.56.1.4 SBI_TRAP_INFO_OFFSET . . . 287
26.56.1.5 SBI_TRAP_INFO_SIZE . . . 288
26.56.1.6 SBI_TRAP_INFO_tinst . . . 288
26.56.1.7 SBI_TRAP_INFO_tval . . . 288
26.56.1.8 SBI_TRAP_INFO_tval2 . . . 288
26.56.1.9 SBI_TRAP_REGS_a0 . . . 288
26.56.1.10 SBI_TRAP_REGS_a1 . . . 288
26.56.1.11 SBI_TRAP_REGS_a2 . . . 288
26.56.1.12 SBI_TRAP_REGS_a3 . . . 288
26.56.1.13 SBI_TRAP_REGS_a4 . . . 288
26.56.1.14 SBI_TRAP_REGS_a5 . . . 288
26.56.1.15 SBI_TRAP_REGS_a6 . . . 288
26.56.1.16 SBI_TRAP_REGS_a7 . . . 288
26.56.1.17 SBI_TRAP_REGS_gp . . . 289
26.56.1.18 SBI_TRAP_REGS_last . . . 289
26.56.1.19 SBI_TRAP_REGS_mepc . . . 289
26.56.1.20 SBI_TRAP_REGS_mstatus . . . 289
26.56.1.21 SBI_TRAP_REGS_mstatusH . . . 289
26.56.1.22 SBI_TRAP_REGS_OFFSET . . . 289
26.56.1.23 SBI_TRAP_REGS_ra . . . 289
26.56.1.24 SBI_TRAP_REGS_s0 . . . 289
26.56.1.25 SBI_TRAP_REGS_s1 . . . 289
26.56.1.26 SBI_TRAP_REGS_s10 . . . 289
26.56.1.27 SBI_TRAP_REGS_s11 . . . 289

26.56.1.28 SBI_TRAP_REGS_s2 . . . 289
26.56.1.29 SBI_TRAP_REGS_s3 . . . 290
26.56.1.30 SBI_TRAP_REGS_s4 . . . 290
26.56.1.31 SBI_TRAP_REGS_s5 . . . 290
26.56.1.32 SBI_TRAP_REGS_s6 . . . 290
26.56.1.33 SBI_TRAP_REGS_s7 . . . 290
26.56.1.34 SBI_TRAP_REGS_s8 . . . 290
26.56.1.35 SBI_TRAP_REGS_s9 . . . 290
26.56.1.36 SBI_TRAP_REGS_SIZE . . . 290
26.56.1.37 SBI_TRAP_REGS_sp . . . 290
26.56.1.38 SBI_TRAP_REGS_t0 . . . 290
26.56.1.39 SBI_TRAP_REGS_t1 . . . 290
26.56.1.40 SBI_TRAP_REGS_t2 . . . 290
26.56.1.41 SBI_TRAP_REGS_t3 . . . 291
26.56.1.42 SBI_TRAP_REGS_t4 . . . 291
26.56.1.43 SBI_TRAP_REGS_t5 . . . 291
26.56.1.44 SBI_TRAP_REGS_t6 . . . 291
26.56.1.45 SBI_TRAP_REGS_tp . . . 291
26.56.1.46 SBI_TRAP_REGS_zero . . . 291
26.56.2 Function Documentation . . . 291
26.56.2.1 sbi_trap_handler() . . . 291
26.56.2.2 sbi_trap_redirect() . . . 292
26.56.3 Variable Documentation . . . 292
26.56.3.1 __packed . . . 293
26.57 include/sbi/sbi_types.h File Reference . . . 293
26.57.1 Macro Definition Documentation . . . 294
26.57.1.1 __noreturn . . . 294
26.57.1.2 __packed . . . 294
26.57.1.3 array_size . . . 294
26.57.1.4 CLAMP . . . 294
26.57.1.5 container_of . . . 294
26.57.1.6 FALSE . . . 294
26.57.1.7 false . . . 294
26.57.1.8 likely . . . 294
26.57.1.9 MAX . . . 294
26.57.1.10 MIN . . . 295
26.57.1.11 NULL . . . 295
26.57.1.12 offsetof . . . 295
26.57.1.13 ROUNDDOWN . . . 295
26.57.1.14 ROUNDUP . . . 295
26.57.1.15 STR . . . 295
26.57.1.16 TRUE . . . 295

26.57.1.17 true . . . 295
26.57.1.18 unlikely . . . 295
26.57.1.19 XSTR . . . 295
26.57.2 Typedef Documentation . . . 296
26.57.2.1 bool . . . 296
26.57.2.2 int16_t . . . 296
26.57.2.3 int32_t . . . 296
26.57.2.4 physical_addr_t . . . 296
26.57.2.5 physical_size_t . . . 296
26.57.2.6 s16 . . . 296
26.57.2.7 s32 . . . 296
26.57.2.8 s8 . . . 296
26.57.2.9 size_t . . . 296
26.57.2.10 ssize_t . . . 296
26.57.2.11 u16 . . . 296
26.57.2.12 u32 . . . 297
26.57.2.13 u8 . . . 297
26.57.2.14 uint16_t . . . 297
26.57.2.15 uint32_t . . . 297
26.57.2.16 uint8_t . . . 297
26.57.2.17 uintptr_t . . . 297
26.57.2.18 ulong . . . 297
26.57.2.19 virtual_addr_t . . . 297
26.57.2.20 virtual_size_t . . . 297
26.58 include/sbi/sbi_unpriv.h File Reference . . . 297
26.58.1 Macro Definition Documentation . . . 298
26.58.1.1 DECLARE_UNPRIVILEGED_LOAD_FUNCTION . . . 298
26.58.1.2 DECLARE_UNPRIVILEGED_STORE_FUNCTION . . . 298
26.58.2 Function Documentation . . . 299
26.58.2.1 sbi_get_insn() . . . 299
26.59 include/sbi/sbi_version.h File Reference . . . 299
26.59.1 Macro Definition Documentation . . . 299
26.59.1.1 OPENSBI_VERSION . . . 300
26.59.1.2 OPENSBI_VERSION_MAJOR . . . 300
26.59.1.3 OPENSBI_VERSION_MINOR . . . 300
26.60 include/sbi_utils/fdt/fdt_fixup.h File Reference . . . 300
26.60.1 Function Documentation . . . 300
26.60.1.1 fdt_cpu_fixup() . . . 300
26.60.1.2 fdt_fixups() . . . 301
26.60.1.3 fdt_plic_fixup() . . . 301
26.60.1.4 fdt_reserved_memory_fixup() . . . 302
26.60.1.5 fdt_reserved_memory_nomap_fixup() . . . 303

26.61 include/sbi_utils/fdt/fdt_helper.h File Reference . . . 304
26.61.1 Function Documentation . . . 305
26.61.1.1 fdt_find_match() . . . 305
26.61.1.2 fdt_get_node_addr_size() . . . 305
26.61.1.3 fdt_match_node() . . . 306
26.61.1.4 fdt_parse_clint_node() . . . 307
26.61.1.5 fdt_parse_compat_addr() . . . 307
26.61.1.6 fdt_parse_hart_id() . . . 308
26.61.1.7 fdt_parse_max_hart_id() . . . 308
26.61.1.8 fdt_parse_plic() . . . 309
26.61.1.9 fdt_parse_plic_node() . . . 309
26.61.1.10 fdt_parse_shakti_uart_node() . . . 310
26.61.1.11 fdt_parse_sifive_uart_node() . . . 310
26.61.1.12 fdt_parse_uart8250() . . . 311
26.61.1.13 fdt_parse_uart8250_node() . . . 311
26.62 include/sbi_utils/ipi/fdt_ipi.h File Reference . . . 312
26.62.1 Function Documentation . . . 313
26.62.1.1 fdt_ipi_clear() . . . 313
26.62.1.2 fdt_ipi_exit() . . . 313
26.62.1.3 fdt_ipi_init() . . . 313
26.62.1.4 fdt_ipi_send() . . . 314
26.63 include/sbi_utils/irqchip/fdt_irqchip.h File Reference . . . 314
26.63.1 Function Documentation . . . 315
26.63.1.1 fdt_irqchip_exit() . . . 315
26.63.1.2 fdt_irqchip_init() . . . 315
26.64 include/sbi_utils/irqchip/plic.h File Reference . . . 315
26.64.1 Function Documentation . . . 316
26.64.1.1 plic_cold_irqchip_init() . . . 316
26.64.1.2 plic_set_ie() . . . 316
26.64.1.3 plic_set_thresh() . . . 317
26.64.1.4 plic_warm_irqchip_init() . . . 317
26.65 include/sbi_utils/reset/fdt_reset.h File Reference . . . 318
26.65.1 Function Documentation . . . 319
26.65.1.1 fdt_reset_init() . . . 319
26.65.1.2 fdt_system_reset() . . . 319
26.66 include/sbi_utils/serial/fdt_serial.h File Reference . . . 319
26.66.1 Function Documentation . . . 320
26.66.1.1 fdt_serial_getc() . . . 320
26.66.1.2 fdt_serial_init() . . . 320
26.66.1.3 fdt_serial_putc() . . . 320
26.67 include/sbi_utils/serial/shakti-uart.h File Reference . . . 320
26.67.1 Function Documentation . . . 321

26.67.1.1 shakti_uart_getc() . . . 321
26.67.1.2 shakti_uart_init() . . . 321
26.67.1.3 shakti_uart_putc() . . . 322
26.68 include/sbi_utils/serial/sifive-uart.h File Reference . . . 322
26.68.1 Function Documentation . . . 323
26.68.1.1 sifive_uart_getc() . . . 323
26.68.1.2 sifive_uart_init() . . . 323
26.68.1.3 sifive_uart_putc() . . . 324
26.69 include/sbi_utils/serial/uart8250.h File Reference . . . 324
26.69.1 Function Documentation . . . 325
26.69.1.1 uart8250_getc() . . . 325
26.69.1.2 uart8250_init() . . . 325
26.69.1.3 uart8250_putc() . . . 326
26.70 include/sbi_utils/sys/clint.h File Reference . . . 326
26.70.1 Function Documentation . . . 327
26.70.1.1 clint_cold_ipi_init() . . . 327
26.70.1.2 clint_cold_timer_init() . . . 328
26.70.1.3 clint_ipi_clear() . . . 328
26.70.1.4 clint_ipi_send() . . . 329
26.70.1.5 clint_timer_event_start() . . . 329
26.70.1.6 clint_timer_event_stop() . . . 329
26.70.1.7 clint_timer_value() . . . 329
26.70.1.8 clint_warm_ipi_init() . . . 329
26.70.1.9 clint_warm_timer_init() . . . 330
26.71 include/sbi_utils/sys/htif.h File Reference . . . 330
26.71.1 Function Documentation . . . 331
26.71.1.1 htif_getc() . . . 331
26.71.1.2 htif_putc() . . . 331
26.71.1.3 htif_system_reset() . . . 331
26.72 include/sbi_utils/sys/sifive_test.h File Reference . . . 332
26.72.1 Function Documentation . . . 332
26.72.1.1 sifive_test_init() . . . 332
26.72.1.2 sifive_test_system_reset() . . . 333
26.73 include/sbi_utils/timer/fdt_timer.h File Reference . . . 333
26.73.1 Function Documentation . . . 334
26.73.1.1 fdt_timer_event_start() . . . 334
26.73.1.2 fdt_timer_event_stop() . . . 334
26.73.1.3 fdt_timer_exit() . . . 334
26.73.1.4 fdt_timer_init() . . . 334
26.73.1.5 fdt_timer_value() . . . 334
26.74 lib/sbi/riscv_asm.c File Reference . . . 335
26.74.1 Function Documentation . . . 335

26.74.1.1 csr_read_num() . . . 335
26.74.1.2 csr_write_num() . . . 336
26.74.1.3 ctz() . . . 336
26.74.1.4 misa_extension_imp() . . . 336
26.74.1.5 misa_string() . . . 337
26.74.1.6 misa_xlen() . . . 337
26.74.1.7 pmp_get() . . . 337
26.74.1.8 pmp_set() . . . 338
26.75 lib/sbi/riscv_atomic.c File Reference . . . 339
26.75.1 Macro Definition Documentation . . . 340
26.75.1.1 __atomic_op_bit . . . 340
26.75.1.2 __atomic_op_bit_ord . . . 340
26.75.1.3 __axchg . . . 340
26.75.1.4 __cmpxchg . . . 341
26.75.1.5 __NOP . . . 341
26.75.1.6 __NOT . . . 341
26.75.1.7 __xchg . . . 341
26.75.1.8 axchg . . . 341
26.75.1.9 cmpxchg . . . 341
26.75.1.10 xchg . . . 342
26.75.2 Function Documentation . . . 342
26.75.2.1 atomic_add_return() . . . 342
26.75.2.2 atomic_clear_bit() . . . 342
26.75.2.3 atomic_cmpxchg() . . . 342
26.75.2.4 atomic_raw_clear_bit() . . . 343
26.75.2.5 atomic_raw_set_bit() . . . 343
26.75.2.6 atomic_raw_xchg_uint() . . . 343
26.75.2.7 atomic_raw_xchg_ulong() . . . 343
26.75.2.8 atomic_read() . . . 344
26.75.2.9 atomic_set_bit() . . . 344
26.75.2.10 atomic_sub_return() . . . 344
26.75.2.11 atomic_write() . . . 345
26.75.2.12 atomic_xchg() . . . 345
26.76 lib/sbi/riscv_locks.c File Reference . . . 345
26.76.1 Function Documentation . . . 345
26.76.1.1 spin_lock() . . . 346
26.76.1.2 spin_lock_check() . . . 346
26.76.1.3 spin_trylock() . . . 347
26.76.1.4 spin_unlock() . . . 347
26.77 lib/sbi/sbi_bitmap.c File Reference . . . 348
26.77.1 Function Documentation . . . 349
26.77.1.1 __bitmap_and() . . . 349

26.77.1.2 __bitmap_or() . . . 349
26.77.1.3 __bitmap_xor() . . . 349
26.78 lib/sbi/sbi_bitops.c File Reference . . . 350
26.78.1 Macro Definition Documentation . . . 350
26.78.1.1 BITOP_WORD . . . 350
26.78.2 Function Documentation . . . 351
26.78.2.1 find_first_bit() . . . 351
26.78.2.2 find_first_zero_bit() . . . 351
26.78.2.3 find_last_bit() . . . 351
26.78.2.4 find_next_bit() . . . 351
26.78.2.5 find_next_zero_bit() . . . 352
26.79 lib/sbi/sbi_console.c File Reference . . . 352
26.79.1 Macro Definition Documentation . . . 353
26.79.1.1 PAD_ALTERNATE . . . 353
26.79.1.2 PAD_RIGHT . . . 353
26.79.1.3 PAD_ZERO . . . 353
26.79.1.4 PRINT_BUF_LEN . . . 353
26.79.1.5 va_arg . . . 353
26.79.1.6 va_end . . . 354
26.79.1.7 va_start . . . 354
26.79.2 Typedef Documentation . . . 354
26.79.2.1 va_list . . . 354
26.79.3 Function Documentation . . . 354
26.79.3.1 print() . . . 354
26.79.3.2 printc() . . . 355
26.79.3.3 printi() . . . 355
26.79.3.4 prints() . . . 356
26.79.3.5 sbi_console_init() . . . 357
26.79.3.6 sbi_dprintf() . . . 357
26.79.3.7 sbi_getc() . . . 358
26.79.3.8 sbi_gets() . . . 358
26.79.3.9 sbi_isprintable() . . . 358
26.79.3.10 sbi_printf() . . . 359
26.79.3.11 sbi_putc() . . . 359
26.79.3.12 sbi_puts() . . . 360
26.79.3.13 sbi_snprintf() . . . 360
26.79.3.14 sbi_sprintf() . . . 361
26.79.4 Variable Documentation . . . 361
26.79.4.1 console_out_lock . . . 361
26.79.4.2 console_plat . . . 361
26.80 lib/sbi/sbi_ecall.c File Reference . . . 361
26.80.1 Function Documentation . . . 362

26.80.1.1 sbi_ecall_find_extension() . . . 362
26.80.1.2 sbi_ecall_get_impid() . . . 363
26.80.1.3 sbi_ecall_handler() . . . 363
26.80.1.4 sbi_ecall_init() . . . 363
26.80.1.5 sbi_ecall_register_extension() . . . 364
26.80.1.6 sbi_ecall_set_impid() . . . 364
26.80.1.7 sbi_ecall_unregister_extension() . . . 364
26.80.1.8 sbi_ecall_version_major() . . . 365
26.80.1.9 sbi_ecall_version_minor() . . . 365
26.80.1.10 SBI_LIST_HEAD() . . . 365
26.80.2 Variable Documentation . . . 365
26.80.2.1 ecall_impid . . . 365
26.81 lib/sbi/sbi_ecall_base.c File Reference . . . 366
26.81.1 Function Documentation . . . 366
26.81.1.1 sbi_ecall_base_handler() . . . 366
26.81.1.2 sbi_ecall_base_probe() . . . 367
26.81.2 Variable Documentation . . . 367
26.81.2.1 ecall_base . . . 367
26.82 lib/sbi/sbi_ecall_hsm.c File Reference . . . 367
26.82.1 Function Documentation . . . 368
26.82.1.1 sbi_ecall_hsm_handler() . . . 368
26.82.2 Variable Documentation . . . 368
26.82.2.1 ecall_hsm . . . 369
26.83 lib/sbi/sbi_ecall_legacy.c File Reference . . . 369
26.83.1 Function Documentation . . . 369
26.83.1.1 sbi_ecall_legacy_handler() . . . 369
26.83.1.2 sbi_load_hart_mask_unpriv() . . . 370
26.83.2 Variable Documentation . . . 370
26.83.2.1 ecall_legacy . . . 370
26.84 lib/sbi/sbi_ecall_replace.c File Reference . . . 371
26.84.1 Function Documentation . . . 371
26.84.1.1 sbi_ecall_ipi_handler() . . . 371
26.84.1.2 sbi_ecall_rfence_handler() . . . 372
26.84.1.3 sbi_ecall_time_handler() . . . 372
26.84.2 Variable Documentation . . . 372
26.84.2.1 ecall_ipi . . . 372
26.84.2.2 ecall_rfence . . . 373
26.84.2.3 ecall_time . . . 373
26.85 lib/sbi/sbi_ecall_vendor.c File Reference . . . 373
26.85.1 Function Documentation . . . 374
26.85.1.1 sbi_ecall_vendor_handler() . . . 374
26.85.1.2 sbi_ecall_vendor_probe() . . . 374

26.85.2 Variable Documentation . . . 374
26.85.2.1 ecall_vendor . . . 374
26.86 lib/sbi/sbi_emulate_csr.c File Reference . . . 375
26.86.1 Function Documentation . . . 375
26.86.1.1 sbi_emulate_csr_read() . . . 375
26.86.1.2 sbi_emulate_csr_write() . . . 376
26.87 lib/sbi/sbi_fifo.c File Reference . . . 376
26.87.1 Function Documentation . . . 377
26.87.1.1 __sbi_fifo_enqueue() . . . 377
26.87.1.2 __sbi_fifo_is_empty() . . . 378
26.87.1.3 __sbi_fifo_is_full() . . . 378
26.87.1.4 __sbi_fifo_reset() . . . 378
26.87.1.5 sbi_fifo_avail() . . . 379
26.87.1.6 sbi_fifo_dequeue() . . . 379
26.87.1.7 sbi_fifo_enqueue() . . . 380
26.87.1.8 sbi_fifo_init() . . . 381
26.87.1.9 sbi_fifo_inplace_update() . . . 381
26.87.1.10 sbi_fifo_is_empty() . . . 382
26.87.1.11 sbi_fifo_is_full() . . . 382
26.87.1.12 sbi_fifo_reset() . . . 383
26.88 lib/sbi/sbi_hart.c File Reference . . . 383
26.88.1 Macro Definition Documentation . . . 384
26.88.1.1 __detect_pmp . . . 384
26.88.2 Function Documentation . . . 385
26.88.2.1 __attribute__() . . . 385
26.88.2.2 __sbi_expected_trap() . . . 385
26.88.2.3 __sbi_expected_trap_hext() . . . 385
26.88.2.4 delegate_traps() . . . 385
26.88.2.5 fp_init() . . . 385
26.88.2.6 hart_detect_features() . . . 386
26.88.2.7 hart_get_features() . . . 386
26.88.2.8 mstatus_init() . . . 386
26.88.2.9 pmp_init() . . . 387
26.88.2.10 sbi_hart_delegation_dump() . . . 388
26.88.2.11 sbi_hart_feature_id2string() . . . 388
26.88.2.12 sbi_hart_get_features_str() . . . 388
26.88.2.13 sbi_hart_has_feature() . . . 389
26.88.2.14 sbi_hart_init() . . . 390
26.88.2.15 sbi_hart_pmp_check_addr() . . . 391
26.88.2.16 sbi_hart_pmp_count() . . . 391
26.88.2.17 sbi_hart_pmp_dump() . . . 392
26.88.2.18 sbi_hart_pmp_get() . . . 392

26.88.3 Variable Documentation . . . 393
26.88.3.1 hart_features_offset . . . 393
26.88.3.2 sbi_hart_expected_trap . . . 393
26.89 lib/sbi/sbi_hsm.c File Reference . . . 393
26.89.1 Function Documentation . . . 394
26.89.1.1 sbi_hsm_exit() . . . 394
26.89.1.2 sbi_hsm_hart_get_state() . . . 395
26.89.1.3 sbi_hsm_hart_start() . . . 395
26.89.1.4 sbi_hsm_hart_started() . . . 396
26.89.1.5 sbi_hsm_hart_started_mask() . . . 397
26.89.1.6 sbi_hsm_hart_state_to_status() . . . 397
26.89.1.7 sbi_hsm_hart_stop() . . . 398
26.89.1.8 sbi_hsm_hart_wait() . . . 398
26.89.1.9 sbi_hsm_init() . . . 399
26.89.1.10 sbi_hsm_prepare_next_jump() . . . 399
26.89.2 Variable Documentation . . . 400
26.89.2.1 hart_data_offset . . . 400
26.90 lib/sbi/sbi_illegal_insn.c File Reference . . . 400
26.90.1 Typedef Documentation . . . 401
26.90.1.1 illegal_insn_func . . . 401
26.90.2 Function Documentation . . . 401
26.90.2.1 sbi_illegal_insn_handler() . . . 401
26.90.2.2 system_opcode_insn() . . . 401
26.90.2.3 truly_illegal_insn() . . . 402
26.90.3 Variable Documentation . . . 402
26.90.3.1 illegal_insn_table . . . 402
26.91 lib/sbi/sbi_init.c File Reference . . . 402
26.91.1 Macro Definition Documentation . . . 403
26.91.1.1 BANNER . . . 403
26.91.2 Function Documentation . . . 404
26.91.2.1 init_coldboot() . . . 404
26.91.2.2 init_warmboot() . . . 405
26.91.2.3 sbi_boot_prints() . . . 406
26.91.2.4 sbi_exit() . . . 406
26.91.2.5 sbi_init() . . . 408
26.91.2.6 sbi_init_count() . . . 409
26.91.2.7 wait_for_coldboot() . . . 410
26.91.2.8 wake_coldboot_harts() . . . 410
26.91.3 Variable Documentation . . . 411
26.91.3.1 coldboot_done . . . 411
26.91.3.2 coldboot_lock . . . 411
26.91.3.3 coldboot_lottery . . . 411

26.91.3.4 coldboot_wait_hmask . . . 411
26.91.3.5 init_count_offset . . . 411
26.92 lib/sbi/sbi_ipi.c File Reference . . . 412
26.92.1 Function Documentation . . . 412
26.92.1.1 sbi_ipi_clear_smode() . . . 413
26.92.1.2 sbi_ipi_event_create() . . . 413
26.92.1.3 sbi_ipi_event_destroy() . . . 413
26.92.1.4 sbi_ipi_exit() . . . 413
26.92.1.5 sbi_ipi_init() . . . 414
26.92.1.6 sbi_ipi_process() . . . 414
26.92.1.7 sbi_ipi_process_halt() . . . 415
26.92.1.8 sbi_ipi_process_smode() . . . 415
26.92.1.9 sbi_ipi_send() . . . 415
26.92.1.10 sbi_ipi_send_halt() . . . 416
26.92.1.11 sbi_ipi_send_many() . . . 417
26.92.1.12 sbi_ipi_send_smode() . . . 417
26.92.2 Variable Documentation . . . 418
26.92.2.1 ipi_data_off . . . 418
26.92.2.2 ipi_halt_event . . . 418
26.92.2.3 ipi_halt_ops . . . 418
26.92.2.4 ipi_ops_array . . . 418
26.92.2.5 ipi_smode_event . . . 418
26.92.2.6 ipi_smode_ops . . . 418
26.93 lib/sbi/sbi_math.c File Reference . . . 419
26.93.1 Function Documentation . . . 419
26.93.1.1 log2roundup() . . . 419
26.94 lib/sbi/sbi_misaligned_ldst.c File Reference . . . 419
26.94.1 Function Documentation . . . 420
26.94.1.1 sbi_misaligned_load_handler() . . . 420
26.94.1.2 sbi_misaligned_store_handler() . . . 420
26.95 lib/sbi/sbi_platform.c File Reference . . . 421
26.95.1 Function Documentation . . . 421
26.95.1.1 sbi_platform_feature_id2string() . . . 421
26.95.1.2 sbi_platform_get_features_str() . . . 422
26.95.1.3 sbi_platform_hart_index() . . . 422
26.96 lib/sbi/sbi_scratch.c File Reference . . . 423
26.96.1 Typedef Documentation . . . 424
26.96.1.1 hartid2scratch . . . 424
26.96.2 Function Documentation . . . 424
26.96.2.1 sbi_scratch_alloc_offset() . . . 424
26.96.2.2 sbi_scratch_free_offset() . . . 425
26.96.2.3 sbi_scratch_init() . . . 425

26.96.3 Variable Documentation . . . 425
26.96.3.1 extra_lock . . . 425
26.96.3.2 extra_offset . . . 425
26.96.3.3 hartid_to_scratch_table . . . 426
26.96.3.4 last_hartid_having_scratch . . . 426
26.97 lib/sbi/sbi_string.c File Reference . . . 426
26.97.1 Function Documentation . . . 426
26.97.1.1 sbi_memchr() . . . 427
26.97.1.2 sbi_memcmp() . . . 427
26.97.1.3 sbi_memcpy() . . . 427
26.97.1.4 sbi_memmove() . . . 427
26.97.1.5 sbi_memset() . . . 427
26.97.1.6 sbi_strchr() . . . 428
26.97.1.7 sbi_strcmp() . . . 428
26.97.1.8 sbi_strcpy() . . . 428
26.97.1.9 sbi_strlen() . . . 429
26.97.1.10 sbi_strncpy() . . . 429
26.97.1.11 sbi_strnlen() . . . 429
26.97.1.12 sbi_strrchr() . . . 429
26.98 lib/sbi/sbi_system.c File Reference . . . 430
26.98.1 Function Documentation . . . 430
26.98.1.1 sbi_system_reset() . . . 430
26.99 lib/sbi/sbi_timer.c File Reference . . . 431
26.99.1 Function Documentation . . . 432
26.99.1.1 get_ticks() . . . 432
26.99.1.2 sbi_timer_event_start() . . . 432
26.99.1.3 sbi_timer_exit() . . . 432
26.99.1.4 sbi_timer_get_delta() . . . 433
26.99.1.5 sbi_timer_init() . . . 433
26.99.1.6 sbi_timer_process() . . . 434
26.99.1.7 sbi_timer_set_delta() . . . 434
26.99.1.8 sbi_timer_set_delta_upper() . . . 435
26.99.1.9 sbi_timer_value() . . . 435
26.99.1.10 sbi_timer_virt_value() . . . 435
26.99.2 Variable Documentation . . . 436
26.99.2.1 get_time_val . . . 436
26.99.2.2 time_delta_off . . . 436
26.100 lib/sbi/sbi_tlb.c File Reference . . . 436
26.100.1 Function Documentation . . . 437
26.100.1.1 __sbi_tlb_range_check() . . . 437
26.100.1.2 sbi_tlb_entry_process() . . . 438
26.100.1.3 sbi_tlb_flush_all() . . . 438

26.100.1.4 sbi_tlb_hfence_gvma() . . . 438
26.100.1.5 sbi_tlb_hfence_gvma_vmid() . . . 439
26.100.1.6 sbi_tlb_hfence_vvma() . . . 440
26.100.1.7 sbi_tlb_hfence_vvma_asid() . . . 440
26.100.1.8 sbi_tlb_init() . . . 441
26.100.1.9 sbi_tlb_local_flush() . . . 441
26.100.1.10 sbi_tlb_process() . . . 442
26.100.1.11 sbi_tlb_process_count() . . . 442
26.100.1.12 sbi_tlb_request() . . . 443
26.100.1.13 sbi_tlb_sfence_vma() . . . 444
26.100.1.14 sbi_tlb_sfence_vma_asid() . . . 444
26.100.1.15 sbi_tlb_sync() . . . 445
26.100.1.16 sbi_tlb_update() . . . 445
26.100.1.17 sbi_tlb_update_cb() . . . 446
26.100.2 Variable Documentation . . . 446
26.100.2.1 tlb_event . . . 446
26.100.2.2 tlb_fifo_mem_off . . . 447
26.100.2.3 tlb_fifo_off . . . 447
26.100.2.4 tlb_ops . . . 447
26.100.2.5 tlb_range_flush_limit . . . 447
26.100.2.6 tlb_sync_off . . . 447
26.101 lib/sbi/sbi_trap.c File Reference . . . 447
26.101.1 Function Documentation . . . 448
26.101.1.1 sbi_trap_error() . . . 448
26.101.1.2 sbi_trap_handler() . . . 448
26.101.1.3 sbi_trap_redirect() . . . 449
26.102 lib/sbi/sbi_unpriv.c File Reference . . . 450
26.102.1 Macro Definition Documentation . . . 450
26.102.1.1 DEFINE_UNPRIVILEGED_LOAD_FUNCTION . . . 450
26.102.1.2 DEFINE_UNPRIVILEGED_STORE_FUNCTION . . . 451
26.102.2 Function Documentation . . . 451
26.102.2.1 sbi_get_insn() . . . 451
26.102.2.2 sbi_load_u64() . . . 452
26.102.2.3 sbi_store_u64() . . . 452
26.103 lib/utils/fdt/fdt_fixup.c File Reference . . . 452
26.103.1 Function Documentation . . . 453
26.103.1.1 fdt_cpu_fixup() . . . 453
26.103.1.2 fdt_fixups() . . . 453
26.103.1.3 fdt_plic_fixup() . . . 454
26.103.1.4 fdt_reserved_memory_fixup() . . . 455
26.103.1.5 fdt_reserved_memory_nomap_fixup() . . . 455
26.103.1.6 fdt_resv_memory_update_node() . . . 456

26.104 lib/utils/fdt/fdt_helper.c File Reference . . . 456
26.104.1 Macro Definition Documentation . . . 457
26.104.1.1 DEFAULT_SHAKTI_UART_BAUD . . . 457
26.104.1.2 DEFAULT_SHAKTI_UART_FREQ . . . 457
26.104.1.3 DEFAULT_SIFIVE_UART_BAUD . . . 458
26.104.1.4 DEFAULT_SIFIVE_UART_FREQ . . . 458
26.104.1.5 DEFAULT_SIFIVE_UART_REG_IO_WIDTH . . . 458
26.104.1.6 DEFAULT_SIFIVE_UART_REG_SHIFT . . . 458
26.104.1.7 DEFAULT_UART_BAUD . . . 458
26.104.1.8 DEFAULT_UART_FREQ . . . 458
26.104.1.9 DEFAULT_UART_REG_IO_WIDTH . . . 458
26.104.1.10 DEFAULT_UART_REG_SHIFT . . . 458
26.104.2 Function Documentation . . . 458
26.104.2.1 fdt_find_match() . . . 458
26.104.2.2 fdt_get_node_addr_size() . . . 459
26.104.2.3 fdt_match_node() . . . 460
26.104.2.4 fdt_parse_clint_node() . . . 460
26.104.2.5 fdt_parse_compat_addr() . . . 461
26.104.2.6 fdt_parse_hart_id() . . . 461
26.104.2.7 fdt_parse_max_hart_id() . . . 462
26.104.2.8 fdt_parse_plic() . . . 462
26.104.2.9 fdt_parse_plic_node() . . . 463
26.104.2.10 fdt_parse_shakti_uart_node() . . . 463
26.104.2.11 fdt_parse_sifive_uart_node() . . . 464
26.104.2.12 fdt_parse_uart8250() . . . 465
26.104.2.13 fdt_parse_uart8250_node() . . . 465
26.105 lib/utils/ipi/fdt_ipi.c File Reference . . . 465
26.105.1 Function Documentation . . . 466
26.105.1.1 dummy_clear() . . . 466
26.105.1.2 dummy_send() . . . 466
26.105.1.3 fdt_ipi_clear() . . . 467
26.105.1.4 fdt_ipi_cold_init() . . . 467
26.105.1.5 fdt_ipi_exit() . . . 467
26.105.1.6 fdt_ipi_init() . . . 467
26.105.1.7 fdt_ipi_send() . . . 467
26.105.1.8 fdt_ipi_warm_init() . . . 468
26.105.2 Variable Documentation . . . 468
26.105.2.1 current_driver . . . 468
26.105.2.2 dummy . . . 468
26.105.2.3 fdt_ipi_clint . . . 468
26.105.2.4 ipi_drivers . . . 468
26.106 lib/utils/ipi/fdt_ipi_clint.c File Reference . . . 468

26.106.1 Macro Definition Documentation . . . 469
26.106.1.1 CLINT_IPI_MAX_NR . . . 469
26.106.2 Function Documentation . . . 469
26.106.2.1 ipi_clint_cold_init() . . . 469
26.106.3 Variable Documentation . . . 470
26.106.3.1 clint_ipi . . . 470
26.106.3.2 clint_ipi_count . . . 470
26.106.3.3 fdt_ipi_clint . . . 470
26.106.3.4 ipi_clint_match . . . 470
26.107 lib/utils/irqchip/fdt_irqchip.c File Reference . . . 470
26.107.1 Function Documentation . . . 471
26.107.1.1 fdt_irqchip_cold_init() . . . 471
26.107.1.2 fdt_irqchip_exit() . . . 472
26.107.1.3 fdt_irqchip_init() . . . 472
26.107.1.4 fdt_irqchip_warm_init() . . . 472
26.107.2 Variable Documentation . . . 472
26.107.2.1 current_driver . . . 472
26.107.2.2 fdt_irqchip_plic . . . 473
26.107.2.3 irqchip_drivers . . . 473
26.108 lib/utils/irqchip/fdt_irqchip_plic.c File Reference . . . 473
26.108.1 Macro Definition Documentation . . . 474
26.108.1.1 PLIC_MAX_NR . . . 474
26.108.2 Function Documentation . . . 474
26.108.2.1 irqchip_plic_cold_init() . . . 474
26.108.2.2 irqchip_plic_update_hartid_table() . . . 474
26.108.2.3 irqchip_plic_warm_init() . . . 475
26.108.3 Variable Documentation . . . 475
26.108.3.1 fdt_irqchip_plic . . . 475
26.108.3.2 irqchip_plic_match . . . 475
26.108.3.3 plic . . . 475
26.108.3.4 plic_count . . . 476
26.108.3.5 plic_hartid2context . . . 476
26.108.3.6 plic_hartid2data . . . 476
26.109 lib/utils/irqchip/plic.c File Reference . . . 476
26.109.1 Macro Definition Documentation . . . 476
26.109.1.1 PLIC_CONTEXT_BASE . . . 476
26.109.1.2 PLIC_CONTEXT_STRIDE . . . 477
26.109.1.3 PLIC_ENABLE_BASE . . . 477
26.109.1.4 PLIC_ENABLE_STRIDE . . . 477
26.109.1.5 PLIC_PENDING_BASE . . . 477
26.109.1.6 PLIC_PRIORITY_BASE . . . 477
26.109.2 Function Documentation . . . 477

26.109.2.1 plic_cold_irqchip_init() . . . 477
26.109.2.2 plic_set_ie() . . . 478
26.109.2.3 plic_set_priority() . . . 478
26.109.2.4 plic_set_thresh() . . . 478
26.109.2.5 plic_warm_irqchip_init() . . . 478
26.110 lib/utils/libfdt/fdt.c File Reference . . . 479
26.110.1 Function Documentation . . . 480
26.110.1.1 check_block_() . . . 480
26.110.1.2 check_off_() . . . 481
26.110.1.3 fdt_check_header() . . . 481
26.110.1.4 fdt_check_node_offset_() . . . 482
26.110.1.5 fdt_check_prop_offset_() . . . 483
26.110.1.6 fdt_find_string_() . . . 484
26.110.1.7 fdt_first_subnode() . . . 484
26.110.1.8 fdt_header_size_() . . . 485
26.110.1.9 fdt_move() . . . 485
26.110.1.10 fdt_next_node() . . . 486
26.110.1.11 fdt_next_subnode() . . . 486
26.110.1.12 fdt_next_tag() . . . 487
26.110.1.13 fdt_offset_ptr() . . . 488
26.110.1.14 fdt_ro_probe_() . . . 488
26.111 lib/utils/libfdt/fdt.h File Reference . . . 489
26.111.1 Macro Definition Documentation . . . 490
26.111.1.1 FDT_BEGIN_NODE . . . 490
26.111.1.2 FDT_END . . . 490
26.111.1.3 FDT_END_NODE . . . 490
26.111.1.4 FDT_MAGIC . . . 490
26.111.1.5 FDT_NOP . . . 490
26.111.1.6 FDT_PROP . . . 490
26.111.1.7 FDT_TAGSIZE . . . 490
26.111.1.8 FDT_V16_SIZE . . . 490
26.111.1.9 FDT_V17_SIZE . . . 490
26.111.1.10 FDT_V1_SIZE . . . 490
26.111.1.11 FDT_V2_SIZE . . . 490
26.111.1.12 FDT_V3_SIZE . . . 491
26.112 lib/utils/libfdt/fdt_addresses.c File Reference . . . 491
26.112.1 Function Documentation . . . 491
26.112.1.1 fdt_address_cells() . . . 491
26.112.1.2 fdt_appendprop_addrrange() . . . 492
26.112.1.3 fdt_cells() . . . 493
26.112.1.4 fdt_size_cells() . . . 493
26.113 lib/utils/libfdt/fdt_empty_tree.c File Reference . . . 494

26.113.1 Function Documentation . . . 495
26.113.1.1 fdt_create_empty_tree() . . . 495
26.114 lib/utils/libfdt/fdt_overlay.c File Reference . . . 496
26.114.1 Function Documentation . . . 497
26.114.1.1 fdt_overlay_apply() . . . 497
26.114.1.2 get_path_len() . . . 498
26.114.1.3 overlay_adjust_local_phandles() . . . 498
26.114.1.4 overlay_adjust_node_phandles() . . . 499
26.114.1.5 overlay_apply_node() . . . 500
26.114.1.6 overlay_fixup_one_phandle() . . . 500
26.114.1.7 overlay_fixup_phandle() . . . 501
26.114.1.8 overlay_fixup_phandles() . . . 501
26.114.1.9 overlay_get_target() . . . 502
26.114.1.10 overlay_get_target_phandle() . . . 503
26.114.1.11 overlay_merge() . . . 503
26.114.1.12 overlay_phandle_add_offset() . . . 504
26.114.1.13 overlay_symbol_update() . . . 505
26.114.1.14 overlay_update_local_node_references() . . . 505
26.114.1.15 overlay_update_local_references() . . . 506
26.115 lib/utils/libfdt/fdt_ro.c File Reference . . . 507
26.115.1 Function Documentation . . . 508
26.115.1.1 fdt_check_full() . . . 508
26.115.1.2 fdt_find_max_phandle() . . . 509
26.115.1.3 fdt_first_property_offset() . . . 510
26.115.1.4 fdt_generate_phandle() . . . 510
26.115.1.5 fdt_get_alias() . . . 511
26.115.1.6 fdt_get_alias_namelen() . . . 511
26.115.1.7 fdt_get_mem_rsv() . . . 512
26.115.1.8 fdt_get_name() . . . 512
26.115.1.9 fdt_get_path() . . . 513
26.115.1.10 fdt_get_phandle() . . . 514
26.115.1.11 fdt_get_property() . . . 515
26.115.1.12 fdt_get_property_by_offset() . . . 515
26.115.1.13 fdt_get_property_by_offset_() . . . 516
26.115.1.14 fdt_get_property_namelen() . . . 517
26.115.1.15 fdt_get_property_namelen_() . . . 518
26.115.1.16 fdt_get_string() . . . 518
26.115.1.17 fdt_getprop() . . . 519
26.115.1.18 fdt_getprop_by_offset() . . . 520
26.115.1.19 fdt_getprop_namelen() . . . 521
26.115.1.20 fdt_mem_rsv() . . . 522
26.115.1.21 fdt_next_property_offset() . . . 523

26.115.1.22 fdt_node_check_compatible() . . . 524
26.115.1.23 fdt_node_depth() . . . 525
26.115.1.24 fdt_node_offset_by_compatible() . . . 526
26.115.1.25 fdt_node_offset_by_phandle() . . . 526
26.115.1.26 fdt_node_offset_by_prop_value() . . . 527
26.115.1.27 fdt_nodename_eq_() . . . 528
26.115.1.28 fdt_num_mem_rsv() . . . 528
26.115.1.29 fdt_parent_offset() . . . 529
26.115.1.30 fdt_path_offset() . . . 530
26.115.1.31 fdt_path_offset_namelen() . . . 531
26.115.1.32 fdt_string() . . . 531
26.115.1.33 fdt_string_eq_() . . . 532
26.115.1.34 fdt_stringlist_contains() . . . 533
26.115.1.35 fdt_stringlist_count() . . . 533
26.115.1.36 fdt_stringlist_get() . . . 534
26.115.1.37 fdt_stringlist_search() . . . 534
26.115.1.38 fdt_subnode_offset() . . . 534
26.115.1.39 fdt_subnode_offset_namelen() . . . 535
26.115.1.40 fdt_supernode_atdepth_offset() . . . 536
26.115.1.41 nextprop_() . . . 537
26.116 lib/utils/libfdt/fdt_rw.c File Reference . . . 538
26.116.1 Macro Definition Documentation . . . 540
26.116.1.1 FDT_RW_PROBE . . . 540
26.116.2 Function Documentation . . . 540
26.116.2.1 fdt_add_mem_rsv() . . . 540
26.116.2.2 fdt_add_property_() . . . 540
26.116.2.3 fdt_add_subnode() . . . 541
26.116.2.4 fdt_add_subnode_namelen() . . . 542
26.116.2.5 fdt_appendprop() . . . 542
26.116.2.6 fdt_blocks_misordered_() . . . 543
26.116.2.7 fdt_data_size_() . . . 543
26.116.2.8 fdt_del_last_string_() . . . 544
26.116.2.9 fdt_del_mem_rsv() . . . 544
26.116.2.10 fdt_del_node() . . . 545
26.116.2.11 fdt_delprop() . . . 545
26.116.2.12 fdt_find_add_string_() . . . 545
26.116.2.13 fdt_open_into() . . . 546
26.116.2.14 fdt_pack() . . . 546
26.116.2.15 fdt_packblocks_() . . . 547
26.116.2.16 fdt_resize_property_() . . . 547
26.116.2.17 fdt_rw_probe_() . . . 548
26.116.2.18 fdt_set_name() . . . 548

26.116.2.19 fdt_setprop() . . . 548
26.116.2.20 fdt_setprop_placeholder() . . . 549
26.116.2.21 fdt_splice_() . . . 550
26.116.2.22 fdt_splice_mem_rsv_() . . . 550
26.116.2.23 fdt_splice_string_() . . . 551
26.116.2.24 fdt_splice_struct_() . . . 551
26.117 lib/utils/libfdt/fdt_strerror.c File Reference . . . 552
26.117.1 Macro Definition Documentation . . . 553
26.117.1.1 FDT_ERRTABENT . . . 553
26.117.1.2 FDT_ERRTABSIZE . . . 553
26.117.2 Function Documentation . . . 554
26.117.2.1 fdt_strerror() . . . 554
26.117.3 Variable Documentation . . . 554
26.117.3.1 fdt_errtable . . . 554
26.118 lib/utils/libfdt/fdt_sw.c File Reference . . . 554
26.118.1 Macro Definition Documentation . . . 556
26.118.1.1 FDT_SW_PROBE . . . 556
26.118.1.2 FDT_SW_PROBE_MEMRSV . . . 556
26.118.1.3 FDT_SW_PROBE_STRUCT . . . 556
26.118.2 Function Documentation . . . 556
26.118.2.1 fdt_add_reservemap_entry() . . . 556
26.118.2.2 fdt_add_string_() . . . 557
26.118.2.3 fdt_begin_node() . . . 557
26.118.2.4 fdt_create() . . . 558
26.118.2.5 fdt_create_with_flags() . . . 558
26.118.2.6 fdt_del_last_string_() . . . 559
26.118.2.7 fdt_end_node() . . . 559
26.118.2.8 fdt_find_add_string_() . . . 559
26.118.2.9 fdt_finish() . . . 560
26.118.2.10 fdt_finish_reservemap() . . . 561
26.118.2.11 fdt_grab_space_() . . . 561
26.118.2.12 fdt_property() . . . 562
26.118.2.13 fdt_property_placeholder() . . . 562
26.118.2.14 fdt_resize() . . . 563
26.118.2.15 fdt_sw_probe_() . . . 563
26.118.2.16 fdt_sw_probe_memrsv_() . . . 564
26.118.2.17 fdt_sw_probe_struct_() . . . 564
26.118.2.18 sw_flags() . . . 564
26.119 lib/utils/libfdt/fdt_wip.c File Reference . . . 564
26.119.1 Function Documentation . . . 565
26.119.1.1 fdt_node_end_offset_() . . . 565
26.119.1.2 fdt_nop_node() . . . 566

26.119.1.3 fdt_nop_property() . . . 566
26.119.1.4 fdt_nop_region_() . . . 567
26.119.1.5 fdt_setprop_inplace() . . . 567
26.119.1.6 fdt_setprop_inplace_namelen_partial() . . . 568
26.120 lib/utils/libfdt/libfdt.h File Reference . . . 568
26.120.1 Macro Definition Documentation . . . 574
26.120.1.1 fdt_appendprop_string . . . 574
26.120.1.2 fdt_boot_cpuid_phys . . . 574
26.120.1.3 FDT_CREATE_FLAG_NO_NAME_DEDUP . . . 574
26.120.1.4 FDT_CREATE_FLAGS_ALL . . . 574
26.120.1.5 FDT_ERR_BADFLAGS . . . 575
26.120.1.6 FDT_ERR_BADLAYOUT . . . 575
26.120.1.7 FDT_ERR_BADMAGIC . . . 575
26.120.1.8 FDT_ERR_BADNCELLS . . . 575
26.120.1.9 FDT_ERR_BADOFFSET . . . 575
26.120.1.10 FDT_ERR_BADOVERLAY . . . 575
26.120.1.11 FDT_ERR_BADPATH . . . 575
26.120.1.12 FDT_ERR_BADPHANDLE . . . 575
26.120.1.13 FDT_ERR_BADSTATE . . . 575
26.120.1.14 FDT_ERR_BADSTRUCTURE . . . 575
26.120.1.15 FDT_ERR_BADVALUE . . . 575
26.120.1.16 FDT_ERR_BADVERSION . . . 575
26.120.1.17 FDT_ERR_EXISTS . . . 576
26.120.1.18 FDT_ERR_INTERNAL . . . 576
26.120.1.19 FDT_ERR_MAX . . . 576
26.120.1.20 FDT_ERR_NOPHANDLES . . . 576
26.120.1.21 FDT_ERR_NOSPACE . . . 576
26.120.1.22 FDT_ERR_NOTFOUND . . . 576
26.120.1.23 FDT_ERR_TRUNCATED . . . 576
26.120.1.24 FDT_FIRST_SUPPORTED_VERSION . . . 576
26.120.1.25 fdt_for_each_property_offset . . . 576
26.120.1.26 fdt_for_each_subnode . . . 577
26.120.1.27 fdt_get_header . . . 577
26.120.1.28 fdt_last_comp_version . . . 577
26.120.1.29 FDT_LAST_SUPPORTED_VERSION . . . 577
26.120.1.30 fdt_magic . . . 577
26.120.1.31 FDT_MAX_NCELLS . . . 577
26.120.1.32 FDT_MAX_PHANDLE . . . 577
26.120.1.33 fdt_off_dt_strings . . . 577
26.120.1.34 fdt_off_dt_struct . . . 578
26.120.1.35 fdt_off_mem_rsvmap . . . 578
26.120.1.36 fdt_property_string . . . 578

26.120.1.37 fdt_set_hdr_ . . . 578
26.120.1.38 fdt_setprop_empty . . . 578
26.120.1.39 fdt_setprop_string . . . 578
26.120.1.40 fdt_size_dt_strings . . . 578
26.120.1.41 fdt_size_dt_struct . . . 578
26.120.1.42 fdt_totalsize . . . 579
26.120.1.43 fdt_version . . . 579
26.120.2 Function Documentation . . . 579
26.120.2.1 fdt32_ld() . . . 579
26.120.2.2 fdt32_st() . . . 579
26.120.2.3 fdt64_ld() . . . 580
26.120.2.4 fdt64_st() . . . 580
26.120.2.5 fdt_add_mem_rsv() . . . 580
26.120.2.6 fdt_add_reservemap_entry() . . . 581
26.120.2.7 fdt_add_subnode() . . . 582
26.120.2.8 fdt_add_subnode_namelen() . . . 582
26.120.2.9 fdt_address_cells() . . . 583
26.120.2.10 fdt_appendprop() . . . 583
26.120.2.11 fdt_appendprop_addrrange() . . . 584
26.120.2.12 fdt_appendprop_cell() . . . 585
26.120.2.13 fdt_appendprop_u32() . . . 585
26.120.2.14 fdt_appendprop_u64() . . . 585
26.120.2.15 fdt_begin_node() . . . 586
26.120.2.16 fdt_check_full() . . . 586
26.120.2.17 fdt_check_header() . . . 587
26.120.2.18 fdt_create() . . . 588
26.120.2.19 fdt_create_empty_tree() . . . 588
26.120.2.20 fdt_create_with_flags() . . . 589
26.120.2.21 fdt_del_mem_rsv() . . . 589
26.120.2.22 fdt_del_node() . . . 590
26.120.2.23 fdt_delprop() . . . 590
26.120.2.24 fdt_end_node() . . . 591
26.120.2.25 fdt_find_max_phandle() . . . 591
26.120.2.26 fdt_finish() . . . 592
26.120.2.27 fdt_finish_reservemap() . . . 592
26.120.2.28 fdt_first_property_offset() . . . 593
26.120.2.29 fdt_first_subnode() . . . 594
26.120.2.30 fdt_generate_phandle() . . . 594
26.120.2.31 fdt_get_alias() . . . 595
26.120.2.32 fdt_get_alias_namelen() . . . 595
26.120.2.33 fdt_get_max_phandle() . . . 596
26.120.2.34 fdt_get_mem_rsv() . . . 596

26.120.2.35 fdt_get_name() . . . 597
26.120.2.36 fdt_get_path() . . . 598
26.120.2.37 fdt_get_phandle() . . . 599
26.120.2.38 fdt_get_property() . . . 599
26.120.2.39 fdt_get_property_by_offset() . . . 600
26.120.2.40 fdt_get_property_namelen() . . . 600
26.120.2.41 fdt_get_property_w() . . . 601
26.120.2.42 fdt_get_string() . . . 601
26.120.2.43 fdt_getprop() . . . 602
26.120.2.44 fdt_getprop_by_offset() . . . 603
26.120.2.45 fdt_getprop_namelen() . . . 604
26.120.2.46 fdt_getprop_namelen_w() . . . 605
26.120.2.47 fdt_getprop_w() . . . 606
26.120.2.48 fdt_header_size() . . . 606
26.120.2.49 fdt_header_size_() . . . 606
26.120.2.50 fdt_move() . . . 607
26.120.2.51 fdt_next_node() . . . 607
26.120.2.52 fdt_next_property_offset() . . . 608
26.120.2.53 fdt_next_subnode() . . . 609
26.120.2.54 fdt_next_tag() . . . 609
26.120.2.55 fdt_node_check_compatible() . . . 610
26.120.2.56 fdt_node_depth() . . . 611
26.120.2.57 fdt_node_offset_by_compatible() . . . 612
26.120.2.58 fdt_node_offset_by_phandle() . . . 613
26.120.2.59 fdt_node_offset_by_prop_value() . . . 614
26.120.2.60 fdt_nop_node() . . . 614
26.120.2.61 fdt_nop_property() . . . 615
26.120.2.62 fdt_num_mem_rsv() . . . 615
26.120.2.63 fdt_offset_ptr() . . . 616
26.120.2.64 fdt_offset_ptr_w() . . . 616
26.120.2.65 fdt_open_into() . . . 617
26.120.2.66 fdt_overlay_apply() . . . 617
26.120.2.67 fdt_pack() . . . 618
26.120.2.68 fdt_parent_offset() . . . 618
26.120.2.69 fdt_path_offset() . . . 619
26.120.2.70 fdt_path_offset_namelen() . . . 620
26.120.2.71 fdt_property() . . . 621
26.120.2.72 fdt_property_cell() . . . 622
26.120.2.73 fdt_property_placeholder() . . . 622
26.120.2.74 fdt_property_u32() . . . 623
26.120.2.75 fdt_property_u64() . . . 623
26.120.2.76 fdt_resize() . . . 623

26.120.2.77 fdt_set_hdr_() [1/10] . . . 624
26.120.2.78 fdt_set_hdr_() [2/10] . . . 624
26.120.2.79 fdt_set_hdr_() [3/10] . . . 624
26.120.2.80 fdt_set_hdr_() [4/10] . . . 624
26.120.2.81 fdt_set_hdr_() [5/10] . . . 624
26.120.2.82 fdt_set_hdr_() [6/10] . . . 624
26.120.2.83 fdt_set_hdr_() [7/10] . . . 624
26.120.2.84 fdt_set_hdr_() [8/10] . . . 624
26.120.2.85 fdt_set_hdr_() [9/10] . . . 624
26.120.2.86 fdt_set_hdr_() [10/10] . . . 625
26.120.2.87 fdt_set_name() . . . 625
26.120.2.88 fdt_setprop() . . . 625
26.120.2.89 fdt_setprop_cell() . . . 626
26.120.2.90 fdt_setprop_inplace() . . . 626
26.120.2.91 fdt_setprop_inplace_cell() . . . 626
26.120.2.92 fdt_setprop_inplace_namelen_partial() . . . 627
26.120.2.93 fdt_setprop_inplace_u32() . . . 627
26.120.2.94 fdt_setprop_inplace_u64() . . . 628
26.120.2.95 fdt_setprop_placeholder() . . . 628
26.120.2.96 fdt_setprop_u32() . . . 629
26.120.2.97 fdt_setprop_u64() . . . 629
26.120.2.98 fdt_size_cells() . . . 629
26.120.2.99 fdt_strerror() . . . 630
26.120.2.100 fdt_string() . . . 630
26.120.2.101 fdt_stringlist_contains() . . . 631
26.120.2.102 fdt_stringlist_count() . . . 631
26.120.2.103 fdt_stringlist_get() . . . 632
26.120.2.104 fdt_stringlist_search() . . . 632
26.120.2.105 fdt_subnode_offset() . . . 632
26.120.2.106 fdt_subnode_offset_namelen() . . . 633
26.120.2.107 fdt_supernode_atdepth_offset() . . . 634
26.121 lib/utils/libfdt/libfdt_env.h File Reference . . . 634
26.121.1 Macro Definition Documentation . . . 636
26.121.1.1 CPU_TO_FDT16 . . . 636
26.121.1.2 CPU_TO_FDT32 . . . 636
26.121.1.3 CPU_TO_FDT64 . . . 636
26.121.1.4 EXTRACT_BYTE . . . 636
26.121.1.5 FDT_BITWISE . . . 636
26.121.1.6 FDT_FORCE . . . 636
26.121.1.7 INT32_MAX . . . 637
26.121.1.8 INT_MAX . . . 637
26.121.1.9 memchr . . . 637

26.121.1.10 memcmp . . . 637
26.121.1.11 memcpy . . . 637
26.121.1.12 memmove . . . 637
26.121.1.13 memset . . . 637
26.121.1.14 strchr . . . 637
26.121.1.15 strcmp . . . 637
26.121.1.16 strcpy . . . 637
26.121.1.17 strlen . . . 637
26.121.1.18 strnlen . . . 637
26.121.1.19 strrchr . . . 638
26.121.1.20 UINT32_MAX . . . 638
26.121.1.21 UINT_MAX . . . 638
26.121.2 Typedef Documentation . . . 638
26.121.2.1 fdt16_t . . . 638
26.121.2.2 fdt32_t . . . 638
26.121.2.3 fdt64_t . . . 638
26.121.3 Function Documentation . . . 638
26.121.3.1 cpu_to_fdt16() . . . 638
26.121.3.2 cpu_to_fdt32() . . . 638
26.121.3.3 cpu_to_fdt64() . . . 639
26.121.3.4 fdt16_to_cpu() . . . 639
26.121.3.5 fdt32_to_cpu() . . . 639
26.121.3.6 fdt64_to_cpu() . . . 640
26.122 lib/utils/libfdt/libfdt_internal.h File Reference . . . 640
26.122.1 Macro Definition Documentation . . . 641
26.122.1.1 FDT_ALIGN . . . 641
26.122.1.2 FDT_RO_PROBE . . . 642
26.122.1.3 FDT_SW_MAGIC . . . 642
26.122.1.4 FDT_TAGALIGN . . . 642
26.122.2 Function Documentation . . . 642
26.122.2.1 fdt_check_node_offset_() . . . 642
26.122.2.2 fdt_check_prop_offset_() . . . 643
26.122.2.3 fdt_find_string_() . . . 644
26.122.2.4 fdt_mem_rsv_() . . . 644
26.122.2.5 fdt_mem_rsv_w_() . . . 645
26.122.2.6 fdt_node_end_offset_() . . . 645
26.122.2.7 fdt_offset_ptr_() . . . 646
26.122.2.8 fdt_offset_ptr_w_() . . . 646
26.122.2.9 fdt_ro_probe_() . . . 647
26.123 lib/utils/reset/fdt_reset.c File Reference . . . 648
26.123.1 Function Documentation . . . 648
26.123.1.1 fdt_reset_init() . . . 648

26.123.1.2 fdt_system_reset() . . . 649
26.123.2 Variable Documentation . . . 649
26.123.2.1 current_driver . . . 649
26.123.2.2 fdt_reset_htif . . . 649
26.123.2.3 fdt_reset_sifive . . . 649
26.123.2.4 reset_drivers . . . 649
26.124 lib/utils/reset/fdt_reset_htif.c File Reference . . . 649
26.124.1 Variable Documentation . . . 650
26.124.1.1 fdt_reset_htif . . . 650
26.124.1.2 htif_reset_match . . . 650
26.125 lib/utils/reset/fdt_reset_sifive.c File Reference . . . 650
26.125.1 Function Documentation . . . 651
26.125.1.1 sifive_test_reset_init() . . . 651
26.125.2 Variable Documentation . . . 651
26.125.2.1 fdt_reset_sifive . . . 651
26.125.2.2 sifive_test_reset_match . . . 652
26.126 lib/utils/serial/fdt_serial.c File Reference . . . 652
26.126.1 Function Documentation . . . 653
26.126.1.1 dummy_getc() . . . 653
26.126.1.2 dummy_putc() . . . 653
26.126.1.3 fdt_serial_getc() . . . 653
26.126.1.4 fdt_serial_init() . . . 653
26.126.1.5 fdt_serial_putc() . . . 653
26.126.2 Variable Documentation . . . 653
26.126.2.1 current_driver . . . 653
26.126.2.2 dummy . . . 654
26.126.2.3 fdt_serial_htif . . . 654
26.126.2.4 fdt_serial_shakti . . . 654
26.126.2.5 fdt_serial_sifive . . . 654
26.126.2.6 fdt_serial_uart8250 . . . 654
26.126.2.7 serial_drivers . . . 654
26.127 lib/utils/serial/fdt_serial_htif.c File Reference . . . 654
26.127.1 Variable Documentation . . . 655
26.127.1.1 fdt_serial_htif . . . 655
26.127.1.2 serial_htif_match . . . 655
26.128 lib/utils/serial/fdt_serial_shakti.c File Reference . . . 655
26.128.1 Function Documentation . . . 656
26.128.1.1 serial_shakti_init() . . . 656
26.128.2 Variable Documentation . . . 656
26.128.2.1 fdt_serial_shakti . . . 656
26.128.2.2 serial_shakti_match . . . 657
26.129 lib/utils/serial/fdt_serial_sifive.c File Reference . . . 657

26.129.1 Function Documentation . . . 657
26.129.1.1 serial_sifive_init() . . . 657
26.129.2 Variable Documentation . . . 658
26.129.2.1 fdt_serial_sifive . . . 658
26.129.2.2 serial_sifive_match . . . 658
26.130 lib/utils/serial/fdt_serial_uart8250.c File Reference . . . 658
26.130.1 Function Documentation . . . 659
26.130.1.1 serial_uart8250_init() . . . 659
26.130.2 Variable Documentation . . . 659
26.130.2.1 fdt_serial_uart8250 . . . 659
26.130.2.2 serial_uart8250_match . . . 659
26.131 lib/utils/serial/shakti-uart.c File Reference . . . 659
26.131.1 Macro Definition Documentation . . . 660
26.131.1.1 REG_BAUD . . . 660
26.131.1.2 REG_CONTROL . . . 660
26.131.1.3 REG_DELAY . . . 661
26.131.1.4 REG_INT_EN . . . 661
26.131.1.5 REG_IQ_CYCLES . . . 661
26.131.1.6 REG_RX . . . 661
26.131.1.7 REG_RX_THRES . . . 661
26.131.1.8 REG_STATUS . . . 661
26.131.1.9 REG_TX . . . 661
26.131.2 Function Documentation . . . 661
26.131.2.1 shakti_uart_getc() . . . 661
26.131.2.2 shakti_uart_init() . . . 661
26.131.2.3 shakti_uart_putc() . . . 662
26.131.3 Variable Documentation . . . 662
26.131.3.1 uart_base . . . 662
26.132 lib/utils/serial/sifive-uart.c File Reference . . . 662
26.132.1 Macro Definition Documentation . . . 663
26.132.1.1 UART_REG_DIV . . . 663
26.132.1.2 UART_REG_IE . . . 663
26.132.1.3 UART_REG_IP . . . 663
26.132.1.4 UART_REG_RXCTRL . . . 663
26.132.1.5 UART_REG_RXFIFO . . . 663
26.132.1.6 UART_REG_TXCTRL . . . 663
26.132.1.7 UART_REG_TXFIFO . . . 664
26.132.1.8 UART_RXCTRL_RXEN . . . 664
26.132.1.9 UART_RXFIFO_DATA . . . 664
26.132.1.10 UART_RXFIFO_EMPTY . . . 664
26.132.1.11 UART_TXCTRL_TXEN . . . 664
26.132.1.12 UART_TXFIFO_FULL . . . 664

26.132.2 Function Documentation . . . 664
26.132.2.1 get_reg() . . . 664
26.132.2.2 set_reg() . . . 664
26.132.2.3 sifive_uart_getc() . . . 665
26.132.2.4 sifive_uart_init() . . . 665
26.132.2.5 sifive_uart_putc() . . . 666
26.132.2.6 uart_min_clk_divisor() . . . 666
26.132.3 Variable Documentation . . . 666
26.132.3.1 uart_base . . . 667
26.132.3.2 uart_baudrate . . . 667
26.132.3.3 uart_in_freq . . . 667
26.133 lib/utils/serial/uart8250.c File Reference . . . 667
26.133.1 Macro Definition Documentation . . . 668
26.133.1.1 UART_DLL_OFFSET . . . 668
26.133.1.2 UART_DLM_OFFSET . . . 668
26.133.1.3 UART_FCR_OFFSET . . . 668
26.133.1.4 UART_IER_OFFSET . . . 668
26.133.1.5 UART_IIR_OFFSET . . . 668
26.133.1.6 UART_LCR_OFFSET . . . 668
26.133.1.7 UART_LSR_BI . . . 669
26.133.1.8 UART_LSR_BRK_ERROR_BITS . . . 669
26.133.1.9 UART_LSR_DR . . . 669
26.133.1.10 UART_LSR_FE . . . 669
26.133.1.11 UART_LSR_FIFOE . . . 669
26.133.1.12 UART_LSR_OE . . . 669
26.133.1.13 UART_LSR_OFFSET . . . 669
26.133.1.14 UART_LSR_PE . . . 669
26.133.1.15 UART_LSR_TEMT . . . 669
26.133.1.16 UART_LSR_THRE . . . 669
26.133.1.17 UART_MCR_OFFSET . . . 669
26.133.1.18 UART_MDR1_OFFSET . . . 669
26.133.1.19 UART_MSR_OFFSET . . . 670
26.133.1.20 UART_RBR_OFFSET . . . 670
26.133.1.21 UART_SCR_OFFSET . . . 670
26.133.1.22 UART_THR_OFFSET . . . 670
26.133.2 Function Documentation . . . 670
26.133.2.1 get_reg() . . . 670
26.133.2.2 set_reg() . . . 670
26.133.2.3 uart8250_getc() . . . 671
26.133.2.4 uart8250_init() . . . 671
26.133.2.5 uart8250_putc() . . . 672
26.133.3 Variable Documentation . . . 672

26.133.3.1 uart8250_base . . . 672
26.133.3.2 uart8250_baudrate . . . 672
26.133.3.3 uart8250_in_freq . . . 672
26.133.3.4 uart8250_reg_shift . . . 672
26.133.3.5 uart8250_reg_width . . . 673
26.134 lib/utils/sys/clint.c File Reference . . . 673
26.134.1 Macro Definition Documentation . . . 674
26.134.1.1 CLINT_IPI_OFF . . . 674
26.134.1.2 CLINT_TIME_CMP_OFF . . . 674
26.134.1.3 CLINT_TIME_VAL_OFF . . . 674
26.134.2 Function Documentation . . . 674
26.134.2.1 clint_cold_ipi_init() . . . 674
26.134.2.2 clint_cold_timer_init() . . . 674
26.134.2.3 clint_ipi_clear() . . . 675
26.134.2.4 clint_ipi_send() . . . 675
26.134.2.5 clint_time_rd32() . . . 676
26.134.2.6 clint_time_rd64() . . . 676
26.134.2.7 clint_time_wr32() . . . 676
26.134.2.8 clint_time_wr64() . . . 676
26.134.2.9 clint_timer_event_start() . . . 677
26.134.2.10 clint_timer_event_stop() . . . 677
26.134.2.11 clint_timer_value() . . . 677
26.134.2.12 clint_warm_ipi_init() . . . 677
26.134.2.13 clint_warm_timer_init() . . . 677
26.134.3 Variable Documentation . . . 678
26.134.3.1 clint_ipi_hartid2data . . . 678
26.134.3.2 clint_timer_hartid2data . . . 678
26.135 lib/utils/sys/htif.c File Reference . . . 678
26.135.1 Macro Definition Documentation . . . 679
26.135.1.1 FROMHOST_CMD . . . 679
26.135.1.2 FROMHOST_DATA . . . 679
26.135.1.3 FROMHOST_DEV . . . 679
26.135.1.4 HTIF_CMD_BITS . . . 679
26.135.1.5 HTIF_CMD_MASK . . . 680
26.135.1.6 HTIF_CMD_SHIFT . . . 680
26.135.1.7 HTIF_CONSOLE_CMD_GETC . . . 680
26.135.1.8 HTIF_CONSOLE_CMD_PUTC . . . 680
26.135.1.9 HTIF_DATA_BITS . . . 680
26.135.1.10 HTIF_DATA_MASK . . . 680
26.135.1.11 HTIF_DATA_SHIFT . . . 680
26.135.1.12 HTIF_DEV_BITS . . . 680
26.135.1.13 HTIF_DEV_CONSOLE . . . 680

26.135.1.14 HTIF_DEV_MASK . . . 680
26.135.1.15 HTIF_DEV_SHIFT . . . 680
26.135.1.16 HTIF_DEV_SYSTEM . . . 680
26.135.1.17 PK_SYS_write . . . 681
26.135.1.18 TOHOST_CMD . . . 681
26.135.2 Function Documentation . . . 681
26.135.2.1 __attribute__() . . . 681
26.135.2.2 __check_fromhost() . . . 681
26.135.2.3 __set_tohost() . . . 681
26.135.2.4 htif_getc() . . . 682
26.135.2.5 htif_putc() . . . 682
26.135.2.6 htif_system_reset() . . . 683
26.135.3 Variable Documentation . . . 683
26.135.3.1 htif_console_buf . . . 683
26.135.3.2 htif_lock . . . 683
26.136 lib/utils/sys/sifive_test.c File Reference . . . 683
26.136.1 Macro Definition Documentation . . . 684
26.136.1.1 FINISHER_FAIL . . . 684
26.136.1.2 FINISHER_PASS . . . 684
26.136.1.3 FINISHER_RESET . . . 684
26.136.2 Function Documentation . . . 684
26.136.2.1 sifive_test_init() . . . 685
26.136.2.2 sifive_test_system_reset() . . . 685
26.136.3 Variable Documentation . . . 685
26.136.3.1 sifive_test_base . . . 685
26.137 lib/utils/timer/fdt_timer.c File Reference . . . 685
26.137.1 Function Documentation . . . 686
26.137.1.1 dummy_event_start() . . . 686
26.137.1.2 dummy_event_stop() . . . 687
26.137.1.3 dummy_value() . . . 687
26.137.1.4 fdt_timer_cold_init() . . . 687
26.137.1.5 fdt_timer_event_start() . . . 687
26.137.1.6 fdt_timer_event_stop() . . . 687
26.137.1.7 fdt_timer_exit() . . . 687
26.137.1.8 fdt_timer_init() . . . 688
26.137.1.9 fdt_timer_value() . . . 688
26.137.1.10 fdt_timer_warm_init() . . . 688
26.137.2 Variable Documentation . . . 688
26.137.2.1 current_driver . . . 688
26.137.2.2 dummy . . . 688
26.137.2.3 fdt_timer_clint . . . 689
26.137.2.4 timer_drivers . . . 689

26.138 lib/utils/timer/fdt_timer_clint.c File Reference 689
26.138.1 Macro Definition Documentation 689
26.138.1.1 CLINT_TIMER_MAX_NR 689
26.138.2 Function Documentation 690
26.138.2.1 timer_clint_cold_init() 690
26.138.3 Variable Documentation 690
26.138.3.1 clint_timer 690
26.138.3.2 clint_timer_count 690
26.138.3.3 fdt_timer_clint 690
26.138.3.4 timer_clint_match 690
26.139 README.md File Reference 691

Index **693**

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 and other contributors.

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

OpenSBI can be compiled natively or cross-compiled on a x86 host. For cross-compilation, you can build your own toolchain or just download a prebuilt one from the Bootlin toolchain repository.

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

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-unknown-elf-* if the gcc executable used is *riscv64-unknown-elf-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.

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-unknown-elf-*, 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-unknown-elf-*.

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-unknown-elf-
export PLATFORM_RISCV_XLEN=32
```

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

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.

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 can be sent in the following ways:

1. Email patches to the OpenSBI mailing list at `opensbi@lists.infradead.org`
2. GitHub Pull Requests (PRs) to the `OpenSBI main repository`

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

The OpenSBI maintainers prefer patches via the OpenSBI mailing list (option 1 above) so that they are visible to a wider audience. All accepted patches on the OpenSBI mailing list will be taken by any of the OpenSBI maintainers and merged into the `OpenSBI main repository` using GitHub PRs.

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.
8. A pull request must get at least one review from a maintainer.
9. A pull request must spend at least 24 hours in review to allow for other developers to review.

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 *riskv-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 a platform configuration file named *config.mk* under the *platform/<xyz>/* directory. This configuration file will provide compiler flags, and select firmware options.
3. Create a *platform/<xyz>/objects.mk* file for listing the platform-specific object files to be compiled.
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 or rv64ima 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 or rv64ima are optional for OpenSBI. Although, OpenSBI 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. *libsbiutils.a* - A static library that will contain all common code required by any platform supported in OpenSBI. It will be built by default and included in libplatsbi.a. This library is installed as ∗<install_directory>/lib/libsbiutils.a∗.

3. *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_TYP↩ ES∗∗∗=OpensbiTypes.h* Above tells *sbi_types.h* to refer to *OpensbiTypes.h* instead of using original definitions of data types.

Chapter 6

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.
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 .text section of the final firmware.

Firmware Configuration and Compilation

All firmware types mandate the definition of the following compile time configuration parameter.

- **FW_TEXT_ADDR** - Defines the address at which the previous booting stage loads OpenSBI firmware.

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 7

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 it's address to *FW_DYNAMIC* via *a2* register of RISC-V CPU.
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 *config.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 requires any platform specific configuration parameters because all required information is passed by previous booting stage at runtime via *struct fw_dynamic_info*.

Chapter 8

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 *config.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 *config.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. This is a mandatory parameter. Compilation errors will result from not defining this address.
- **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.

∗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 9

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 .text 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 *config.mk* configuration file.

The compiled *FW_PAYLOAD* firmware ELF file is named *fw_jump.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 *config.mk* configuration file. The parameters currently defined are as follows:

- **FW_PAYLOAD_OFFSET** - Offset from *FW_TEXT_BASE* 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_PAYL↩OAD_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_PATH** - Path to an external flattened device tree binary file to be embedded in the ∗.text∗ 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_PAYLOAD_FDT_ADDR** - Address where the FDT passed by the prior booting stage or specified by the *FW_PAYLOAD_FDT_PATH* parameter and embedded in the ∗.text∗ 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∗ 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 10

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 ../platform/qemu_virt.md "QEMU" and the ../platform/sifive_fu540.md "HiFive Unleashed" platform guides.

Chapter 11

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 12

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 fundamentalAH↩B/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=andes/ae350* parameter to the top level make command.

Platform Options

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

Building Andes AE350 Platform

To use Linux v5.2 should be used to build Andes AE350 OpenSBI binaries by using the compile time option FW_↩PAYLOAD_FDT_PATH.
AE350's dts is included in https://github.com/andestech/linux/tree/ast-v3_2_↩0-release-public

Linux Kernel Payload

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

Chapter 13

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 F↩ PGA 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 conncet 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 14

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 15

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 FW_TEXT_START is 0x80000000
2. platform features are default
3. platform stack size is default
4. 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.
For custom FW_TEXT_START, we can build the platform-specific library and firmware images by passing *PLAT↩FORM=generic FW_TEXT_START=<custom_text_start>* 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

- **QEMU RISC-V Virt Machine** ([*qemu_virt.md*])
- **Spike** ([*spike.md*])
- **Shakti C-class SoC Platform** ([*shakti_cclass.md*])

Chapter 16

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 17

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]*.
- **T-HEAD C910**: Platform support for the T-HEAD C910 Processor. More details on this platform can be found in the file *[thead-c910.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]*.

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 *object.mk*, *config.mk* and *platform.c* template files provides enough comments to facilitate the implementation.

Chapter 18

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 \
    -kernel 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 \
    -kernel 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*.
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```
Run:
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -kernel 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 \
    -kernel 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 \
    -kernel 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 \
    -kernel 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"
```

Chapter 19

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_PAYLOAD_FDT_PATH=<shakti.dtb path>
```

Test Payload

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

Chapter 20

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=sifive/fu540* 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_PAYLOAD_FDT_PATH.*

```
make PLATFORM=sifive/fu540 FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
or
(For Linux v5.2 or higher)
make PLATFORM=sifive/fu540 FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
       FW_PAYLOAD_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 (or higher).
The detailed U-Boot booting guide is avaialble at `U-Boot`.

```
make PLATFORM=sifive/fu540 FW_PAYLOAD_PATH=<u-boot_build_dir>/u-boot-dtb.bin
```

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/sifive/fu540/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)

1. 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_PAYLOAD_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 v2020.01 (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_fu540_↩ 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=sifive/fu540 FW_PAYLOAD_PATH=<u-boot_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 21

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 --initrd <path_to_cpio_ramdisk> build/platform/generic/firmware/fw_payload.elf
```

Execution on QEMU RISC-V 64-bit

No Payload Case
Build:
```
make PLATFORM=generic
```
Run:
```
qemu-system-riscv64 -M spike -m 256M -nographic \
    -kernel 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 \
    -kernel 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 22

T-HEAD C910 Processor

C910 is a 12-stage, 3 issues, 8 executions, out-of-order 64-bit RISC-V CPU which supports 16 cores, runs with 2.5GHz, and is capable of running Linux.
To build platform specific library and firmwares, provide the *PLATFORM=thead/c910* parameter to the top level make command.

Platform Options

The *T-HEAD C910* platform does not have any platform-specific options.

Building T-HEAD C910 Platform

```
make PLATFORM=thead/c910
```

Booting T-HEAD C910 Platform

No Payload
As there's no payload, you may download vmlinux or u-boot to FW_JUMP_ADDR which specified in config.mk or compile commands with GDB. And the execution flow will turn to vmlinux or u-boot when opensbi ends.
Linux Kernel Payload
You can also choose to use Linux kernel as payload by enabling FW_PAYLOAD=y along with specifying FW_P↩AYLOAD_OFFSET. The kernel image will be embedded in the OPENSBI firmware binary, T-head will directly boot into Linux after OpenSBI.

Chapter 23

Data Structure Index

23.1 Data Structures

Here are the data structures with brief descriptions:

atomic_t 53
clint_data 53
fdt_header 55
fdt_ipi 56
fdt_irqchip 57
fdt_match 57
fdt_node_header 58
fdt_property 58
fdt_reserve_entry 59
fdt_reset 59
fdt_serial 60
fdt_timer 60
fw_dynamic_info 61
platform_uart_data 62
plic_data 63
sbi_dlist 64
sbi_ecall_extension 64
sbi_fifo 65
sbi_hartmask 66
sbi_ipi_event_ops 66
sbi_platform 67
sbi_platform_operations 68
sbi_scratch 72
sbi_tlb_info 73
sbi_trap_info 74
sbi_trap_regs 74
spinlock_t 78

Chapter 24

File Index

24.1 File List

Here is a list of all files with brief descriptions:

include/sbi/fw_dynamic.h . . . 82
include/sbi/riscv_asm.h . . . 85
include/sbi/riscv_atomic.h . . . 92
include/sbi/riscv_barrier.h . . . 96
include/sbi/riscv_encoding.h . . . 98
include/sbi/riscv_fp.h . . . 148
include/sbi/riscv_io.h . . . 150
include/sbi/riscv_locks.h . . . 155
include/sbi/sbi_bitmap.h . . . 158
include/sbi/sbi_bitops.h . . . 164
include/sbi/sbi_console.h . . . 171
include/sbi/sbi_const.h . . . 175
include/sbi/sbi_csr_detect.h . . . 177
include/sbi/sbi_ecall.h . . . 178
include/sbi/sbi_ecall_interface.h . . . 184
include/sbi/sbi_emulate_csr.h . . . 189
include/sbi/sbi_error.h . . . 191
include/sbi/sbi_fifo.h . . . 193
include/sbi/sbi_hart.h . . . 199
include/sbi/sbi_hartmask.h . . . 207
include/sbi/sbi_hfence.h . . . 214
include/sbi/sbi_hsm.h . . . 217
include/sbi/sbi_illegal_insn.h . . . 223
include/sbi/sbi_init.h . . . 225
include/sbi/sbi_ipi.h . . . 229
include/sbi/sbi_list.h . . . 234
include/sbi/sbi_math.h . . . 241
include/sbi/sbi_misaligned_ldst.h . . . 241
include/sbi/sbi_platform.h . . . 244
include/sbi/sbi_scratch.h . . . 267
include/sbi/sbi_string.h . . . 272
include/sbi/sbi_system.h . . . 276
include/sbi/sbi_timer.h . . . 277
include/sbi/sbi_tlb.h . . . 282
include/sbi/sbi_trap.h . . . 285
include/sbi/sbi_types.h . . . 293
include/sbi/sbi_unpriv.h . . . 297
include/sbi/sbi_version.h . . . 299
include/sbi_utils/fdt/fdt_fixup.h . . . 300
include/sbi_utils/fdt/fdt_helper.h . . . 304

include/sbi_utils/ipi/fdt_ipi.h . . . 312
include/sbi_utils/irqchip/fdt_irqchip.h . . . 314
include/sbi_utils/irqchip/plic.h . . . 315
include/sbi_utils/reset/fdt_reset.h . . . 318
include/sbi_utils/serial/fdt_serial.h . . . 319
include/sbi_utils/serial/shakti-uart.h . . . 320
include/sbi_utils/serial/sifive-uart.h . . . 322
include/sbi_utils/serial/uart8250.h . . . 324
include/sbi_utils/sys/clint.h . . . 326
include/sbi_utils/sys/htif.h . . . 330
include/sbi_utils/sys/sifive_test.h . . . 332
include/sbi_utils/timer/fdt_timer.h . . . 333
lib/sbi/riscv_asm.c . . . 335
lib/sbi/riscv_atomic.c . . . 339
lib/sbi/riscv_locks.c . . . 345
lib/sbi/sbi_bitmap.c . . . 348
lib/sbi/sbi_bitops.c . . . 350
lib/sbi/sbi_console.c . . . 352
lib/sbi/sbi_ecall.c . . . 361
lib/sbi/sbi_ecall_base.c . . . 366
lib/sbi/sbi_ecall_hsm.c . . . 367
lib/sbi/sbi_ecall_legacy.c . . . 369
lib/sbi/sbi_ecall_replace.c . . . 371
lib/sbi/sbi_ecall_vendor.c . . . 373
lib/sbi/sbi_emulate_csr.c . . . 375
lib/sbi/sbi_fifo.c . . . 376
lib/sbi/sbi_hart.c . . . 383
lib/sbi/sbi_hsm.c . . . 393
lib/sbi/sbi_illegal_insn.c . . . 400
lib/sbi/sbi_init.c . . . 402
lib/sbi/sbi_ipi.c . . . 412
lib/sbi/sbi_math.c . . . 419
lib/sbi/sbi_misaligned_ldst.c . . . 419
lib/sbi/sbi_platform.c . . . 421
lib/sbi/sbi_scratch.c . . . 423
lib/sbi/sbi_string.c . . . 426
lib/sbi/sbi_system.c . . . 430
lib/sbi/sbi_timer.c . . . 431
lib/sbi/sbi_tlb.c . . . 436
lib/sbi/sbi_trap.c . . . 447
lib/sbi/sbi_unpriv.c . . . 450
lib/utils/fdt/fdt_fixup.c . . . 452
lib/utils/fdt/fdt_helper.c . . . 456
lib/utils/ipi/fdt_ipi.c . . . 465
lib/utils/ipi/fdt_ipi_clint.c . . . 468
lib/utils/irqchip/fdt_irqchip.c . . . 470
lib/utils/irqchip/fdt_irqchip_plic.c . . . 473
lib/utils/irqchip/plic.c . . . 476
lib/utils/libfdt/fdt.c . . . 479
lib/utils/libfdt/fdt.h . . . 489
lib/utils/libfdt/fdt_addresses.c . . . 491
lib/utils/libfdt/fdt_empty_tree.c . . . 494
lib/utils/libfdt/fdt_overlay.c . . . 496
lib/utils/libfdt/fdt_ro.c . . . 507
lib/utils/libfdt/fdt_rw.c . . . 538
lib/utils/libfdt/fdt_strerror.c . . . 552
lib/utils/libfdt/fdt_sw.c . . . 554
lib/utils/libfdt/fdt_wip.c . . . 564

lib/utils/libfdt/libfdt.h . . . 568
lib/utils/libfdt/libfdt_env.h . . . 634
lib/utils/libfdt/libfdt_internal.h . . . 640
lib/utils/reset/fdt_reset.c . . . 648
lib/utils/reset/fdt_reset_htif.c . . . 649
lib/utils/reset/fdt_reset_sifive.c . . . 650
lib/utils/serial/fdt_serial.c . . . 652
lib/utils/serial/fdt_serial_htif.c . . . 654
lib/utils/serial/fdt_serial_shakti.c . . . 655
lib/utils/serial/fdt_serial_sifive.c . . . 657
lib/utils/serial/fdt_serial_uart8250.c . . . 658
lib/utils/serial/shakti-uart.c . . . 659
lib/utils/serial/sifive-uart.c . . . 662
lib/utils/serial/uart8250.c . . . 667
lib/utils/sys/clint.c . . . 673
lib/utils/sys/htif.c . . . 678
lib/utils/sys/sifive_test.c . . . 683
lib/utils/timer/fdt_timer.c . . . 685
lib/utils/timer/fdt_timer_clint.c . . . 689

Chapter 25

Data Structure Documentation

25.1 atomic_t Struct Reference

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

Data Fields

- volatile long counter

25.1.1 Field Documentation

25.1.1.1 counter

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

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

- include/sbi/riscv_atomic.h

25.2 clint_data Struct Reference

```
#include <clint.h>
```

Data Fields

- unsigned long addr
- u32 first_hartid
- u32 hart_count
- bool has_64bit_mmio
- u32 ∗ ipi
- struct clint_data ∗ time_delta_reference
- unsigned long time_delta_computed
- u64 time_delta
- u64 ∗ time_val
- u64 ∗ time_cmp
- u64(∗ time_rd)(volatile u64 ∗addr)
- void(∗ time_wr)(u64 value, volatile u64 ∗addr)

25.2.1 Field Documentation

25.2.1.1 addr

```
unsigned long clint_data::addr
```

25.2.1.2 first_hartid

```
u32 clint_data::first_hartid
```

25.2.1.3 hart_count

```
u32 clint_data::hart_count
```

25.2.1.4 has_64bit_mmio

```
bool clint_data::has_64bit_mmio
```

25.2.1.5 ipi

```
u32* clint_data::ipi
```

25.2.1.6 time_cmp

```
u64* clint_data::time_cmp
```

25.2.1.7 time_delta

```
u64 clint_data::time_delta
```

25.2.1.8 time_delta_computed

```
unsigned long clint_data::time_delta_computed
```

25.2.1.9 time_delta_reference

```
struct clint_data* clint_data::time_delta_reference
```

25.2.1.10 time_rd

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

25.2.1.11 time_val

```
u64* clint_data::time_val
```

25.2.1.12 time_wr

```
void(* clint_data::time_wr) (u64 value, volatile u64 *addr)
```

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

- include/sbi_utils/sys/clint.h

25.3 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

25.3.1 Field Documentation

25.3.1.1 boot_cpuid_phys

```
fdt32_t fdt_header::boot_cpuid_phys
```

25.3.1.2 last_comp_version

```
fdt32_t fdt_header::last_comp_version
```

25.3.1.3 magic

```
fdt32_t fdt_header::magic
```

25.3.1.4 off_dt_strings

```
fdt32_t fdt_header::off_dt_strings
```

25.3.1.5 off_dt_struct

```
fdt32_t fdt_header::off_dt_struct
```

25.3.1.6 off_mem_rsvmap

```
fdt32_t fdt_header::off_mem_rsvmap
```

25.3.1.7 size_dt_strings

```
fdt32_t fdt_header::size_dt_strings
```

25.3.1.8 size_dt_struct

```
fdt32_t fdt_header::size_dt_struct
```

25.3.1.9 totalsize

```
fdt32_t fdt_header::totalsize
```

25.3.1.10 version

```
fdt32_t fdt_header::version
```

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

- lib/utils/libfdt/fdt.h

25.4 fdt_ipi Struct Reference

```
#include <fdt_ipi.h>
```

Data Fields

- const struct fdt_match * match_table
- int(* cold_init)(void *fdt, int nodeoff, const struct fdt_match *match)
- int(* warm_init)(void)
- void(* exit)(void)
- void(* send)(u32 target_hart)
- void(* clear)(u32 target_hart)

25.4.1 Field Documentation

25.4.1.1 clear

```
void(* fdt_ipi::clear) (u32 target_hart)
```

25.4.1.2 cold_init

```
int(* fdt_ipi::cold_init) (void *fdt, int nodeoff, const struct fdt_match *match)
```

25.4.1.3 exit

```
void(* fdt_ipi::exit) (void)
```

25.4.1.4 match_table

```
const struct fdt_match* fdt_ipi::match_table
```

25.4.1.5 send

```
void(* fdt_ipi::send) (u32 target_hart)
```

25.4.1.6 warm_init

```
int(* fdt_ipi::warm_init) (void)
```

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

- include/sbi_utils/ipi/fdt_ipi.h

25.5 fdt_irqchip Struct Reference

```
#include <fdt_irqchip.h>
```

Data Fields

- const struct fdt_match ∗ match_table
- int(∗ cold_init)(void ∗fdt, int nodeoff, const struct fdt_match ∗match)
- int(∗ warm_init)(void)
- void(∗ exit)(void)

25.5.1 Field Documentation

25.5.1.1 cold_init

```
int(* fdt_irqchip::cold_init) (void *fdt, int nodeoff, const struct fdt_match *match)
```

25.5.1.2 exit

```
void(* fdt_irqchip::exit) (void)
```

25.5.1.3 match_table

```
const struct fdt_match* fdt_irqchip::match_table
```

25.5.1.4 warm_init

```
int(* fdt_irqchip::warm_init) (void)
```

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

- include/sbi_utils/irqchip/fdt_irqchip.h

25.6 fdt_match Struct Reference

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

Data Fields

- const char ∗ compatible
- void ∗ data

25.6.1 Field Documentation

25.6.1.1 compatible

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

25.6.1.2 data

`void* fdt_match::data`

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

- include/sbi_utils/fdt/fdt_helper.h

25.7 fdt_node_header Struct Reference

`#include <fdt.h>`

Data Fields

- fdt32_t tag
- char name [0]

25.7.1 Field Documentation

25.7.1.1 name

`char fdt_node_header::name[0]`

25.7.1.2 tag

`fdt32_t fdt_node_header::tag`

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

- lib/utils/libfdt/fdt.h

25.8 fdt_property Struct Reference

`#include <fdt.h>`

Data Fields

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

25.8.1 Field Documentation

25.8.1.1 data

`char fdt_property::data[0]`

25.8.1.2 len

`fdt32_t fdt_property::len`

25.8.1.3 nameoff

```
fdt32_t fdt_property::nameoff
```

25.8.1.4 tag

```
fdt32_t fdt_property::tag
```

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

- lib/utils/libfdt/fdt.h

25.9 fdt_reserve_entry Struct Reference

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

Data Fields

- fdt64_t address
- fdt64_t size

25.9.1 Field Documentation

25.9.1.1 address

```
fdt64_t fdt_reserve_entry::address
```

25.9.1.2 size

```
fdt64_t fdt_reserve_entry::size
```

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

- lib/utils/libfdt/fdt.h

25.10 fdt_reset Struct Reference

```
#include <fdt_reset.h>
```

Data Fields

- const struct fdt_match * match_table
- int(* init)(void *fdt, int nodeoff, const struct fdt_match *match)
- int(* system_reset)(u32 reset_type)

25.10.1 Field Documentation

25.10.1.1 init

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

25.10.1.2 match_table

```
const struct fdt_match* fdt_reset::match_table
```

25.10.1.3 system_reset

```
int(* fdt_reset::system_reset) (u32 reset_type)
```

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

- include/sbi_utils/reset/fdt_reset.h

25.11 fdt_serial Struct Reference

```
#include <fdt_serial.h>
```

Data Fields

- const struct fdt_match ∗ match_table
- int(∗ init)(void ∗fdt, int nodeoff, const struct fdt_match ∗match)
- void(∗ putc)(char ch)
- int(∗ getc)(void)

25.11.1 Field Documentation

25.11.1.1 getc

```
int(* fdt_serial::getc) (void)
```

25.11.1.2 init

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

25.11.1.3 match_table

```
const struct fdt_match* fdt_serial::match_table
```

25.11.1.4 putc

```
void(* fdt_serial::putc) (char ch)
```

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

- include/sbi_utils/serial/fdt_serial.h

25.12 fdt_timer Struct Reference

```
#include <fdt_timer.h>
```

Data Fields

- const struct fdt_match ∗ match_table
- int(∗ cold_init)(void ∗fdt, int nodeoff, const struct fdt_match ∗match)
- int(∗ warm_init)(void)
- void(∗ exit)(void)
- u64(∗ value)(void)
- void(∗ event_stop)(void)
- void(∗ event_start)(u64 next_event)

25.12.1 Field Documentation

25.12.1.1 cold_init

```
int(* fdt_timer::cold_init) (void *fdt, int nodeoff, const struct fdt_match *match)
```

25.12.1.2 event_start

```
void(* fdt_timer::event_start) (u64 next_event)
```

25.12.1.3 event_stop

```
void(* fdt_timer::event_stop) (void)
```

25.12.1.4 exit

```
void(* fdt_timer::exit) (void)
```

25.12.1.5 match_table

```
const struct fdt_match* fdt_timer::match_table
```

25.12.1.6 value

```
u64(* fdt_timer::value) (void)
```

25.12.1.7 warm_init

```
int(* fdt_timer::warm_init) (void)
```

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

- include/sbi_utils/timer/fdt_timer.h

25.13 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

25.13.1 Detailed Description

Representation dynamic info passed by previous booting stage

25.13.2 Field Documentation

25.13.2.1 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.

25.13.2.2 magic

`unsigned long fw_dynamic_info::magic`

Info magic

25.13.2.3 next_addr

`unsigned long fw_dynamic_info::next_addr`

Next booting stage address

25.13.2.4 next_mode

`unsigned long fw_dynamic_info::next_mode`

Next booting stage mode

25.13.2.5 options

`unsigned long fw_dynamic_info::options`

Options for OpenSBI library

25.13.2.6 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

25.14 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

25.14.1 Field Documentation

25.14.1.1 addr

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

25.14.1.2 baud

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

25.14.1.3 freq

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

25.14.1.4 reg_io_width

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

25.14.1.5 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

25.15 plic_data Struct Reference

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

Data Fields

- unsigned long addr
- unsigned long num_src

25.15.1 Field Documentation

25.15.1.1 addr

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

25.15.1.2 num_src

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

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

- include/sbi_utils/irqchip/plic.h

25.16 sbi_dlist Struct Reference

```
#include <sbi_list.h>
```

Data Fields

- struct sbi_dlist * next
- struct sbi_dlist * prev

25.16.1 Field Documentation

25.16.1.1 next

```
struct sbi_dlist* sbi_dlist::next
```

25.16.1.2 prev

```
struct sbi_dlist * sbi_dlist::prev
```

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

- include/sbi/sbi_list.h

25.17 sbi_ecall_extension Struct Reference

```
#include <sbi_ecall.h>
```

Data Fields

- struct sbi_dlist head
- unsigned long extid_start
- unsigned long extid_end
- int(* probe)(unsigned long extid, unsigned long *out_val)
- int(* handle)(unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct sbi_trap_info *out_trap)

25.17.1 Field Documentation

25.17.1.1 extid_end

```
unsigned long sbi_ecall_extension::extid_end
```

25.17.1.2 extid_start

```
unsigned long sbi_ecall_extension::extid_start
```

25.17.1.3 handle

```
int(* sbi_ecall_extension::handle) (unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct sbi_trap_info *out_trap)
```

25.17.1.4 head

```
struct sbi_dlist sbi_ecall_extension::head
```

25.17.1.5 probe

```
int(* sbi_ecall_extension::probe) (unsigned long extid, unsigned long *out_val)
```

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

- include/sbi/sbi_ecall.h

25.18 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

25.18.1 Field Documentation

25.18.1.1 avail

```
u16 sbi_fifo::avail
```

25.18.1.2 entry_size

```
u16 sbi_fifo::entry_size
```

25.18.1.3 num_entries

```
u16 sbi_fifo::num_entries
```

25.18.1.4 qlock

```
spinlock_t sbi_fifo::qlock
```

25.18.1.5 queue

```
void* sbi_fifo::queue
```

25.18.1.6 tail

```
u16 sbi_fifo::tail
```

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

- include/sbi/sbi_fifo.h

25.19 sbi_hartmask Struct Reference

```
#include <sbi_hartmask.h>
```

Public Member Functions

- DECLARE_BITMAP (bits, SBI_HARTMASK_MAX_BITS)

25.19.1 Detailed Description

Representation of hartmask

25.19.2 Member Function Documentation

25.19.2.1 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

25.20 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_hartid, void ∗data)
- void(∗ sync)(struct sbi_scratch ∗scratch)
- void(∗ process)(struct sbi_scratch ∗scratch)

25.20.1 Detailed Description

IPI event operations or callbacks

25.20.2 Field Documentation

25.20.2.1 name

```
char sbi_ipi_event_ops::name[32]
```

Name of the IPI event operations

25.20.2.2 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.

25.20.2.3 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.

25.20.2.4 update

`int(* sbi_ipi_event_ops::update) (struct sbi_scratch *scratch, struct sbi_scratch *remote_↩scratch, u32 remote_hartid, 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.
The documentation for this struct was generated from the following file:

- include/sbi/sbi_ipi.h

25.21 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
- unsigned long platform_ops_addr
- unsigned long firmware_context
- const u32 * hart_index2id

25.21.1 Detailed Description

Representation of a platform

25.21.2 Field Documentation

25.21.2.1 features

`u64 sbi_platform::features`
Supported features

25.21.2.2 firmware_context

`unsigned long sbi_platform::firmware_context`
Pointer to system firmware specific context

25.21.2.3 hart_count

`u32 sbi_platform::hart_count`
Total number of HARTs

25.21.2.4 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

25.21.2.5 hart_stack_size

`u32 sbi_platform::hart_stack_size`
Per-HART stack size for exception/interrupt handling

25.21.2.6 name

`char sbi_platform::name[64]`
Name of the platform

25.21.2.7 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

25.21.2.8 platform_ops_addr

`unsigned long sbi_platform::platform_ops_addr`
Pointer to sbi platform operations

25.21.2.9 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
The documentation for this struct was generated from the following file:

- include/sbi/sbi_platform.h

25.22 sbi_platform_operations Struct Reference

`#include <sbi_platform.h>`

Data Fields

- 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)
- u32(∗ pmp_region_count)(u32 hartid)
- int(∗ pmp_region_info)(u32 hartid, u32 index, ulong ∗prot, ulong ∗addr, ulong ∗log2size)
- void(∗ console_putc)(char ch)
- int(∗ console_getc)(void)
- int(∗ console_init)(void)
- int(∗ irqchip_init)(bool cold_boot)
- void(∗ irqchip_exit)(void)
- void(∗ ipi_send)(u32 target_hart)
- void(∗ ipi_clear)(u32 target_hart)
- int(∗ ipi_init)(bool cold_boot)
- void(∗ ipi_exit)(void)
- u64(∗ get_tlbr_flush_limit)(void)
- u64(∗ timer_value)(void)
- void(∗ timer_event_start)(u64 next_event)
- void(∗ timer_event_stop)(void)
- int(∗ timer_init)(bool cold_boot)
- void(∗ timer_exit)(void)
- int(∗ hart_start)(u32 hartid, ulong saddr)
- int(∗ hart_stop)(void)
- int(∗ system_reset)(u32 reset_type)
- int(∗ vendor_ext_check)(long extid)
- int(∗ vendor_ext_provider)(long extid, long funcid, unsigned long ∗args, unsigned long ∗out_value, struct sbi_trap_info ∗out_trap)

25.22.1 Detailed Description

Platform functions

25.22.2 Field Documentation

25.22.2.1 console_getc

`int(* sbi_platform_operations::console_getc) (void)`

Read a character from the platform console input

25.22.2.2 console_init

`int(* sbi_platform_operations::console_init) (void)`

Initialize the platform console

25.22.2.3 console_putc

`void(* sbi_platform_operations::console_putc) (char ch)`

Write a character to the platform console output

25.22.2.4 early_exit

`void(* sbi_platform_operations::early_exit) (void)`

Platform early exit

25.22.2.5 early_init

`int(* sbi_platform_operations::early_init) (bool cold_boot)`

Platform early initialization

25.22.2.6 final_exit

```
void(* sbi_platform_operations::final_exit) (void)
```
Platform final exit

25.22.2.7 final_init

```
int(* sbi_platform_operations::final_init) (bool cold_boot)
```
Platform final initialization

25.22.2.8 get_tlbr_flush_limit

```
u64(* sbi_platform_operations::get_tlbr_flush_limit) (void)
```
Get tlb flush limit value

25.22.2.9 hart_start

```
int(* sbi_platform_operations::hart_start) (u32 hartid, ulong saddr)
```
Bringup the given hart

25.22.2.10 hart_stop

```
int(* sbi_platform_operations::hart_stop) (void)
```
Stop the current hart from running. This call doesn't expect to return if success.

25.22.2.11 ipi_clear

```
void(* sbi_platform_operations::ipi_clear) (u32 target_hart)
```
Clear IPI for a target HART

25.22.2.12 ipi_exit

```
void(* sbi_platform_operations::ipi_exit) (void)
```
Exit IPI for current HART

25.22.2.13 ipi_init

```
int(* sbi_platform_operations::ipi_init) (bool cold_boot)
```
Initialize IPI for current HART

25.22.2.14 ipi_send

```
void(* sbi_platform_operations::ipi_send) (u32 target_hart)
```
Send IPI to a target HART

25.22.2.15 irqchip_exit

```
void(* sbi_platform_operations::irqchip_exit) (void)
```
Exit the platform interrupt controller for current HART

25.22.2.16 irqchip_init

```
int(* sbi_platform_operations::irqchip_init) (bool cold_boot)
```
Initialize the platform interrupt controller for current HART

25.22.2.17 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.

25.22.2.18 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.

25.22.2.19 pmp_region_count

```
u32(* sbi_platform_operations::pmp_region_count) (u32 hartid)
```

Get number of PMP regions for given HART

25.22.2.20 pmp_region_info

```
int(* sbi_platform_operations::pmp_region_info) (u32 hartid, u32 index, ulong *prot, ulong
*addr, ulong *log2size)
```

Get PMP regions details (namely: protection, base address, and size) for given HART

25.22.2.21 system_reset

```
int(* sbi_platform_operations::system_reset) (u32 reset_type)
```

25.22.2.22 timer_event_start

```
void(* sbi_platform_operations::timer_event_start) (u64 next_event)
```

Start platform timer event for current HART

25.22.2.23 timer_event_stop

```
void(* sbi_platform_operations::timer_event_stop) (void)
```

Stop platform timer event for current HART

25.22.2.24 timer_exit

```
void(* sbi_platform_operations::timer_exit) (void)
```

Exit platform timer for current HART

25.22.2.25 timer_init

```
int(* sbi_platform_operations::timer_init) (bool cold_boot)
```

Initialize platform timer for current HART

25.22.2.26 timer_value

```
u64(* sbi_platform_operations::timer_value) (void)
```

Get platform timer value

25.22.2.27 vendor_ext_check

```
int(* sbi_platform_operations::vendor_ext_check) (long extid)
```

platform specific SBI extension implementation probe function

25.22.2.28 vendor_ext_provider

```
int(* sbi_platform_operations::vendor_ext_provider) (long extid, long funcid, unsigned long
*args, unsigned long *out_value, struct sbi_trap_info *out_trap)
```

platform specific SBI extension implementation provider

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

- include/sbi/sbi_platform.h

25.23 sbi_scratch Struct Reference

```
#include <sbi_scratch.h>
```

Data Fields

- unsigned long fw_start
- unsigned long fw_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 tmp0
- unsigned long options

25.23.1 Detailed Description

Representation of per-HART scratch space

25.23.2 Field Documentation

25.23.2.1 fw_size

`unsigned long sbi_scratch::fw_size`
Size (in bytes) of firmware linked to OpenSBI library

25.23.2.2 fw_start

`unsigned long sbi_scratch::fw_start`
Start (or base) address of firmware linked to OpenSBI library

25.23.2.3 hartid_to_scratch

`unsigned long sbi_scratch::hartid_to_scratch`
Address of HART ID to sbi_scratch conversion function

25.23.2.4 next_addr

`unsigned long sbi_scratch::next_addr`
Address of next booting stage for this HART

25.23.2.5 next_arg1

`unsigned long sbi_scratch::next_arg1`
Arg1 (or 'a1' register) of next booting stage for this HART

25.23.2.6 next_mode

`unsigned long sbi_scratch::next_mode`
Priviledge mode of next booting stage for this HART

25.23.2.7 options

`unsigned long sbi_scratch::options`
Options for OpenSBI library

25.23.2.8 platform_addr

`unsigned long sbi_scratch::platform_addr`
Address of sbi_platform

25.23.2.9 tmp0

`unsigned long sbi_scratch::tmp0`
Temporary storage

25.23.2.10 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

25.24 sbi_tlb_info Struct Reference

```
#include <sbi_tlb.h>
```

Data Fields

- unsigned long start
- unsigned long size
- unsigned long asid
- unsigned long vmid
- unsigned long type
- struct sbi_hartmask smask

25.24.1 Field Documentation

25.24.1.1 asid

`unsigned long sbi_tlb_info::asid`

25.24.1.2 size

`unsigned long sbi_tlb_info::size`

25.24.1.3 smask

`struct sbi_hartmask sbi_tlb_info::smask`

25.24.1.4 start

`unsigned long sbi_tlb_info::start`

25.24.1.5 type

`unsigned long sbi_tlb_info::type`

25.24.1.6 vmid

`unsigned long sbi_tlb_info::vmid`

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

- include/sbi/sbi_tlb.h

25.25 sbi_trap_info Struct Reference

`#include <sbi_trap.h>`

Data Fields

- unsigned long epc
- unsigned long cause
- unsigned long tval
- unsigned long tval2
- unsigned long tinst

25.25.1 Detailed Description

Representation of trap details

25.25.2 Field Documentation

25.25.2.1 cause

`unsigned long sbi_trap_info::cause`

cause Trap exception cause

25.25.2.2 epc

`unsigned long sbi_trap_info::epc`

epc Trap program counter

25.25.2.3 tinst

`unsigned long sbi_trap_info::tinst`

tinst Trap instruction

25.25.2.4 tval

`unsigned long sbi_trap_info::tval`

tval Trap value

25.25.2.5 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

25.26 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

25.26.1 Detailed Description

Representation of register state at time of trap/interrupt

25.26.2 Field Documentation

25.26.2.1 a0

`unsigned long sbi_trap_regs::a0`

a0 register state

25.26.2.2 a1

`unsigned long sbi_trap_regs::a1`

a1 register state

25.26.2.3 a2

`unsigned long sbi_trap_regs::a2`
a2 register state

25.26.2.4 a3

`unsigned long sbi_trap_regs::a3`
a3 register state

25.26.2.5 a4

`unsigned long sbi_trap_regs::a4`
a4 register state

25.26.2.6 a5

`unsigned long sbi_trap_regs::a5`
a5 register state

25.26.2.7 a6

`unsigned long sbi_trap_regs::a6`
a6 register state

25.26.2.8 a7

`unsigned long sbi_trap_regs::a7`
a7 register state

25.26.2.9 gp

`unsigned long sbi_trap_regs::gp`
gp register state

25.26.2.10 mepc

`unsigned long sbi_trap_regs::mepc`
mepc register state

25.26.2.11 mstatus

`unsigned long sbi_trap_regs::mstatus`
mstatus register state

25.26.2.12 mstatusH

`unsigned long sbi_trap_regs::mstatusH`
mstatusH register state (only for 32-bit)

25.26.2.13 ra

`unsigned long sbi_trap_regs::ra`
ra register state

25.26.2.14 s0

`unsigned long sbi_trap_regs::s0`
s0 register state

25.26.2.15 s1

`unsigned long sbi_trap_regs::s1`
s1 register state

25.26.2.16 s10

`unsigned long sbi_trap_regs::s10`
s10 register state

25.26.2.17 s11

`unsigned long sbi_trap_regs::s11`
s11 register state

25.26.2.18 s2

`unsigned long sbi_trap_regs::s2`
s2 register state

25.26.2.19 s3

`unsigned long sbi_trap_regs::s3`
s3 register state

25.26.2.20 s4

`unsigned long sbi_trap_regs::s4`
s4 register state

25.26.2.21 s5

`unsigned long sbi_trap_regs::s5`
s5 register state

25.26.2.22 s6

`unsigned long sbi_trap_regs::s6`
s6 register state

25.26.2.23 s7

`unsigned long sbi_trap_regs::s7`
s7 register state

25.26.2.24 s8

`unsigned long sbi_trap_regs::s8`
s8 register state

25.26.2.25 s9

`unsigned long sbi_trap_regs::s9`
s9 register state

25.26.2.26 sp

`unsigned long sbi_trap_regs::sp`
sp register state

25.26.2.27 t0

`unsigned long sbi_trap_regs::t0`
t0 register state

25.26.2.28 t1

`unsigned long sbi_trap_regs::t1`
t1 register state

25.26.2.29 t2

`unsigned long sbi_trap_regs::t2`
t2 register state

25.26.2.30 t3

`unsigned long sbi_trap_regs::t3`
t3 register state

25.26.2.31 t4

`unsigned long sbi_trap_regs::t4`
t4 register state

25.26.2.32 t5

`unsigned long sbi_trap_regs::t5`
t5 register state

25.26.2.33 t6

`unsigned long sbi_trap_regs::t6`
t6 register state

25.26.2.34 tp

`unsigned long sbi_trap_regs::tp`
tp register state

25.26.2.35 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

25.27 spinlock_t Struct Reference

`#include <riscv_locks.h>`

Data Fields

- volatile long lock

25.27.1 Field Documentation

25.27.1.1 lock

`volatile long spinlock_t::lock`

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

- include/sbi/riscv_locks.h

Chapter 26

File Documentation

26.1 docs/contributing.md File Reference

26.2 docs/firmware/fw.md File Reference

26.3 docs/firmware/fw_dynamic.md File Reference

26.4 docs/firmware/fw_jump.md File Reference

26.5 docs/firmware/fw_payload.md File Reference

26.6 docs/firmware/payload_linux.md File Reference

26.7 docs/firmware/payload_uboot.md File Reference

26.8 docs/library_usage.md File Reference

26.9 docs/platform/andes-ae350.md File Reference

26.10 docs/platform/fpga-ariane.md File Reference

26.11 docs/platform/fpga-openpiton.md File Reference

26.12 docs/platform/generic.md File Reference

26.13 docs/platform/nuclei_ux600.md File Reference

26.14 docs/platform/platform.md File Reference

26.15 docs/platform/qemu_virt.md File Reference

26.16 docs/platform/shakti_cclass.md File Reference

26.17 docs/platform/sifive_fu540.md File Reference

26.18 docs/platform/spike.md File Reference

26.19 docs/platform/thead-c910.md File Reference

26.20 docs/platform_guide.md File Reference

26.21 docs/platform_requirements.md File Reference

26.22 include/sbi/fw_dynamic.h File Reference

```
#include <sbi/riscv_asm.h>
```

```
#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_POINTER__)
- #define FW_DYNAMIC_INFO_VERSION_OFFSET (1 ∗ __SIZEOF_POINTER__)
- #define FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET (2 ∗ __SIZEOF_POINTER__)
- #define FW_DYNAMIC_INFO_NEXT_MODE_OFFSET (3 ∗ __SIZEOF_POINTER__)
- #define FW_DYNAMIC_INFO_OPTIONS_OFFSET (4 ∗ __SIZEOF_POINTER__)
- #define FW_DYNAMIC_INFO_BOOT_HART_OFFSET (5 ∗ __SIZEOF_POINTER__)
- #define FW_DYNAMIC_INFO_MAGIC_VALUE 0x4942534f
- #define FW_DYNAMIC_INFO_VERSION_MAX 0x2
- #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

26.22.1 Macro Definition Documentation

26.22.1.1 FW_DYNAMIC_INFO_BOOT_HART_OFFSET

```
#define FW_DYNAMIC_INFO_BOOT_HART_OFFSET (5 * __SIZEOF_POINTER__)
```

Offset of boot_hart member in fw_dynamic_info (version >= 2)

26.22.1.2 FW_DYNAMIC_INFO_MAGIC_OFFSET

```
#define FW_DYNAMIC_INFO_MAGIC_OFFSET (0 * __SIZEOF_POINTER__)
```

Offset of magic member in fw_dynamic_info

26.22.1.3 FW_DYNAMIC_INFO_MAGIC_VALUE

```
#define FW_DYNAMIC_INFO_MAGIC_VALUE 0x4942534f
```

Expected value of info magic ('OSBI' ascii string in hex)

26.22.1.4 FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET

```
#define FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET (2 * __SIZEOF_POINTER__)
```

Offset of next_addr member in fw_dynamic_info (version >= 1)

26.22.1.5 FW_DYNAMIC_INFO_NEXT_MODE_M

```
#define FW_DYNAMIC_INFO_NEXT_MODE_M 0x3
```

26.22.1.6 FW_DYNAMIC_INFO_NEXT_MODE_OFFSET

```
#define FW_DYNAMIC_INFO_NEXT_MODE_OFFSET (3 * __SIZEOF_POINTER__)
```

Offset of next_mode member in fw_dynamic_info (version >= 1)

26.22.1.7 FW_DYNAMIC_INFO_NEXT_MODE_S

```
#define FW_DYNAMIC_INFO_NEXT_MODE_S 0x1
```

26.22.1.8 FW_DYNAMIC_INFO_NEXT_MODE_U

```
#define FW_DYNAMIC_INFO_NEXT_MODE_U 0x0
```

Possible next mode values

26.22.1.9 FW_DYNAMIC_INFO_OPTIONS_OFFSET

```
#define FW_DYNAMIC_INFO_OPTIONS_OFFSET (4 * __SIZEOF_POINTER__)
```

Offset of options member in fw_dynamic_info (version >= 1)

26.22.1.10 FW_DYNAMIC_INFO_VERSION_MAX

```
#define FW_DYNAMIC_INFO_VERSION_MAX 0x2
```

Maximum supported info version

26.22.1.11 FW_DYNAMIC_INFO_VERSION_OFFSET

```
#define FW_DYNAMIC_INFO_VERSION_OFFSET (1 * __SIZEOF_POINTER__)
```

Offset of version member in fw_dynamic_info

26.22.2 Variable Documentation

26.22.2.1 __packed

```
struct fw_dynamic_info __packed
```

26.23 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_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 current_hartid() ((unsigned int)csr_read(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_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 *size)

26.23.1 Macro Definition Documentation

26.23.1.1 __ASM_STR

```
#define __ASM_STR(
            x ) #x
```

26.23.1.2 csr_clear

```
#define csr_clear(
            csr,
            val )
```

Value:

```
({                                                          \
    unsigned long __v = (unsigned long)(val);           \
    __asm__ __volatile__("csrc " __ASM_STR(csr) ", %0" \
                :                                       \
                : "rK"(__v)                             \
                : "memory");                            \
})
```

26.23.1.3 csr_read

```
#define csr_read(
            csr )
```

Value:

```
({                                                      \
    register unsigned long __v;                     \
    __asm__ __volatile__("csrr %0, " __ASM_STR(csr) \
                : "=r"(__v)                         \
                :                                   \
                : "memory");                        \
    __v;                                            \
})
```

26.23.1.4 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;                                                        \
})
```

26.23.1.5 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;                                                            \
})
```

26.23.1.6 csr_set

```
#define csr_set(
            csr,
            val )
```

Value:

```
({                                                          \
    unsigned long __v = (unsigned long)(val);               \
    __asm__ __volatile__("csrs " __ASM_STR(csr) ", %0" \
                 :                                          \
                 : "rK"(__v)                                \
                 : "memory");                               \
})
```

26.23.1.7 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;                                                            \
})
```

26.23.1.8 csr_write

```
#define csr_write(
            csr,
            val )
```

Value:

```
({                                                          \
    unsigned long __v = (unsigned long)(val);               \
    __asm__ __volatile__("csrw " __ASM_STR(csr) ", %0" \
                 :                                          \
                 : "rK"(__v)                                \
                 : "memory");                               \
})
```

26.23.1.9 current_hartid

```
#define current_hartid( ) ((unsigned int)csr_read(CSR_MHARTID))
```

26.23.1.10 LGREG

```
#define LGREG __REG_SEL(3, 2)
```

26.23.1.11 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);\
})
```

26.23.1.12 PAGE_MASK

```
#define PAGE_MASK (~(PAGE_SIZE - 1))
```

26.23.1.13 PAGE_SHIFT

```
#define PAGE_SHIFT (12)
```

26.23.1.14 PAGE_SIZE

```
#define PAGE_SIZE (_AC(1, UL) << PAGE_SHIFT)
```

26.23.1.15 REG_L

```
#define REG_L __REG_SEL(ld, lw)
```

26.23.1.16 REG_S

```
#define REG_S __REG_SEL(sd, sw)
```

26.23.1.17 SZREG

```
#define SZREG __REG_SEL(8, 4)
```

26.23.1.18 wfi

```
#define wfi( )
```

Value:

```
    do {                                                \
        __asm__ __volatile__("wfi" ::: "memory"); \
    } while (0)
```

26.23.2 Function Documentation

26.23.2.1 csr_read_num()

```
unsigned long csr_read_num (
            int csr_num )
```

Here is the caller graph for this function:

26.23.2.2 csr_write_num()

```
void csr_write_num (
            int csr_num,
            unsigned long val )
```

Here is the caller graph for this function:

26.23.2.3 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:

26.23.2.4 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:

26.23.2.5 misa_xlen()

```
int misa_xlen (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.23.2.6 pmp_get()

```
int pmp_get (
            unsigned int n,
```

```
            unsigned long * prot_out,
            unsigned long * addr_out,
            unsigned long * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.23.2.7 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:

26.24 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)

26.24.1 Macro Definition Documentation

26.24.1.1 ATOMIC_INIT

```
#define ATOMIC_INIT(
            _lptr,
            val ) (_lptr)->counter = (val)
```

26.24.1.2 ATOMIC_INITIALIZER

```
#define ATOMIC_INITIALIZER(
               val )
```

Value:

```
{                                   \
    .counter = (val), \
}
```

26.24.2 Function Documentation

26.24.2.1 atomic_add_return()

```
long atomic_add_return (
            atomic_t * atom,
            long value )
```

26.24.2.2 atomic_clear_bit()

```
int atomic_clear_bit (
            int nr,
            atomic_t * atom )  [inline]
```

Clear a bit in an atomic variable and return the new value. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:

26.24.2.3 atomic_cmpxchg()

```
long atomic_cmpxchg (
            atomic_t * atom,
            long oldval,
            long newval )
```

Here is the caller graph for this function:

26.24.2.4 atomic_raw_clear_bit()

```
int atomic_raw_clear_bit (
            int nr,
            volatile unsigned long * addr )  [inline]
```

Clear a bit in any address and return the new value . @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:

26.24.2.5 atomic_raw_set_bit()

```
int atomic_raw_set_bit (
            int nr,
            volatile unsigned long * addr )  [inline]
```

Set a bit in any address and return the new value . @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:

26.24.2.6 atomic_raw_xchg_uint()

```
unsigned int atomic_raw_xchg_uint (
            volatile unsigned int * ptr,
            unsigned int newval )
```

26.24.2.7 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:

26.24.2.8 atomic_read()

```
long atomic_read (
            atomic_t * atom )
```

Here is the caller graph for this function:

26.24.2.9 atomic_set_bit()

```
int atomic_set_bit (
            int nr,
            atomic_t * atom )  [inline]
```

Set a bit in an atomic variable and return the new value. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:

26.24.2.10 atomic_sub_return()

```
long atomic_sub_return (
            atomic_t * atom,
```

```
            long value )
```

26.24.2.11 atomic_write()

```
void atomic_write (
            atomic_t * atom,
            long value )
```

26.24.2.12 atomic_xchg()

```
long atomic_xchg (
            atomic_t * atom,
            long newval )
```

Here is the caller graph for this function:

26.25 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 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() asm volatile ("" : : : "memory")
- #define __smp_store_release(p, v)
- #define __smp_load_acquire(p)

26.25.1 Macro Definition Documentation

26.25.1.1 __smp_load_acquire

```
#define __smp_load_acquire(
            p )
```

Value:

```
({                                  \
    typeof(*p) ___p1 = *(p); \
    RISCV_FENCE(r, rw);      \
    ___p1;                   \
})
```

26.25.1.2 __smp_store_release

```
#define __smp_store_release(
            p,
            v )
```

Value:

```
do {                        \
    RISCV_FENCE(rw, w); \
    *(p) = (v);         \
} while (0)
```

26.25.1.3 cpu_relax

```
#define cpu_relax( ) asm volatile ("" : : : "memory")
```

26.25.1.4 mb

```
#define mb( ) RISCV_FENCE(iorw,iorw)
```

26.25.1.5 RISCV_ACQUIRE_BARRIER

```
#define RISCV_ACQUIRE_BARRIER "\tfence r , rw\n"
```

26.25.1.6 RISCV_FENCE

```
#define RISCV_FENCE(
            p,
            s ) __asm__ __volatile__ ("fence " #p "," #s : : : "memory")
```

26.25.1.7 RISCV_RELEASE_BARRIER

```
#define RISCV_RELEASE_BARRIER "\tfence rw, w\n"
```

26.25.1.8 rmb

```
#define rmb( ) RISCV_FENCE(ir,ir)
```

26.25.1.9 smp_mb

```
#define smp_mb( ) RISCV_FENCE(rw,rw)
```

26.25.1.10 smp_rmb

```
#define smp_rmb( ) RISCV_FENCE(r,r)
```

26.25.1.11 smp_wmb

```
#define smp_wmb( ) RISCV_FENCE(w,w)
```

26.25.1.12 wmb

```
#define wmb( ) RISCV_FENCE(ow,ow)
```

26.26 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(0x01800000)
- #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 MSTATUS32_SD _UL(0x80000000)
- #define MSTATUSH_SBE _UL(0x00000010)
- #define MSTATUSH_MBE _UL(0x00000020)
- #define MSTATUSH_MPV _UL(0x00000080)
- #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 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 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 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 16
- #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 CSR_USTATUS 0x0
- #define CSR_FFLAGS 0x1
- #define CSR_FRM 0x2
- #define CSR_FCSR 0x3
- #define CSR_CYCLE 0xc00
- #define CSR_UIE 0x4
- #define CSR_UTVEC 0x5
- #define CSR_USCRATCH 0x40
- #define CSR_UEPC 0x41
- #define CSR_UCAUSE 0x42

- #define CSR_UTVAL 0x43
- #define CSR_UIP 0x44
- #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_SSTATUS 0x100
- #define CSR_SIE 0x104
- #define CSR_STVEC 0x105
- #define CSR_SCOUNTEREN 0x106
- #define CSR_SSCRATCH 0x140
- #define CSR_SEPC 0x141
- #define CSR_SCAUSE 0x142
- #define CSR_STVAL 0x143
- #define CSR_SIP 0x144
- #define CSR_SATP 0x180
- #define CSR_HSTATUS 0x600
- #define CSR_HEDELEG 0x602
- #define CSR_HIDELEG 0x603
- #define CSR_HIE 0x604
- #define CSR_HTIMEDELTA 0x605
- #define CSR_HTIMEDELTAH 0x615
- #define CSR_HCOUNTERNEN 0x606
- #define CSR_HGEIE 0x607
- #define CSR_HTVAL 0x643
- #define CSR_HIP 0x644
- #define CSR_HTINST 0x64a
- #define CSR_HGATP 0x680

- #define CSR_HGEIP 0xe07
- #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_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_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_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_TSELECT 0x7a0
- #define CSR_TDATA1 0x7a1
- #define CSR_TDATA2 0x7a2
- #define CSR_TDATA3 0x7a3
- #define CSR_DCSR 0x7b0
- #define CSR_DPC 0x7b1
- #define CSR_DSCRATCH 0x7b2
- #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_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_MVENDORID 0xf11
- #define CSR_MARCHID 0xf12
- #define CSR_MIMPID 0xf13
- #define CSR_MHARTID 0xf14
- #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_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 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_HYPERVISOR_ECALL 0x9
- #define CAUSE_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_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 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_MASK_WFI 0xffffff00
- #define INSN_MATCH_WFI 0x10500000
- #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_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) >> 12) & 7)
- #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 MASK_FUNCT3 0x7000

26.26.1 Macro Definition Documentation

26.26.1.1 CAUSE_BREAKPOINT

```
#define CAUSE_BREAKPOINT 0x3
```

26.26.1.2 CAUSE_FETCH_ACCESS

```
#define CAUSE_FETCH_ACCESS 0x1
```

26.26.1.3 CAUSE_FETCH_GUEST_PAGE_FAULT

```
#define CAUSE_FETCH_GUEST_PAGE_FAULT 0x14
```

26.26.1.4 CAUSE_FETCH_PAGE_FAULT

```
#define CAUSE_FETCH_PAGE_FAULT 0xc
```

26.26.1.5 CAUSE_HYPERVISOR_ECALL

```
#define CAUSE_HYPERVISOR_ECALL 0x9
```

26.26.1.6 CAUSE_ILLEGAL_INSTRUCTION

```
#define CAUSE_ILLEGAL_INSTRUCTION 0x2
```

26.26.1.7 CAUSE_LOAD_ACCESS

```
#define CAUSE_LOAD_ACCESS 0x5
```

26.26.1.8 CAUSE_LOAD_GUEST_PAGE_FAULT

```
#define CAUSE_LOAD_GUEST_PAGE_FAULT 0x15
```

26.26.1.9 CAUSE_LOAD_PAGE_FAULT

```
#define CAUSE_LOAD_PAGE_FAULT 0xd
```

26.26.1.10 CAUSE_MACHINE_ECALL

```
#define CAUSE_MACHINE_ECALL 0xb
```

26.26.1.11 CAUSE_MISALIGNED_FETCH

```
#define CAUSE_MISALIGNED_FETCH 0x0
```

26.26.1.12 CAUSE_MISALIGNED_LOAD

```
#define CAUSE_MISALIGNED_LOAD 0x4
```

26.26.1.13 CAUSE_MISALIGNED_STORE

```
#define CAUSE_MISALIGNED_STORE 0x6
```

26.26.1.14 CAUSE_STORE_ACCESS

```
#define CAUSE_STORE_ACCESS 0x7
```

26.26.1.15 CAUSE_STORE_GUEST_PAGE_FAULT

```
#define CAUSE_STORE_GUEST_PAGE_FAULT 0x17
```

26.26.1.16 CAUSE_STORE_PAGE_FAULT

```
#define CAUSE_STORE_PAGE_FAULT 0xf
```

26.26.1.17 CAUSE_SUPERVISOR_ECALL

```
#define CAUSE_SUPERVISOR_ECALL 0xa
```

26.26.1.18 CAUSE_USER_ECALL

```
#define CAUSE_USER_ECALL 0x8
```

26.26.1.19 CAUSE_VIRTUAL_INST_FAULT

```
#define CAUSE_VIRTUAL_INST_FAULT 0x16
```

26.26.1.20 CSR_CYCLE

```
#define CSR_CYCLE 0xc00
```

26.26.1.21 CSR_CYCLEH

```
#define CSR_CYCLEH 0xc80
```

26.26.1.22 CSR_DCSR

```
#define CSR_DCSR 0x7b0
```

26.26.1.23 CSR_DPC

```
#define CSR_DPC 0x7b1
```

26.26.1.24 CSR_DSCRATCH

```
#define CSR_DSCRATCH 0x7b2
```

26.26.1.25 CSR_FCSR

```
#define CSR_FCSR 0x3
```

26.26.1.26 CSR_FFLAGS

```
#define CSR_FFLAGS 0x1
```

26.26.1.27 CSR_FRM

```
#define CSR_FRM 0x2
```

26.26.1.28 CSR_HCOUNTERNEN

```
#define CSR_HCOUNTERNEN 0x606
```

26.26.1.29 CSR_HEDELEG

```
#define CSR_HEDELEG 0x602
```

26.26.1.30 CSR_HGATP

```
#define CSR_HGATP 0x680
```

26.26.1.31 CSR_HGEIE

```
#define CSR_HGEIE 0x607
```

26.26.1.32 CSR_HGEIP

```
#define CSR_HGEIP 0xe07
```

26.26.1.33 CSR_HIDELEG

```
#define CSR_HIDELEG 0x603
```

26.26.1.34 CSR_HIE

```
#define CSR_HIE 0x604
```

26.26.1.35 CSR_HIP

```
#define CSR_HIP 0x644
```

26.26.1.36 CSR_HPMCOUNTER10

```
#define CSR_HPMCOUNTER10 0xc0a
```

26.26.1.37 CSR_HPMCOUNTER10H

```
#define CSR_HPMCOUNTER10H 0xc8a
```

26.26.1.38 CSR_HPMCOUNTER11

```
#define CSR_HPMCOUNTER11 0xc0b
```

26.26.1.39 CSR_HPMCOUNTER11H

```
#define CSR_HPMCOUNTER11H 0xc8b
```

26.26.1.40 CSR_HPMCOUNTER12

```
#define CSR_HPMCOUNTER12 0xc0c
```

26.26.1.41 CSR_HPMCOUNTER12H

```
#define CSR_HPMCOUNTER12H 0xc8c
```

26.26.1.42 CSR_HPMCOUNTER13

```
#define CSR_HPMCOUNTER13 0xc0d
```

26.26.1.43 CSR_HPMCOUNTER13H

```
#define CSR_HPMCOUNTER13H 0xc8d
```

26.26.1.44 CSR_HPMCOUNTER14

```
#define CSR_HPMCOUNTER14 0xc0e
```

26.26.1.45 CSR_HPMCOUNTER14H

```
#define CSR_HPMCOUNTER14H 0xc8e
```

26.26.1.46 CSR_HPMCOUNTER15

```
#define CSR_HPMCOUNTER15 0xc0f
```

26.26.1.47 CSR_HPMCOUNTER15H

```
#define CSR_HPMCOUNTER15H 0xc8f
```

26.26.1.48 CSR_HPMCOUNTER16

```
#define CSR_HPMCOUNTER16 0xc10
```

26.26.1.49 CSR_HPMCOUNTER16H

```
#define CSR_HPMCOUNTER16H 0xc90
```

26.26.1.50 CSR_HPMCOUNTER17

```
#define CSR_HPMCOUNTER17 0xc11
```

26.26.1.51 CSR_HPMCOUNTER17H

```
#define CSR_HPMCOUNTER17H 0xc91
```

26.26.1.52 CSR_HPMCOUNTER18

```
#define CSR_HPMCOUNTER18 0xc12
```

26.26.1.53 CSR_HPMCOUNTER18H

```
#define CSR_HPMCOUNTER18H 0xc92
```

26.26.1.54 CSR_HPMCOUNTER19

```
#define CSR_HPMCOUNTER19 0xc13
```

26.26.1.55 CSR_HPMCOUNTER19H

```
#define CSR_HPMCOUNTER19H 0xc93
```

26.26.1.56 CSR_HPMCOUNTER20

```
#define CSR_HPMCOUNTER20 0xc14
```

26.26.1.57 CSR_HPMCOUNTER20H

```
#define CSR_HPMCOUNTER20H 0xc94
```

26.26.1.58 CSR_HPMCOUNTER21

```
#define CSR_HPMCOUNTER21 0xc15
```

26.26.1.59 CSR_HPMCOUNTER21H

```
#define CSR_HPMCOUNTER21H 0xc95
```

26.26.1.60 CSR_HPMCOUNTER22

```
#define CSR_HPMCOUNTER22 0xc16
```

26.26.1.61 CSR_HPMCOUNTER22H

```
#define CSR_HPMCOUNTER22H 0xc96
```

26.26.1.62 CSR_HPMCOUNTER23

```
#define CSR_HPMCOUNTER23 0xc17
```

26.26.1.63 CSR_HPMCOUNTER23H

```
#define CSR_HPMCOUNTER23H 0xc97
```

26.26.1.64 CSR_HPMCOUNTER24

```
#define CSR_HPMCOUNTER24 0xc18
```

26.26.1.65 CSR_HPMCOUNTER24H

```
#define CSR_HPMCOUNTER24H 0xc98
```

26.26.1.66 CSR_HPMCOUNTER25

```
#define CSR_HPMCOUNTER25 0xc19
```

26.26.1.67 CSR_HPMCOUNTER25H

```
#define CSR_HPMCOUNTER25H 0xc99
```

26.26.1.68 CSR_HPMCOUNTER26

```
#define CSR_HPMCOUNTER26 0xc1a
```

26.26.1.69 CSR_HPMCOUNTER26H

```
#define CSR_HPMCOUNTER26H 0xc9a
```

26.26.1.70 CSR_HPMCOUNTER27

```
#define CSR_HPMCOUNTER27 0xc1b
```

26.26.1.71 CSR_HPMCOUNTER27H

```
#define CSR_HPMCOUNTER27H 0xc9b
```

26.26.1.72 CSR_HPMCOUNTER28

```
#define CSR_HPMCOUNTER28 0xc1c
```

26.26.1.73 CSR_HPMCOUNTER28H

```
#define CSR_HPMCOUNTER28H 0xc9c
```

26.26.1.74 CSR_HPMCOUNTER29

```
#define CSR_HPMCOUNTER29 0xc1d
```

26.26.1.75 CSR_HPMCOUNTER29H

```
#define CSR_HPMCOUNTER29H 0xc9d
```

26.26.1.76 CSR_HPMCOUNTER3

```
#define CSR_HPMCOUNTER3 0xc03
```

26.26.1.77 CSR_HPMCOUNTER30

```
#define CSR_HPMCOUNTER30 0xc1e
```

26.26.1.78 CSR_HPMCOUNTER30H

```
#define CSR_HPMCOUNTER30H 0xc9e
```

26.26.1.79 CSR_HPMCOUNTER31

```
#define CSR_HPMCOUNTER31 0xc1f
```

26.26.1.80 CSR_HPMCOUNTER31H

```
#define CSR_HPMCOUNTER31H 0xc9f
```

26.26.1.81 CSR_HPMCOUNTER3H

```
#define CSR_HPMCOUNTER3H 0xc83
```

26.26.1.82 CSR_HPMCOUNTER4

```
#define CSR_HPMCOUNTER4 0xc04
```

26.26.1.83 CSR_HPMCOUNTER4H

```
#define CSR_HPMCOUNTER4H 0xc84
```

26.26.1.84 CSR_HPMCOUNTER5

```
#define CSR_HPMCOUNTER5 0xc05
```

26.26.1.85 CSR_HPMCOUNTER5H

```
#define CSR_HPMCOUNTER5H 0xc85
```

26.26.1.86 CSR_HPMCOUNTER6

```
#define CSR_HPMCOUNTER6 0xc06
```

26.26.1.87 CSR_HPMCOUNTER6H

```
#define CSR_HPMCOUNTER6H 0xc86
```

26.26.1.88 CSR_HPMCOUNTER7

```
#define CSR_HPMCOUNTER7 0xc07
```

26.26.1.89 CSR_HPMCOUNTER7H

```
#define CSR_HPMCOUNTER7H 0xc87
```

26.26.1.90 CSR_HPMCOUNTER8

```
#define CSR_HPMCOUNTER8 0xc08
```

26.26.1.91 CSR_HPMCOUNTER8H

```
#define CSR_HPMCOUNTER8H 0xc88
```

26.26.1.92 CSR_HPMCOUNTER9

```
#define CSR_HPMCOUNTER9 0xc09
```

26.26.1.93 CSR_HPMCOUNTER9H

```
#define CSR_HPMCOUNTER9H 0xc89
```

26.26.1.94 CSR_HSTATUS

```
#define CSR_HSTATUS 0x600
```

26.26.1.95 CSR_HTIMEDELTA

```
#define CSR_HTIMEDELTA 0x605
```

26.26.1.96 CSR_HTIMEDELTAH

```
#define CSR_HTIMEDELTAH 0x615
```

26.26.1.97 CSR_HTINST

```
#define CSR_HTINST 0x64a
```

26.26.1.98 CSR_HTVAL

```
#define CSR_HTVAL 0x643
```

26.26.1.99 CSR_INSTRET

```
#define CSR_INSTRET 0xc02
```

26.26.1.100 CSR_INSTRETH

```
#define CSR_INSTRETH 0xc82
```

26.26.1.101 CSR_MARCHID

```
#define CSR_MARCHID 0xf12
```

26.26.1.102 CSR_MCAUSE

```
#define CSR_MCAUSE 0x342
```

26.26.1.103 CSR_MCOUNTEREN

```
#define CSR_MCOUNTEREN 0x306
```

26.26.1.104 CSR_MCYCLE

```
#define CSR_MCYCLE 0xb00
```

26.26.1.105 CSR_MCYCLEH

```
#define CSR_MCYCLEH 0xb80
```

26.26.1.106 CSR_MEDELEG

```
#define CSR_MEDELEG 0x302
```

26.26.1.107 CSR_MEPC

```
#define CSR_MEPC 0x341
```

26.26.1.108 CSR_MHARTID

```
#define CSR_MHARTID 0xf14
```

26.26.1.109 CSR_MHPMCOUNTER10

```
#define CSR_MHPMCOUNTER10 0xb0a
```

26.26.1.110 CSR_MHPMCOUNTER10H

```
#define CSR_MHPMCOUNTER10H 0xb8a
```

26.26.1.111 CSR_MHPMCOUNTER11

```
#define CSR_MHPMCOUNTER11 0xb0b
```

26.26.1.112 CSR_MHPMCOUNTER11H

```
#define CSR_MHPMCOUNTER11H 0xb8b
```

26.26.1.113 CSR_MHPMCOUNTER12

```
#define CSR_MHPMCOUNTER12 0xb0c
```

26.26.1.114 CSR_MHPMCOUNTER12H

```
#define CSR_MHPMCOUNTER12H 0xb8c
```

26.26.1.115 CSR_MHPMCOUNTER13

```
#define CSR_MHPMCOUNTER13 0xb0d
```

26.26.1.116 CSR_MHPMCOUNTER13H

```
#define CSR_MHPMCOUNTER13H 0xb8d
```

26.26.1.117 CSR_MHPMCOUNTER14

```
#define CSR_MHPMCOUNTER14 0xb0e
```

26.26.1.118 CSR_MHPMCOUNTER14H

```
#define CSR_MHPMCOUNTER14H 0xb8e
```

26.26.1.119 CSR_MHPMCOUNTER15

```
#define CSR_MHPMCOUNTER15 0xb0f
```

26.26.1.120 CSR_MHPMCOUNTER15H

```
#define CSR_MHPMCOUNTER15H 0xb8f
```

26.26.1.121 CSR_MHPMCOUNTER16

```
#define CSR_MHPMCOUNTER16 0xb10
```

26.26.1.122 CSR_MHPMCOUNTER16H

```
#define CSR_MHPMCOUNTER16H 0xb90
```

26.26.1.123 CSR_MHPMCOUNTER17

```
#define CSR_MHPMCOUNTER17 0xb11
```

26.26.1.124 CSR_MHPMCOUNTER17H

```
#define CSR_MHPMCOUNTER17H 0xb91
```

26.26.1.125 CSR_MHPMCOUNTER18

```
#define CSR_MHPMCOUNTER18 0xb12
```

26.26.1.126 CSR_MHPMCOUNTER18H

```
#define CSR_MHPMCOUNTER18H 0xb92
```

26.26.1.127 CSR_MHPMCOUNTER19

```
#define CSR_MHPMCOUNTER19 0xb13
```

26.26.1.128 CSR_MHPMCOUNTER19H

```
#define CSR_MHPMCOUNTER19H 0xb93
```

26.26.1.129 CSR_MHPMCOUNTER20

```
#define CSR_MHPMCOUNTER20 0xb14
```

26.26.1.130 CSR_MHPMCOUNTER20H

```
#define CSR_MHPMCOUNTER20H 0xb94
```

26.26.1.131 CSR_MHPMCOUNTER21

```
#define CSR_MHPMCOUNTER21 0xb15
```

26.26.1.132 CSR_MHPMCOUNTER21H

```
#define CSR_MHPMCOUNTER21H 0xb95
```

26.26.1.133 CSR_MHPMCOUNTER22

```
#define CSR_MHPMCOUNTER22 0xb16
```

26.26.1.134 CSR_MHPMCOUNTER22H

```
#define CSR_MHPMCOUNTER22H 0xb96
```

26.26.1.135 CSR_MHPMCOUNTER23

```
#define CSR_MHPMCOUNTER23 0xb17
```

26.26.1.136 CSR_MHPMCOUNTER23H

```
#define CSR_MHPMCOUNTER23H 0xb97
```

26.26.1.137 CSR_MHPMCOUNTER24

```
#define CSR_MHPMCOUNTER24 0xb18
```

26.26.1.138 CSR_MHPMCOUNTER24H

```
#define CSR_MHPMCOUNTER24H 0xb98
```

26.26.1.139 CSR_MHPMCOUNTER25

```
#define CSR_MHPMCOUNTER25 0xb19
```

26.26.1.140 CSR_MHPMCOUNTER25H

```
#define CSR_MHPMCOUNTER25H 0xb99
```

26.26.1.141 CSR_MHPMCOUNTER26

```
#define CSR_MHPMCOUNTER26 0xb1a
```

26.26.1.142 CSR_MHPMCOUNTER26H

```
#define CSR_MHPMCOUNTER26H 0xb9a
```

26.26.1.143 CSR_MHPMCOUNTER27

```
#define CSR_MHPMCOUNTER27 0xb1b
```

26.26.1.144 CSR_MHPMCOUNTER27H

```
#define CSR_MHPMCOUNTER27H 0xb9b
```

26.26.1.145 CSR_MHPMCOUNTER28

```
#define CSR_MHPMCOUNTER28 0xb1c
```

26.26.1.146 CSR_MHPMCOUNTER28H

```
#define CSR_MHPMCOUNTER28H 0xb9c
```

26.26.1.147 CSR_MHPMCOUNTER29

```
#define CSR_MHPMCOUNTER29 0xb1d
```

26.26.1.148 CSR_MHPMCOUNTER29H

```
#define CSR_MHPMCOUNTER29H 0xb9d
```

26.26.1.149 CSR_MHPMCOUNTER3

```
#define CSR_MHPMCOUNTER3 0xb03
```

26.26.1.150 CSR_MHPMCOUNTER30

```
#define CSR_MHPMCOUNTER30 0xb1e
```

26.26.1.151 CSR_MHPMCOUNTER30H

```
#define CSR_MHPMCOUNTER30H 0xb9e
```

26.26.1.152 CSR_MHPMCOUNTER31

```
#define CSR_MHPMCOUNTER31 0xb1f
```

26.26.1.153 CSR_MHPMCOUNTER31H

```
#define CSR_MHPMCOUNTER31H 0xb9f
```

26.26.1.154 CSR_MHPMCOUNTER3H

```
#define CSR_MHPMCOUNTER3H 0xb83
```

26.26.1.155 CSR_MHPMCOUNTER4

```
#define CSR_MHPMCOUNTER4 0xb04
```

26.26.1.156 CSR_MHPMCOUNTER4H

```
#define CSR_MHPMCOUNTER4H 0xb84
```

26.26.1.157 CSR_MHPMCOUNTER5

```
#define CSR_MHPMCOUNTER5 0xb05
```

26.26.1.158 CSR_MHPMCOUNTER5H

```
#define CSR_MHPMCOUNTER5H 0xb85
```

26.26.1.159 CSR_MHPMCOUNTER6

```
#define CSR_MHPMCOUNTER6 0xb06
```

26.26.1.160 CSR_MHPMCOUNTER6H

```
#define CSR_MHPMCOUNTER6H 0xb86
```

26.26.1.161 CSR_MHPMCOUNTER7

```
#define CSR_MHPMCOUNTER7 0xb07
```

26.26.1.162 CSR_MHPMCOUNTER7H

```
#define CSR_MHPMCOUNTER7H 0xb87
```

26.26.1.163 CSR_MHPMCOUNTER8

```
#define CSR_MHPMCOUNTER8 0xb08
```

26.26.1.164 CSR_MHPMCOUNTER8H

```
#define CSR_MHPMCOUNTER8H 0xb88
```

26.26.1.165 CSR_MHPMCOUNTER9

```
#define CSR_MHPMCOUNTER9 0xb09
```

26.26.1.166 CSR_MHPMCOUNTER9H

```
#define CSR_MHPMCOUNTER9H 0xb89
```

26.26.1.167 CSR_MHPMEVENT10

```
#define CSR_MHPMEVENT10 0x32a
```

26.26.1.168 CSR_MHPMEVENT11

```
#define CSR_MHPMEVENT11 0x32b
```

26.26.1.169 CSR_MHPMEVENT12

```
#define CSR_MHPMEVENT12 0x32c
```

26.26.1.170 CSR_MHPMEVENT13

```
#define CSR_MHPMEVENT13 0x32d
```

26.26.1.171 CSR_MHPMEVENT14

```
#define CSR_MHPMEVENT14 0x32e
```

26.26.1.172 CSR_MHPMEVENT15

```
#define CSR_MHPMEVENT15 0x32f
```

26.26.1.173 CSR_MHPMEVENT16

```
#define CSR_MHPMEVENT16 0x330
```

26.26.1.174 CSR_MHPMEVENT17

```
#define CSR_MHPMEVENT17 0x331
```

26.26.1.175 CSR_MHPMEVENT18

```
#define CSR_MHPMEVENT18 0x332
```

26.26.1.176 CSR_MHPMEVENT19

```
#define CSR_MHPMEVENT19 0x333
```

26.26.1.177 CSR_MHPMEVENT20

```
#define CSR_MHPMEVENT20 0x334
```

26.26.1.178 CSR_MHPMEVENT21

```
#define CSR_MHPMEVENT21 0x335
```

26.26.1.179 CSR_MHPMEVENT22

```
#define CSR_MHPMEVENT22 0x336
```

26.26.1.180 CSR_MHPMEVENT23

```
#define CSR_MHPMEVENT23 0x337
```

26.26.1.181 CSR_MHPMEVENT24

```
#define CSR_MHPMEVENT24 0x338
```

26.26.1.182 CSR_MHPMEVENT25

```
#define CSR_MHPMEVENT25 0x339
```

26.26.1.183 CSR_MHPMEVENT26

```
#define CSR_MHPMEVENT26 0x33a
```

26.26.1.184 CSR_MHPMEVENT27

```
#define CSR_MHPMEVENT27 0x33b
```

26.26.1.185 CSR_MHPMEVENT28

```
#define CSR_MHPMEVENT28 0x33c
```

26.26.1.186 CSR_MHPMEVENT29

```
#define CSR_MHPMEVENT29 0x33d
```

26.26.1.187 CSR_MHPMEVENT3

```
#define CSR_MHPMEVENT3 0x323
```

26.26.1.188 CSR_MHPMEVENT30

```
#define CSR_MHPMEVENT30 0x33e
```

26.26.1.189 CSR_MHPMEVENT31

```
#define CSR_MHPMEVENT31 0x33f
```

26.26.1.190 CSR_MHPMEVENT4

```
#define CSR_MHPMEVENT4 0x324
```

26.26.1.191 CSR_MHPMEVENT5

```
#define CSR_MHPMEVENT5 0x325
```

26.26.1.192 CSR_MHPMEVENT6

```
#define CSR_MHPMEVENT6 0x326
```

26.26.1.193 CSR_MHPMEVENT7

```
#define CSR_MHPMEVENT7 0x327
```

26.26.1.194 CSR_MHPMEVENT8

```
#define CSR_MHPMEVENT8 0x328
```

26.26.1.195 CSR_MHPMEVENT9

```
#define CSR_MHPMEVENT9 0x329
```

26.26.1.196 CSR_MIDELEG

```
#define CSR_MIDELEG 0x303
```

26.26.1.197 CSR_MIE

```
#define CSR_MIE 0x304
```

26.26.1.198 CSR_MIMPID

```
#define CSR_MIMPID 0xf13
```

26.26.1.199 CSR_MINSTRET

```
#define CSR_MINSTRET 0xb02
```

26.26.1.200 CSR_MINSTRETH

```
#define CSR_MINSTRETH 0xb82
```

26.26.1.201 CSR_MIP

```
#define CSR_MIP 0x344
```

26.26.1.202 CSR_MISA

```
#define CSR_MISA 0x301
```

26.26.1.203 CSR_MSCRATCH

```
#define CSR_MSCRATCH 0x340
```

26.26.1.204 CSR_MSTATUS

```
#define CSR_MSTATUS 0x300
```

26.26.1.205 CSR_MSTATUSH

```
#define CSR_MSTATUSH 0x310
```

26.26.1.206 CSR_MTINST

```
#define CSR_MTINST 0x34a
```

26.26.1.207 CSR_MTVAL

```
#define CSR_MTVAL 0x343
```

26.26.1.208 CSR_MTVAL2

```
#define CSR_MTVAL2 0x34b
```

26.26.1.209 CSR_MTVEC

```
#define CSR_MTVEC 0x305
```

26.26.1.210 CSR_MVENDORID

```
#define CSR_MVENDORID 0xf11
```

26.26.1.211 CSR_PMPADDR0

```
#define CSR_PMPADDR0 0x3b0
```

26.26.1.212 CSR_PMPADDR1

```
#define CSR_PMPADDR1 0x3b1
```

26.26.1.213 CSR_PMPADDR10

```
#define CSR_PMPADDR10 0x3ba
```

26.26.1.214 CSR_PMPADDR11

```
#define CSR_PMPADDR11 0x3bb
```

26.26.1.215 CSR_PMPADDR12

```
#define CSR_PMPADDR12 0x3bc
```

26.26.1.216 CSR_PMPADDR13

```
#define CSR_PMPADDR13 0x3bd
```

26.26.1.217 CSR_PMPADDR14

```
#define CSR_PMPADDR14 0x3be
```

26.26.1.218 CSR_PMPADDR15

```
#define CSR_PMPADDR15 0x3bf
```

26.26.1.219 CSR_PMPADDR2

```
#define CSR_PMPADDR2 0x3b2
```

26.26.1.220 CSR_PMPADDR3

```
#define CSR_PMPADDR3 0x3b3
```

26.26.1.221 CSR_PMPADDR4

```
#define CSR_PMPADDR4 0x3b4
```

26.26.1.222 CSR_PMPADDR5

```
#define CSR_PMPADDR5 0x3b5
```

26.26.1.223 CSR_PMPADDR6

```
#define CSR_PMPADDR6 0x3b6
```

26.26.1.224 CSR_PMPADDR7

```
#define CSR_PMPADDR7 0x3b7
```

26.26.1.225 CSR_PMPADDR8

```
#define CSR_PMPADDR8 0x3b8
```

26.26.1.226 CSR_PMPADDR9

```
#define CSR_PMPADDR9 0x3b9
```

26.26.1.227 CSR_PMPCFG0

```
#define CSR_PMPCFG0 0x3a0
```

26.26.1.228 CSR_PMPCFG1

```
#define CSR_PMPCFG1 0x3a1
```

26.26.1.229 CSR_PMPCFG2

```
#define CSR_PMPCFG2 0x3a2
```

26.26.1.230 CSR_PMPCFG3

```
#define CSR_PMPCFG3 0x3a3
```

26.26.1.231 CSR_SATP

```
#define CSR_SATP 0x180
```

26.26.1.232 CSR_SCAUSE

```
#define CSR_SCAUSE 0x142
```

26.26.1.233 CSR_SCOUNTEREN

```
#define CSR_SCOUNTEREN 0x106
```

26.26.1.234 CSR_SEPC

```
#define CSR_SEPC 0x141
```

26.26.1.235 CSR_SIE

```
#define CSR_SIE 0x104
```

26.26.1.236 CSR_SIP

```
#define CSR_SIP 0x144
```

26.26.1.237 CSR_SSCRATCH

```
#define CSR_SSCRATCH 0x140
```

26.26.1.238 CSR_SSTATUS

```
#define CSR_SSTATUS 0x100
```

26.26.1.239 CSR_STVAL

```
#define CSR_STVAL 0x143
```

26.26.1.240 CSR_STVEC

```
#define CSR_STVEC 0x105
```

26.26.1.241 CSR_TDATA1

```
#define CSR_TDATA1 0x7a1
```

26.26.1.242 CSR_TDATA2

```
#define CSR_TDATA2 0x7a2
```

26.26.1.243 CSR_TDATA3

```
#define CSR_TDATA3 0x7a3
```

26.26.1.244 CSR_TIME

```
#define CSR_TIME 0xc01
```

26.26.1.245 CSR_TIMEH

```
#define CSR_TIMEH 0xc81
```

26.26.1.246 CSR_TSELECT

```
#define CSR_TSELECT 0x7a0
```

26.26.1.247 CSR_UCAUSE

```
#define CSR_UCAUSE 0x42
```

26.26.1.248 CSR_UEPC

```
#define CSR_UEPC 0x41
```

26.26.1.249 CSR_UIE

```
#define CSR_UIE 0x4
```

26.26.1.250 CSR_UIP

```
#define CSR_UIP 0x44
```

26.26.1.251 CSR_USCRATCH

```
#define CSR_USCRATCH 0x40
```

26.26.1.252 CSR_USTATUS

```
#define CSR_USTATUS 0x0
```

26.26.1.253 CSR_UTVAL

```
#define CSR_UTVAL 0x43
```

26.26.1.254 CSR_UTVEC

```
#define CSR_UTVEC 0x5
```

26.26.1.255 CSR_VSATP

```
#define CSR_VSATP 0x280
```

26.26.1.256 CSR_VSCAUSE

```
#define CSR_VSCAUSE 0x242
```

26.26.1.257 CSR_VSEPC

```
#define CSR_VSEPC 0x241
```

26.26.1.258 CSR_VSIE

```
#define CSR_VSIE 0x204
```

26.26.1.259 CSR_VSIP

```
#define CSR_VSIP 0x244
```

26.26.1.260 CSR_VSSCRATCH

```
#define CSR_VSSCRATCH 0x240
```

26.26.1.261 CSR_VSSTATUS

```
#define CSR_VSSTATUS 0x200
```

26.26.1.262 CSR_VSTVAL

```
#define CSR_VSTVAL 0x243
```

26.26.1.263 CSR_VSTVEC

```
#define CSR_VSTVEC 0x205
```

26.26.1.264 GET_RM

```
#define GET_RM(
            insn ) (((insn) >> 12) & 7)
```

26.26.1.265 GET_RS1

```
#define GET_RS1(
            insn,
            regs ) (*REG_PTR(insn, SH_RS1, regs))
```

26.26.1.266 GET_RS1S

```
#define GET_RS1S(
            insn,
            regs ) (*REG_PTR(RVC_RS1S(insn), 0, regs))
```

26.26.1.267 GET_RS2

```
#define GET_RS2(
            insn,
            regs ) (*REG_PTR(insn, SH_RS2, regs))
```

26.26.1.268 GET_RS2C

```
#define GET_RS2C(
            insn,
            regs ) (*REG_PTR(insn, SH_RS2C, regs))
```

26.26.1.269 GET_RS2S

```
#define GET_RS2S(
            insn,
            regs ) (*REG_PTR(RVC_RS2S(insn), 0, regs))
```

26.26.1.270 GET_SP

```
#define GET_SP(
            regs ) (*REG_PTR(2, 0, regs))
```

26.26.1.271 HGATP32_MODE_SHIFT

```
#define HGATP32_MODE_SHIFT 31
```

26.26.1.272 HGATP32_PPN

```
#define HGATP32_PPN _UL(0x003FFFFF)
```

26.26.1.273 HGATP32_VMID_MASK

```
#define HGATP32_VMID_MASK _UL(0x1FC00000)
```

26.26.1.274 HGATP32_VMID_SHIFT

```
#define HGATP32_VMID_SHIFT 22
```

26.26.1.275 HGATP64_MODE_SHIFT

```
#define HGATP64_MODE_SHIFT 60
```

26.26.1.276 HGATP64_PPN

```
#define HGATP64_PPN _ULL(0x00000FFFFFFFFFFF)
```

26.26.1.277 HGATP64_VMID_MASK

```
#define HGATP64_VMID_MASK _ULL(0x03FFF00000000000)
```

26.26.1.278 HGATP64_VMID_SHIFT

```
#define HGATP64_VMID_SHIFT 44
```

26.26.1.279 HGATP_MODE_OFF

```
#define HGATP_MODE_OFF _UL(0)
```

26.26.1.280 HGATP_MODE_SHIFT

```
#define HGATP_MODE_SHIFT HGATP32_MODE_SHIFT
```

26.26.1.281 HGATP_MODE_SV32X4

```
#define HGATP_MODE_SV32X4 _UL(1)
```

26.26.1.282 HGATP_MODE_SV39X4

```
#define HGATP_MODE_SV39X4 _UL(8)
```

26.26.1.283 HGATP_MODE_SV48X4

```
#define HGATP_MODE_SV48X4 _UL(9)
```

26.26.1.284 HGATP_PPN

```
#define HGATP_PPN HGATP32_PPN
```

26.26.1.285 HGATP_VMID_MASK

```
#define HGATP_VMID_MASK HGATP32_VMID_MASK
```

26.26.1.286 HGATP_VMID_SHIFT

```
#define HGATP_VMID_SHIFT HGATP32_VMID_SHIFT
```

26.26.1.287 HSTATUS_GVA

```
#define HSTATUS_GVA _UL(0x00000040)
```

26.26.1.288 HSTATUS_HU

```
#define HSTATUS_HU _UL(0x00000200)
```

26.26.1.289 HSTATUS_SPV

```
#define HSTATUS_SPV _UL(0x00000080)
```

26.26.1.290 HSTATUS_SPVP

```
#define HSTATUS_SPVP _UL(0x00000100)
```

26.26.1.291 HSTATUS_VGEIN

```
#define HSTATUS_VGEIN _UL(0x0003f000)
```

26.26.1.292 HSTATUS_VGEIN_SHIFT

```
#define HSTATUS_VGEIN_SHIFT 12
```

26.26.1.293 HSTATUS_VSBE

```
#define HSTATUS_VSBE _UL(0x00000020)
```

26.26.1.294 HSTATUS_VTSR

```
#define HSTATUS_VTSR _UL(0x00400000)
```

26.26.1.295 HSTATUS_VTVM

```
#define HSTATUS_VTVM _UL(0x00100000)
```

26.26.1.296 HSTATUS_VTW

```
#define HSTATUS_VTW _UL(0x00200000)
```

26.26.1.297 IMM_I

```
#define IMM_I(
            insn ) ((s32)(insn) >> 20)
```

26.26.1.298 IMM_S

```
#define IMM_S(
            insn )
```

Value:

```
(((s32)(insn) » 25 « 5) | \
(s32)(((insn) » 7) & 0x1f))
```

26.26.1.299 INSN_16BIT_MASK

```
#define INSN_16BIT_MASK 0x3
```

26.26.1.300 INSN_32BIT_MASK

```
#define INSN_32BIT_MASK 0x1c
```

26.26.1.301 INSN_IS_16BIT

```
#define INSN_IS_16BIT(
            insn ) (((insn) & INSN_16BIT_MASK) != INSN_16BIT_MASK)
```

26.26.1.302 INSN_IS_32BIT

```
#define INSN_IS_32BIT(
              insn )
```

Value:

```
    (((insn) & INSN_16BIT_MASK) == INSN_16BIT_MASK && \
     ((insn) & INSN_32BIT_MASK) != INSN_32BIT_MASK)
```

26.26.1.303 INSN_LEN

```
#define INSN_LEN(
              insn ) (INSN_IS_16BIT(insn) ?  2 :  4)
```

26.26.1.304 INSN_MASK_C_FLD

```
#define INSN_MASK_C_FLD 0xe003
```

26.26.1.305 INSN_MASK_C_FLDSP

```
#define INSN_MASK_C_FLDSP 0xe003
```

26.26.1.306 INSN_MASK_C_FLW

```
#define INSN_MASK_C_FLW 0xe003
```

26.26.1.307 INSN_MASK_C_FLWSP

```
#define INSN_MASK_C_FLWSP 0xe003
```

26.26.1.308 INSN_MASK_C_FSD

```
#define INSN_MASK_C_FSD 0xe003
```

26.26.1.309 INSN_MASK_C_FSDSP

```
#define INSN_MASK_C_FSDSP 0xe003
```

26.26.1.310 INSN_MASK_C_FSW

```
#define INSN_MASK_C_FSW 0xe003
```

26.26.1.311 INSN_MASK_C_FSWSP

```
#define INSN_MASK_C_FSWSP 0xe003
```

26.26.1.312 INSN_MASK_C_LD

```
#define INSN_MASK_C_LD 0xe003
```

26.26.1.313 INSN_MASK_C_LDSP

```
#define INSN_MASK_C_LDSP 0xe003
```

26.26.1.314 INSN_MASK_C_LW

```
#define INSN_MASK_C_LW 0xe003
```

26.26.1.315 INSN_MASK_C_LWSP

```
#define INSN_MASK_C_LWSP 0xe003
```

26.26.1.316 INSN_MASK_C_SD

```
#define INSN_MASK_C_SD 0xe003
```

26.26.1.317 INSN_MASK_C_SDSP

```
#define INSN_MASK_C_SDSP 0xe003
```

26.26.1.318 INSN_MASK_C_SW

```
#define INSN_MASK_C_SW 0xe003
```

26.26.1.319 INSN_MASK_C_SWSP

```
#define INSN_MASK_C_SWSP 0xe003
```

26.26.1.320 INSN_MASK_FLD

```
#define INSN_MASK_FLD 0x707f
```

26.26.1.321 INSN_MASK_FLQ

```
#define INSN_MASK_FLQ 0x707f
```

26.26.1.322 INSN_MASK_FLW

```
#define INSN_MASK_FLW 0x707f
```

26.26.1.323 INSN_MASK_FSD

```
#define INSN_MASK_FSD 0x707f
```

26.26.1.324 INSN_MASK_FSQ

```
#define INSN_MASK_FSQ 0x707f
```

26.26.1.325 INSN_MASK_FSW

```
#define INSN_MASK_FSW 0x707f
```

26.26.1.326 INSN_MASK_LB

```
#define INSN_MASK_LB 0x707f
```

26.26.1.327 INSN_MASK_LBU

```
#define INSN_MASK_LBU 0x707f
```

26.26.1.328 INSN_MASK_LD

```
#define INSN_MASK_LD 0x707f
```

26.26.1.329 INSN_MASK_LH

```
#define INSN_MASK_LH 0x707f
```

26.26.1.330 INSN_MASK_LHU

```
#define INSN_MASK_LHU 0x707f
```

26.26.1.331 INSN_MASK_LW

```
#define INSN_MASK_LW 0x707f
```

26.26.1.332 INSN_MASK_LWU

```
#define INSN_MASK_LWU 0x707f
```

26.26.1.333 INSN_MASK_SB

```
#define INSN_MASK_SB 0x707f
```

26.26.1.334 INSN_MASK_SD

```
#define INSN_MASK_SD 0x707f
```

26.26.1.335 INSN_MASK_SH

```
#define INSN_MASK_SH 0x707f
```

26.26.1.336 INSN_MASK_SW

```
#define INSN_MASK_SW 0x707f
```

26.26.1.337 INSN_MASK_WFI

```
#define INSN_MASK_WFI 0xffffff00
```

26.26.1.338 INSN_MATCH_C_FLD

```
#define INSN_MATCH_C_FLD 0x2000
```

26.26.1.339 INSN_MATCH_C_FLDSP

```
#define INSN_MATCH_C_FLDSP 0x2002
```

26.26.1.340 INSN_MATCH_C_FLW

```
#define INSN_MATCH_C_FLW 0x6000
```

26.26.1.341 INSN_MATCH_C_FLWSP

```
#define INSN_MATCH_C_FLWSP 0x6002
```

26.26.1.342 INSN_MATCH_C_FSD

```
#define INSN_MATCH_C_FSD 0xa000
```

26.26.1.343 INSN_MATCH_C_FSDSP

```
#define INSN_MATCH_C_FSDSP 0xa002
```

26.26.1.344 INSN_MATCH_C_FSW

```
#define INSN_MATCH_C_FSW 0xe000
```

26.26.1.345 INSN_MATCH_C_FSWSP

```
#define INSN_MATCH_C_FSWSP 0xe002
```

26.26.1.346 INSN_MATCH_C_LD

```
#define INSN_MATCH_C_LD 0x6000
```

26.26.1.347 INSN_MATCH_C_LDSP

```
#define INSN_MATCH_C_LDSP 0x6002
```

26.26.1.348 INSN_MATCH_C_LW

```
#define INSN_MATCH_C_LW 0x4000
```

26.26.1.349 INSN_MATCH_C_LWSP

```
#define INSN_MATCH_C_LWSP 0x4002
```

26.26.1.350 INSN_MATCH_C_SD

```
#define INSN_MATCH_C_SD 0xe000
```

26.26.1.351 INSN_MATCH_C_SDSP

```
#define INSN_MATCH_C_SDSP 0xe002
```

26.26.1.352 INSN_MATCH_C_SW

```
#define INSN_MATCH_C_SW 0xc000
```

26.26.1.353 INSN_MATCH_C_SWSP

```
#define INSN_MATCH_C_SWSP 0xc002
```

26.26.1.354 INSN_MATCH_FLD

```
#define INSN_MATCH_FLD 0x3007
```

26.26.1.355 INSN_MATCH_FLQ

```
#define INSN_MATCH_FLQ 0x4007
```

26.26.1.356 INSN_MATCH_FLW

```
#define INSN_MATCH_FLW 0x2007
```

26.26.1.357 INSN_MATCH_FSD

```
#define INSN_MATCH_FSD 0x3027
```

26.26.1.358 INSN_MATCH_FSQ

```
#define INSN_MATCH_FSQ 0x4027
```

26.26.1.359 INSN_MATCH_FSW

```
#define INSN_MATCH_FSW 0x2027
```

26.26.1.360 INSN_MATCH_LB

```
#define INSN_MATCH_LB 0x3
```

26.26.1.361 INSN_MATCH_LBU

```
#define INSN_MATCH_LBU 0x4003
```

26.26.1.362 INSN_MATCH_LD

```
#define INSN_MATCH_LD 0x3003
```

26.26.1.363 INSN_MATCH_LH

```
#define INSN_MATCH_LH 0x1003
```

26.26.1.364 INSN_MATCH_LHU

```
#define INSN_MATCH_LHU 0x5003
```

26.26.1.365 INSN_MATCH_LW

```
#define INSN_MATCH_LW 0x2003
```

26.26.1.366 INSN_MATCH_LWU

```
#define INSN_MATCH_LWU 0x6003
```

26.26.1.367 INSN_MATCH_SB

```
#define INSN_MATCH_SB 0x23
```

26.26.1.368 INSN_MATCH_SD

```
#define INSN_MATCH_SD 0x3023
```

26.26.1.369 INSN_MATCH_SH

```
#define INSN_MATCH_SH 0x1023
```

26.26.1.370 INSN_MATCH_SW

```
#define INSN_MATCH_SW 0x2023
```

26.26.1.371 INSN_MATCH_WFI

```
#define INSN_MATCH_WFI 0x10500000
```

26.26.1.372 IRQ_M_EXT

```
#define IRQ_M_EXT 11
```

26.26.1.373 IRQ_M_SOFT

```
#define IRQ_M_SOFT 3
```

26.26.1.374 IRQ_M_TIMER

```
#define IRQ_M_TIMER 7
```

26.26.1.375 IRQ_S_EXT

```
#define IRQ_S_EXT 9
```

26.26.1.376 IRQ_S_GEXT

```
#define IRQ_S_GEXT 12
```

26.26.1.377 IRQ_S_SOFT

```
#define IRQ_S_SOFT 1
```

26.26.1.378 IRQ_S_TIMER

```
#define IRQ_S_TIMER 5
```

26.26.1.379 IRQ_VS_EXT

```
#define IRQ_VS_EXT 10
```

26.26.1.380 IRQ_VS_SOFT

```
#define IRQ_VS_SOFT 2
```

26.26.1.381 IRQ_VS_TIMER

```
#define IRQ_VS_TIMER 6
```

26.26.1.382 LOG_REGBYTES

```
#define LOG_REGBYTES 2
```

26.26.1.383 MASK_FUNCT3

```
#define MASK_FUNCT3 0x7000
```

26.26.1.384 MIP_MEIP

```
#define MIP_MEIP (_UL(1) << IRQ_M_EXT)
```

26.26.1.385 MIP_MSIP

```
#define MIP_MSIP (_UL(1) << IRQ_M_SOFT)
```

26.26.1.386 MIP_MTIP

```
#define MIP_MTIP (_UL(1) << IRQ_M_TIMER)
```

26.26.1.387 MIP_SEIP

```
#define MIP_SEIP (_UL(1) << IRQ_S_EXT)
```

26.26.1.388 MIP_SGEIP

```
#define MIP_SGEIP (_UL(1) << IRQ_S_GEXT)
```

26.26.1.389 MIP_SSIP

```
#define MIP_SSIP (_UL(1) << IRQ_S_SOFT)
```

26.26.1.390 MIP_STIP

```
#define MIP_STIP (_UL(1) << IRQ_S_TIMER)
```

26.26.1.391 MIP_VSEIP

```
#define MIP_VSEIP (_UL(1) << IRQ_VS_EXT)
```

26.26.1.392 MIP_VSSIP

```
#define MIP_VSSIP (_UL(1) << IRQ_VS_SOFT)
```

26.26.1.393 MIP_VSTIP

```
#define MIP_VSTIP (_UL(1) << IRQ_VS_TIMER)
```

26.26.1.394 MSTATUS32_SD [1/2]

```
#define MSTATUS32_SD _UL(0x80000000)
```

26.26.1.395 MSTATUS32_SD [2/2]

```
#define MSTATUS32_SD _UL(0x80000000)
```

26.26.1.396 MSTATUS64_SD

```
#define MSTATUS64_SD _ULL(0x8000000000000000)
```

26.26.1.397 MSTATUS_FS

```
#define MSTATUS_FS _UL(0x00006000)
```

26.26.1.398 MSTATUS_MIE

```
#define MSTATUS_MIE _UL(0x00000008)
```

26.26.1.399 MSTATUS_MPIE

```
#define MSTATUS_MPIE _UL(0x00000080)
```

26.26.1.400 MSTATUS_MPP

```
#define MSTATUS_MPP (_UL(3) << MSTATUS_MPP_SHIFT)
```

26.26.1.401 MSTATUS_MPP_SHIFT

```
#define MSTATUS_MPP_SHIFT 11
```

26.26.1.402 MSTATUS_MPRV

```
#define MSTATUS_MPRV _UL(0x00020000)
```

26.26.1.403 MSTATUS_MXR

```
#define MSTATUS_MXR _UL(0x00080000)
```

26.26.1.404 MSTATUS_SD

```
#define MSTATUS_SD MSTATUS32_SD
```

26.26.1.405 MSTATUS_SIE

```
#define MSTATUS_SIE _UL(0x00000002)
```

26.26.1.406 MSTATUS_SPIE

```
#define MSTATUS_SPIE (_UL(1) << MSTATUS_SPIE_SHIFT)
```

26.26.1.407 MSTATUS_SPIE_SHIFT

```
#define MSTATUS_SPIE_SHIFT 5
```

26.26.1.408 MSTATUS_SPP

```
#define MSTATUS_SPP (_UL(1) << MSTATUS_SPP_SHIFT)
```

26.26.1.409 MSTATUS_SPP_SHIFT

```
#define MSTATUS_SPP_SHIFT 8
```

26.26.1.410 MSTATUS_SUM

```
#define MSTATUS_SUM _UL(0x00040000)
```

26.26.1.411 MSTATUS_TSR

```
#define MSTATUS_TSR _UL(0x00400000)
```

26.26.1.412 MSTATUS_TVM

```
#define MSTATUS_TVM _UL(0x00100000)
```

26.26.1.413 MSTATUS_TW

```
#define MSTATUS_TW _UL(0x00200000)
```

26.26.1.414 MSTATUS_UBE

```
#define MSTATUS_UBE _UL(0x00000040)
```

26.26.1.415 MSTATUS_VS

```
#define MSTATUS_VS _UL(0x01800000)
```

26.26.1.416 MSTATUS_XS

```
#define MSTATUS_XS _UL(0x00018000)
```

26.26.1.417 MSTATUSH_MBE

```
#define MSTATUSH_MBE _UL(0x00000020)
```

26.26.1.418 MSTATUSH_MPV

```
#define MSTATUSH_MPV _UL(0x00000080)
```

26.26.1.419 MSTATUSH_SBE

```
#define MSTATUSH_SBE _UL(0x00000010)
```

26.26.1.420 PMP_A

```
#define PMP_A _UL(0x18)
```

26.26.1.421 PMP_A_NA4

```
#define PMP_A_NA4 _UL(0x10)
```

26.26.1.422 PMP_A_NAPOT

```
#define PMP_A_NAPOT _UL(0x18)
```

26.26.1.423 PMP_A_TOR

```
#define PMP_A_TOR _UL(0x08)
```

26.26.1.424 PMP_COUNT

```
#define PMP_COUNT 16
```

26.26.1.425 PMP_L

```
#define PMP_L _UL(0x80)
```

26.26.1.426 PMP_R

```
#define PMP_R _UL(0x01)
```

26.26.1.427 PMP_SHIFT

```
#define PMP_SHIFT 2
```

26.26.1.428 PMP_W

```
#define PMP_W _UL(0x02)
```

26.26.1.429 PMP_X

```
#define PMP_X _UL(0x04)
```

26.26.1.430 PRV_M

```
#define PRV_M _UL(3)
```

26.26.1.431 PRV_S

```
#define PRV_S _UL(1)
```

26.26.1.432 PRV_U

```
#define PRV_U _UL(0)
```

26.26.1.433 REG_MASK

```
#define REG_MASK ((1 << (5 + LOG_REGBYTES)) - (1 << LOG_REGBYTES))
```

26.26.1.434 REG_OFFSET

```
#define REG_OFFSET(
            insn,
            pos ) (SHIFT_RIGHT((insn), (pos) - LOG_REGBYTES) & REG_MASK)
```

26.26.1.435 REG_PTR

```
#define REG_PTR(
            insn,
            pos,
            regs ) (ulong *)((ulong)(regs) + REG_OFFSET(insn, pos))
```

26.26.1.436 REGBYTES

```
#define REGBYTES (1 << LOG_REGBYTES)
```

26.26.1.437 RV_X

```
#define RV_X(
            x,
            s,
            n ) (((x) >> (s)) & ((1 << (n)) - 1))
```

26.26.1.438 RVC_LD_IMM

```
#define RVC_LD_IMM(
            x )
```

Value:

```
((RV_X(x, 10, 3) « 3) | \
(RV_X(x, 5, 2) « 6))
```

26.26.1.439 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))
```

26.26.1.440 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))
```

26.26.1.441 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))
```

26.26.1.442 RVC_RS1S

```
#define RVC_RS1S(
            insn ) (8 + RV_X(insn, SH_RD, 3))
```

26.26.1.443 RVC_RS2

```
#define RVC_RS2(
            insn ) RV_X(insn, SH_RS2C, 5)
```

26.26.1.444 RVC_RS2S

```
#define RVC_RS2S(
            insn ) (8 + RV_X(insn, SH_RS2C, 3))
```

26.26.1.445 RVC_SDSP_IMM

```
#define RVC_SDSP_IMM(
            x )
```

Value:

```
((RV_X(x, 10, 3) « 3) | \
(RV_X(x, 7, 3) « 6))
```

26.26.1.446 RVC_SWSP_IMM

```
#define RVC_SWSP_IMM(
            x )
```

Value:

```
((RV_X(x, 9, 4) « 2) | \
(RV_X(x, 7, 2) « 6))
```

26.26.1.447 SATP32_ASID

```
#define SATP32_ASID _UL(0x7FC00000)
```

26.26.1.448 SATP32_MODE

```
#define SATP32_MODE _UL(0x80000000)
```

26.26.1.449 SATP32_PPN

```
#define SATP32_PPN _UL(0x003FFFFF)
```

26.26.1.450 SATP64_ASID

```
#define SATP64_ASID _ULL(0x0FFFF00000000000)
```

26.26.1.451 SATP64_MODE

```
#define SATP64_MODE _ULL(0xF000000000000000)
```

26.26.1.452 SATP64_PPN

```
#define SATP64_PPN _ULL(0x00000FFFFFFFFFFF)
```

26.26.1.453 SATP_MODE

```
#define SATP_MODE SATP32_MODE
```

26.26.1.454 SATP_MODE_OFF

```
#define SATP_MODE_OFF _UL(0)
```

26.26.1.455 SATP_MODE_SV32

```
#define SATP_MODE_SV32 _UL(1)
```

26.26.1.456 SATP_MODE_SV39

```
#define SATP_MODE_SV39 _UL(8)
```

26.26.1.457 SATP_MODE_SV48

```
#define SATP_MODE_SV48 _UL(9)
```

26.26.1.458 SATP_MODE_SV57

```
#define SATP_MODE_SV57 _UL(10)
```

26.26.1.459 SATP_MODE_SV64

```
#define SATP_MODE_SV64 _UL(11)
```

26.26.1.460 SET_RD

```
#define SET_RD(
              insn,
              regs,
              val ) (*REG_PTR(insn, SH_RD, regs) = (val))
```

26.26.1.461 SH_RD

```
#define SH_RD 7
```

26.26.1.462 SH_RS1

```
#define SH_RS1 15
```

26.26.1.463 SH_RS2

```
#define SH_RS2 20
```

26.26.1.464 SH_RS2C

```
#define SH_RS2C 2
```

26.26.1.465 SHIFT_RIGHT

```
#define SHIFT_RIGHT(
            x,
            y ) ((y) < 0 ? ((x) << -(y)) : ((x) >> (y)))
```

26.26.1.466 SIP_SSIP

```
#define SIP_SSIP MIP_SSIP
```

26.26.1.467 SIP_STIP

```
#define SIP_STIP MIP_STIP
```

26.26.1.468 SSTATUS32_SD

```
#define SSTATUS32_SD MSTATUS32_SD
```

26.26.1.469 SSTATUS64_SD

```
#define SSTATUS64_SD MSTATUS64_SD
```

26.26.1.470 SSTATUS64_UXL

```
#define SSTATUS64_UXL MSTATUS_UXL
```

26.26.1.471 SSTATUS_FS

```
#define SSTATUS_FS MSTATUS_FS
```

26.26.1.472 SSTATUS_MXR

```
#define SSTATUS_MXR MSTATUS_MXR
```

26.26.1.473 SSTATUS_SD

```
#define SSTATUS_SD SSTATUS32_SD
```

26.26.1.474 SSTATUS_SIE

```
#define SSTATUS_SIE MSTATUS_SIE
```

26.26.1.475 SSTATUS_SPIE

```
#define SSTATUS_SPIE MSTATUS_SPIE
```

26.26.1.476 SSTATUS_SPIE_SHIFT

```
#define SSTATUS_SPIE_SHIFT MSTATUS_SPIE_SHIFT
```

26.26.1.477 SSTATUS_SPP

```
#define SSTATUS_SPP MSTATUS_SPP
```

26.26.1.478 SSTATUS_SPP_SHIFT

```
#define SSTATUS_SPP_SHIFT MSTATUS_SPP_SHIFT
```

26.26.1.479 SSTATUS_SUM

```
#define SSTATUS_SUM MSTATUS_SUM
```

26.26.1.480 SSTATUS_VS

```
#define SSTATUS_VS MSTATUS_VS
```

26.26.1.481 SSTATUS_XS

```
#define SSTATUS_XS MSTATUS_XS
```

26.27 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 GET_RM(insn) (((insn) >> 12) & 7)
- #define PRECISION_S 0
- #define PRECISION_D 1

26.27.1 Macro Definition Documentation

26.27.1.1 GET_PRECISION

```
#define GET_PRECISION(
            insn ) (((insn) >> 25) & 3)
```

26.27.1.2 GET_RM

```
#define GET_RM(
              insn ) (((insn) >> 12) & 7)
```

26.27.1.3 PRECISION_D

```
#define PRECISION_D 1
```

26.27.1.4 PRECISION_S

```
#define PRECISION_S 0
```

26.28 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)

26.28.1 Macro Definition Documentation

26.28.1.1 __io_ar

```
#define __io_ar( ) __asm__ __volatile__ ("fence i,r" : : : "memory");
```

26.28.1.2 __io_aw

```
#define __io_aw( ) do {} while (0)
```

26.28.1.3 __io_br

```
#define __io_br( ) do {} while (0)
```

26.28.1.4 __io_bw

```
#define __io_bw( ) __asm__ __volatile__ ("fence w,o" : : : "memory");
```

26.28.1.5 __io_rar

```
#define __io_rar( ) do {} while (0)
```

26.28.1.6 __io_raw

```
#define __io_raw( ) do {} while (0)
```

26.28.1.7 __io_rbr

```
#define __io_rbr( ) do {} while (0)
```

26.28.1.8 __io_rbw

```
#define __io_rbw( ) do {} while (0)
```

26.28.1.9 readb

```
#define readb(
            c ) ({ u8 __v; __io_br(); __v = __raw_readb(c); __io_ar(); __v; })
```

26.28.1.10 readb_relaxed

```
#define readb_relaxed(
            c ) ({ u8 __v; __io_rbr(); __v = __raw_readb(c); __io_rar(); __v; })
```

26.28.1.11 readl

```
#define readl(
            c ) ({ u32 __v; __io_br(); __v = __raw_readl(c); __io_ar(); __v; })
```

26.28.1.12 readl_relaxed

```
#define readl_relaxed(
            c ) ({ u32 __v; __io_rbr(); __v = __raw_readl(c); __io_rar(); __v; })
```

26.28.1.13 readq

```
#define readq(
            c ) ({ u64 __v; __io_br(); __v = __raw_readq(c); __io_ar(); __v; })
```

26.28.1.14 readq_relaxed

```
#define readq_relaxed(
            c ) ({ u64 __v; __io_rbr(); __v = __raw_readq(c); __io_rar(); __v; })
```

26.28.1.15 readw

```
#define readw(
            c ) ({ u16 __v; __io_br(); __v = __raw_readw(c); __io_ar(); __v; })
```

26.28.1.16 readw_relaxed

```
#define readw_relaxed(
            c ) ({ u16 __v; __io_rbr(); __v = __raw_readw(c); __io_rar(); __v; })
```

26.28.1.17 writeb

```
#define writeb(
            v,
            c ) ({ __io_bw(); __raw_writeb((v),(c)); __io_aw(); })
```

26.28.1.18 writeb_relaxed

```
#define writeb_relaxed(
            v,
            c ) ({ __io_rbw(); __raw_writeb((v),(c)); __io_raw(); })
```

26.28.1.19 writel

```
#define writel(
            v,
            c ) ({ __io_bw(); __raw_writel((v),(c)); __io_aw(); })
```

26.28.1.20 writel_relaxed

```
#define writel_relaxed(
            v,
            c ) ({ __io_rbw(); __raw_writel((v),(c)); __io_raw(); })
```

26.28.1.21 writeq

```
#define writeq(
            v,
            c ) ({ __io_bw(); __raw_writeq((v),(c)); __io_aw(); })
```

26.28.1.22 writeq_relaxed

```
#define writeq_relaxed(
            v,
            c ) ({ __io_rbw(); __raw_writeq((v),(c)); __io_raw(); })
```

26.28.1.23 writew

```
#define writew(
            v,
            c ) ({ __io_bw(); __raw_writew((v),(c)); __io_aw(); })
```

26.28.1.24 writew_relaxed

```
#define writew_relaxed(
            v,
            c ) ({ __io_rbw(); __raw_writew((v),(c)); __io_raw(); })
```

26.28.2 Function Documentation

26.28.2.1 __raw_readb()

```
static u8 __raw_readb (
            const volatile void * addr )  [inline], [static]
```

26.28.2.2 __raw_readl()

```
static u32 __raw_readl (
            const volatile void * addr )  [inline], [static]
```

26.28.2.3 __raw_readq()

```
static u64 __raw_readq (
            const volatile void * addr )  [inline], [static]
```

26.28.2.4 __raw_readw()

```
static u16 __raw_readw (
            const volatile void * addr )  [inline], [static]
```

26.28.2.5 __raw_writeb()

```
static void __raw_writeb (
            u8 val,
            volatile void * addr )  [inline], [static]
```

26.28.2.6 __raw_writel()

```
static void __raw_writel (
            u32 val,
            volatile void * addr )  [inline], [static]
```

26.28.2.7 __raw_writeq()

```
static void __raw_writeq (
            u64 val,
            volatile void * addr )  [inline], [static]
```

26.28.2.8 __raw_writew()

```
static void __raw_writew (
            u16 val,
            volatile void * addr )  [inline], [static]
```

26.29 include/sbi/riscv_locks.h File Reference

This graph shows which files directly or indirectly include this file:

Data Structures

- struct spinlock_t

Macros

- #define __RISCV_SPIN_UNLOCKED 0
- #define SPIN_LOCK_INIT(_lptr) (_lptr)->lock = __RISCV_SPIN_UNLOCKED
- #define SPIN_LOCK_INITIALIZER

Functions

- int spin_lock_check (spinlock_t *lock)
- int spin_trylock (spinlock_t *lock)
- void spin_lock (spinlock_t *lock)
- void spin_unlock (spinlock_t *lock)

26.29.1 Macro Definition Documentation

26.29.1.1 __RISCV_SPIN_UNLOCKED

```
#define __RISCV_SPIN_UNLOCKED 0
```

26.29.1.2 SPIN_LOCK_INIT

```
#define SPIN_LOCK_INIT(
            _lptr ) (_lptr)->lock = __RISCV_SPIN_UNLOCKED
```

26.29.1.3 SPIN_LOCK_INITIALIZER

```
#define SPIN_LOCK_INITIALIZER
```

Value:

```
    {                                   \
        .lock = __RISCV_SPIN_UNLOCKED, \
    }
```

26.29.2 Function Documentation

26.29.2.1 spin_lock()

```
void spin_lock (
            spinlock_t * lock )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.29.2.2 spin_lock_check()

```
int spin_lock_check (
            spinlock_t * lock )
```

Here is the caller graph for this function:

26.29.2.3 spin_trylock()

```
int spin_trylock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

26.29.2.4 spin_unlock()

```
void spin_unlock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

26.30 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 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)

26.30.1 Macro Definition Documentation

26.30.1.1 BITMAP_FIRST_WORD_MASK

```
#define BITMAP_FIRST_WORD_MASK(
            start ) (~0UL << ((start) % BITS_PER_LONG))
```

26.30.1.2 BITMAP_LAST_WORD_MASK

```
#define BITMAP_LAST_WORD_MASK(
            nbits )
```

Value:

```
(                                   \
    ((nbits) % BITS_PER_LONG) ?                 \
        ((1UL « ((nbits) % BITS_PER_LONG)) - 1) : ~0UL      \
)
```

26.30.1.3 DECLARE_BITMAP

```
#define DECLARE_BITMAP(
              name,
              nbits ) unsigned long name[BITS_TO_LONGS(nbits)]
```

26.30.1.4 DEFINE_BITMAP

```
#define DEFINE_BITMAP(
              name ) extern unsigned long name[]
```

26.30.1.5 small_const_nbits

```
#define small_const_nbits(
              nbits ) (__builtin_constant_p(nbits) && (nbits) <= BITS_PER_LONG)
```

26.30.2 Function Documentation

26.30.2.1 __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:

26.30.2.2 __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:

26.30.2.3 __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:

26.30.2.4 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:

26.30.2.5 bitmap_clear()

```
static void bitmap_clear (
            unsigned long * bmap,
            int start,
            int len )  [inline], [static]
```

26.30.2.6 bitmap_copy()

```
static void bitmap_copy (
            unsigned long * dst,
            const unsigned long * src,
            int nbits )  [inline], [static]
```

26.30.2.7 bitmap_estimate_size()

```
static unsigned long bitmap_estimate_size (
            int nbits )  [inline], [static]
```

26.30.2.8 bitmap_fill()

```
static void bitmap_fill (
            unsigned long * dst,
            int nbits )  [inline], [static]
```

Here is the caller graph for this function:

26.30.2.9 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:

26.30.2.10 bitmap_set()

```
static void bitmap_set (
            unsigned long * bmap,
            int start,
            int len )  [inline], [static]
```

26.30.2.11 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:

26.30.2.12 bitmap_zero()

```
static void bitmap_zero (
            unsigned long * dst,
            int nbits )  [inline], [static]
```

Here is the caller graph for this function:

26.30.2.13 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:

26.31 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 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 GENMASK(h, l) (((~0UL) - (1UL << (l)) + 1) & (~0UL >> (BITS_PER_LONG - 1 - (h))))
- #define ffz(x) __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 ffs (int x)
- static int __ffs (unsigned long word)
- static int fls (int x)
- static unsigned long __fls (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)

26.31.1 Macro Definition Documentation

26.31.1.1 BIT

```
#define BIT(
            nr ) (1UL << (nr))
```

26.31.1.2 BIT_MASK

```
#define BIT_MASK(
            nr ) (1UL << ((nr) % BITS_PER_LONG))
```

26.31.1.3 BIT_WORD

```
#define BIT_WORD(
            bit ) ((bit) / BITS_PER_LONG)
```

26.31.1.4 BIT_WORD_OFFSET

```
#define BIT_WORD_OFFSET(
            bit ) ((bit) & (BITS_PER_LONG - 1))
```

26.31.1.5 BITS_TO_LONGS

```
#define BITS_TO_LONGS(
            nbits )
```

Value:

```
(((nbits) + BITS_PER_LONG - 1) / \
BITS_PER_LONG)
```

26.31.1.6 EXTRACT_FIELD

```
#define EXTRACT_FIELD(
            val,
            which ) (((val) & (which)) / ((which) & ~((which)-1)))
```

26.31.1.7 ffz

```
#define ffz(
            x ) __ffs(~(x))
```

26.31.1.8 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))
```

26.31.1.9 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))
```

26.31.1.10 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))
```

26.31.1.11 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))
```

26.31.1.12 GENMASK

```
#define GENMASK(
                h,
                l ) (((~0UL) - (1UL << (l)) + 1) & (~0UL >> (BITS_PER_LONG - 1 - (h))))
```

26.31.1.13 INSERT_FIELD

```
#define INSERT_FIELD(
                val,
                which,
                fieldval ) (((val) & ~(which)) | ((fieldval) * ((which) & ~((which)-1))))
```

26.31.2 Function Documentation

26.31.2.1 __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.

26.31.2.2 __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:

26.31.2.3 __ffs()

```
static int __ffs (
            unsigned long word )  [inline], [static]
```

__ffs - find first bit in 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:

26.31.2.4 __fls()

```
static unsigned long __fls (
            unsigned long word )  [inline], [static]
```

__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:

26.31.2.5 __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:

26.31.2.6 __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.

26.31.2.7 __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.

26.31.2.8 __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:

26.31.2.9 ffs()

```
static int ffs (
            int x )  [inline], [static]
```

ffs - Find first bit set @x: the word to search
This is defined the same way as the libc and compiler builtin ffs routines, therefore differs in spirit from the above ffz (man ffs).

26.31.2.10 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:

26.31.2.11 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.

26.31.2.12 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:

26.31.2.13 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:

26.31.2.14 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

26.31.2.15 fls()

```
static int fls (
            int x )  [inline], [static]
```

fls - find last (most-significant) bit set @x: the word to search
This is defined the same way as ffs. Note fls(0) = 0, fls(1) = 1, fls(0x80000000) = 32.

26.32 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:

Macros

- #define __printf(a, b) __attribute__((format(printf, a, b)))

Functions

- bool sbi_isprintable (char ch)
- int sbi_getc (void)
- void sbi_putc (char ch)
- void sbi_puts (const char *str)
- void sbi_gets (char *s, int maxwidth, char endchar)
- 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 sbi_console_init (struct sbi_scratch *scratch)

Variables

- int const char * format
- int const char int u32 out_sz

26.32.1 Macro Definition Documentation

26.32.1.1 __printf

```
#define __printf(
            a,
            b ) __attribute__((format(printf, a, b)))
```

26.32.2 Function Documentation

26.32.2.1 __printf() [1/3]

```
int const char int u32 const char int int __printf (
            1 ,
            2  ) const
```

26.32.2.2 __printf() [2/3]

```
int __printf (
            2 ,
            3  )
```

26.32.2.3 __printf() [3/3]

```
int const char int __printf (
            3 ,
            4  )
```

26.32.2.4 sbi_console_init()

```
int sbi_console_init (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.32.2.5 sbi_getc()

```
int sbi_getc (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.32.2.6 sbi_gets()

```
void sbi_gets (
            char * s,
            int maxwidth,
            char endchar )
```

Here is the call graph for this function:

26.32.2.7 sbi_isprintable()

```
bool sbi_isprintable (
            char ch )
```

26.32.2.8 sbi_putc()

```
void sbi_putc (
            char ch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.32.2.9 sbi_puts()

```
void sbi_puts (
            const char * str )
```

Here is the call graph for this function:

26.32.3 Variable Documentation

26.32.3.1 format

```
int const char int u32 const char * format
```

26.32.3.2 out_sz

```
int const char int u32 out_sz
```

26.33 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)

26.33.1 Macro Definition Documentation

26.33.1.1 __AC

```
#define __AC(
            X,
            Y ) (X##Y)
```

26.33.1.2 __STR

```
#define __STR(
            s ) #s
```

26.33.1.3 _AC

```
#define _AC(
            X,
            Y ) __AC(X,Y)
```

26.33.1.4 _AT

```
#define _AT(
            T,
            X ) ((T)(X))
```

26.33.1.5 _BITUL

```
#define _BITUL(
            x ) (_UL(1) << (x))
```

26.33.1.6 _BITULL

```
#define _BITULL(
            x ) (_ULL(1) << (x))
```

26.33.1.7 _UL

```
#define _UL(
            x ) (_AC(x, UL))
```

26.33.1.8 _ULL

```
#define _ULL(
            x ) (_AC(x, ULL))
```

26.33.1.9 STRINGIFY

```
#define STRINGIFY(
            s ) __STR(s)
```

26.33.1.10 UL

```
#define UL(
            x ) (_UL(x))
```

26.33.1.11 ULL

```
#define ULL(
            x ) (_ULL(x))
```

26.34 include/sbi/sbi_csr_detect.h File Reference

```
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_hart.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)

26.34.1 Macro Definition Documentation

26.34.1.1 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;                         \
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;                                    \
})                                  \
```

26.34.1.2 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();        \
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");                            \
})                                  \
```

26.35 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_extension

Macros

- #define SBI_ECALL_VERSION_MAJOR 0
- #define SBI_ECALL_VERSION_MINOR 2
- #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)
- 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_regs ∗regs)
- int sbi_ecall_init (void)

Variables

- struct sbi_ecall_extension ecall_base
- struct sbi_ecall_extension ecall_legacy
- struct sbi_ecall_extension ecall_time
- struct sbi_ecall_extension ecall_rfence

- struct sbi_ecall_extension ecall_ipi
- struct sbi_ecall_extension ecall_vendor
- struct sbi_ecall_extension ecall_hsm

26.35.1 Macro Definition Documentation

26.35.1.1 SBI_ECALL_VERSION_MAJOR

```
#define SBI_ECALL_VERSION_MAJOR 0
```

26.35.1.2 SBI_ECALL_VERSION_MINOR

```
#define SBI_ECALL_VERSION_MINOR 2
```

26.35.1.3 SBI_OPENSBI_IMPID

```
#define SBI_OPENSBI_IMPID 1
```

26.35.2 Function Documentation

26.35.2.1 sbi_ecall_find_extension()

```
struct sbi_ecall_extension* sbi_ecall_find_extension (
            unsigned long extid )
```

Here is the caller graph for this function:

26.35.2.2 sbi_ecall_get_impid()

```
unsigned long sbi_ecall_get_impid (
            void )
```

Here is the caller graph for this function:

26.35.2.3 sbi_ecall_handler()

```
int sbi_ecall_handler (
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.35.2.4 sbi_ecall_init()

```
int sbi_ecall_init (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.35.2.5 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:

26.35.2.6 sbi_ecall_set_impid()

```
void sbi_ecall_set_impid (
            unsigned long impid )
```

26.35.2.7 sbi_ecall_unregister_extension()

```
void sbi_ecall_unregister_extension (
            struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:

26.35.2.8 sbi_ecall_version_major()

```
u16 sbi_ecall_version_major (
            void )
```

Here is the caller graph for this function:

26.35.2.9 sbi_ecall_version_minor()

```
u16 sbi_ecall_version_minor (
            void  )
```

Here is the caller graph for this function:

26.35.3 Variable Documentation

26.35.3.1 ecall_base

```
struct sbi_ecall_extension ecall_base
```

26.35.3.2 ecall_hsm

```
struct sbi_ecall_extension ecall_hsm
```

26.35.3.3 ecall_ipi

```
struct sbi_ecall_extension ecall_ipi
```

26.35.3.4 ecall_legacy

```
struct sbi_ecall_extension ecall_legacy
```

26.35.3.5 ecall_rfence

```
struct sbi_ecall_extension ecall_rfence
```

26.35.3.6 ecall_time

```
struct sbi_ecall_extension ecall_time
```

26.35.3.7 ecall_vendor

```
struct sbi_ecall_extension ecall_vendor
```

26.36 include/sbi/sbi_ecall_interface.h File Reference

This graph shows which files directly or indirectly include this file:

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_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 0x3
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID 0x4
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA 0x5
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID 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_HSM_HART_STATUS_STARTED 0x0
- #define SBI_HSM_HART_STATUS_STOPPED 0x1
- #define SBI_HSM_HART_STATUS_START_PENDING 0x2
- #define SBI_HSM_HART_STATUS_STOP_PENDING 0x3
- #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_LAST_ERR SBI_ERR_ALREADY_AVAILABLE

26.36.1 Macro Definition Documentation

26.36.1.1 SBI_ERR_ALREADY_AVAILABLE

```
#define SBI_ERR_ALREADY_AVAILABLE -6
```

26.36.1.2 SBI_ERR_DENIED

```
#define SBI_ERR_DENIED -4
```

26.36.1.3 SBI_ERR_FAILED

```
#define SBI_ERR_FAILED -1
```

26.36.1.4 SBI_ERR_INVALID_ADDRESS

```
#define SBI_ERR_INVALID_ADDRESS -5
```

26.36.1.5 SBI_ERR_INVALID_PARAM

```
#define SBI_ERR_INVALID_PARAM -3
```

26.36.1.6 SBI_ERR_NOT_SUPPORTED

```
#define SBI_ERR_NOT_SUPPORTED -2
```

26.36.1.7 SBI_EXT_0_1_CLEAR_IPI

```
#define SBI_EXT_0_1_CLEAR_IPI 0x3
```

26.36.1.8 SBI_EXT_0_1_CONSOLE_GETCHAR

```
#define SBI_EXT_0_1_CONSOLE_GETCHAR 0x2
```

26.36.1.9 SBI_EXT_0_1_CONSOLE_PUTCHAR

```
#define SBI_EXT_0_1_CONSOLE_PUTCHAR 0x1
```

26.36.1.10 SBI_EXT_0_1_REMOTE_FENCE_I

```
#define SBI_EXT_0_1_REMOTE_FENCE_I 0x5
```

26.36.1.11 SBI_EXT_0_1_REMOTE_SFENCE_VMA

```
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA 0x6
```

26.36.1.12 SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID

```
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID 0x7
```

26.36.1.13 SBI_EXT_0_1_SEND_IPI

```
#define SBI_EXT_0_1_SEND_IPI 0x4
```

26.36.1.14 SBI_EXT_0_1_SET_TIMER

```
#define SBI_EXT_0_1_SET_TIMER 0x0
```

26.36.1.15 SBI_EXT_0_1_SHUTDOWN

```
#define SBI_EXT_0_1_SHUTDOWN 0x8
```

26.36.1.16 SBI_EXT_BASE

```
#define SBI_EXT_BASE 0x10
```

26.36.1.17 SBI_EXT_BASE_GET_IMP_ID

```
#define SBI_EXT_BASE_GET_IMP_ID 0x1
```

26.36.1.18 SBI_EXT_BASE_GET_IMP_VERSION

```
#define SBI_EXT_BASE_GET_IMP_VERSION 0x2
```

26.36.1.19 SBI_EXT_BASE_GET_MARCHID

```
#define SBI_EXT_BASE_GET_MARCHID 0x5
```

26.36.1.20 SBI_EXT_BASE_GET_MIMPID

```
#define SBI_EXT_BASE_GET_MIMPID 0x6
```

26.36.1.21 SBI_EXT_BASE_GET_MVENDORID

```
#define SBI_EXT_BASE_GET_MVENDORID 0x4
```

26.36.1.22 SBI_EXT_BASE_GET_SPEC_VERSION

```
#define SBI_EXT_BASE_GET_SPEC_VERSION 0x0
```

26.36.1.23 SBI_EXT_BASE_PROBE_EXT

```
#define SBI_EXT_BASE_PROBE_EXT 0x3
```

26.36.1.24 SBI_EXT_FIRMWARE_END

```
#define SBI_EXT_FIRMWARE_END 0x0AFFFFFF
```

26.36.1.25 SBI_EXT_FIRMWARE_START

```
#define SBI_EXT_FIRMWARE_START 0x0A000000
```

26.36.1.26 SBI_EXT_HSM

```
#define SBI_EXT_HSM 0x48534D
```

26.36.1.27 SBI_EXT_HSM_HART_GET_STATUS

```
#define SBI_EXT_HSM_HART_GET_STATUS 0x2
```

26.36.1.28 SBI_EXT_HSM_HART_START

```
#define SBI_EXT_HSM_HART_START 0x0
```

26.36.1.29 SBI_EXT_HSM_HART_STOP

```
#define SBI_EXT_HSM_HART_STOP 0x1
```

26.36.1.30 SBI_EXT_IPI

```
#define SBI_EXT_IPI 0x735049
```

26.36.1.31 SBI_EXT_IPI_SEND_IPI

```
#define SBI_EXT_IPI_SEND_IPI 0x0
```

26.36.1.32 SBI_EXT_RFENCE

```
#define SBI_EXT_RFENCE 0x52464E43
```

26.36.1.33 SBI_EXT_RFENCE_REMOTE_FENCE_I

```
#define SBI_EXT_RFENCE_REMOTE_FENCE_I 0x0
```

26.36.1.34 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA 0x3
```

26.36.1.35 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID 0x4
```

26.36.1.36 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA 0x5
```

26.36.1.37 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID 0x6
```

26.36.1.38 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA

```
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA 0x1
```

26.36.1.39 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID

```
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID 0x2
```

26.36.1.40 SBI_EXT_TIME

```
#define SBI_EXT_TIME 0x54494D45
```

26.36.1.41 SBI_EXT_TIME_SET_TIMER

```
#define SBI_EXT_TIME_SET_TIMER 0x0
```

26.36.1.42 SBI_EXT_VENDOR_END

```
#define SBI_EXT_VENDOR_END 0x09FFFFFF
```

26.36.1.43 SBI_EXT_VENDOR_START

```
#define SBI_EXT_VENDOR_START 0x09000000
```

26.36.1.44 SBI_HSM_HART_STATUS_START_PENDING

```
#define SBI_HSM_HART_STATUS_START_PENDING 0x2
```

26.36.1.45 SBI_HSM_HART_STATUS_STARTED

```
#define SBI_HSM_HART_STATUS_STARTED 0x0
```

26.36.1.46 SBI_HSM_HART_STATUS_STOP_PENDING

```
#define SBI_HSM_HART_STATUS_STOP_PENDING 0x3
```

26.36.1.47 SBI_HSM_HART_STATUS_STOPPED

```
#define SBI_HSM_HART_STATUS_STOPPED 0x1
```

26.36.1.48 SBI_LAST_ERR

```
#define SBI_LAST_ERR SBI_ERR_ALREADY_AVAILABLE
```

26.36.1.49 SBI_SPEC_VERSION_MAJOR_MASK

```
#define SBI_SPEC_VERSION_MAJOR_MASK 0x7f
```

26.36.1.50 SBI_SPEC_VERSION_MAJOR_OFFSET

```
#define SBI_SPEC_VERSION_MAJOR_OFFSET 24
```

26.36.1.51 SBI_SPEC_VERSION_MINOR_MASK

```
#define SBI_SPEC_VERSION_MINOR_MASK 0xffffff
```

26.36.1.52 SBI_SUCCESS

```
#define SBI_SUCCESS 0
```

26.37 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)

26.37.1 Function Documentation

26.37.1.1 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:

26.37.1.2 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:

26.38 include/sbi/sbi_error.h File Reference

```
#include <sbi/sbi_ecall_interface.h>
```

Include dependency graph for sbi_error.h:

This graph shows which files directly or indirectly include this file:

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_ENODEV -1000
- #define SBI_ENOSYS -1001
- #define SBI_ETIMEDOUT -1002
- #define SBI_EIO -1003
- #define SBI_EILL -1004
- #define SBI_ENOSPC -1005
- #define SBI_ENOMEM -1006
- #define SBI_ETRAP -1007
- #define SBI_EUNKNOWN -1008
- #define SBI_ENOENT -1009

26.38.1 Macro Definition Documentation

26.38.1.1 SBI_EALREADY

```
#define SBI_EALREADY SBI_ERR_ALREADY_AVAILABLE
```

26.38.1.2 SBI_EDENIED

```
#define SBI_EDENIED SBI_ERR_DENIED
```

26.38.1.3 SBI_EFAIL

```
#define SBI_EFAIL SBI_ERR_FAILED
```

26.38.1.4 SBI_EILL

```
#define SBI_EILL -1004
```

26.38.1.5 SBI_EINVAL

```
#define SBI_EINVAL SBI_ERR_INVALID_PARAM
```

26.38.1.6 SBI_EINVALID_ADDR

```
#define SBI_EINVALID_ADDR SBI_ERR_INVALID_ADDRESS
```

26.38.1.7 SBI_EIO

```
#define SBI_EIO -1003
```

26.38.1.8 SBI_ENODEV

```
#define SBI_ENODEV -1000
```

26.38.1.9 SBI_ENOENT

```
#define SBI_ENOENT -1009
```

26.38.1.10 SBI_ENOMEM

```
#define SBI_ENOMEM -1006
```

26.38.1.11 SBI_ENOSPC

```
#define SBI_ENOSPC -1005
```

26.38.1.12 SBI_ENOSYS

```
#define SBI_ENOSYS -1001
```

26.38.1.13 SBI_ENOTSUPP

```
#define SBI_ENOTSUPP SBI_ERR_NOT_SUPPORTED
```

26.38.1.14 SBI_ETIMEDOUT

```
#define SBI_ETIMEDOUT -1002
```

26.38.1.15 SBI_ETRAP

```
#define SBI_ETRAP -1007
```

26.38.1.16 SBI_EUNKNOWN

```
#define SBI_EUNKNOWN -1008
```

26.38.1.17 SBI_OK

```
#define SBI_OK 0
```

26.39 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

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)
- void sbi_fifo_init (struct sbi_fifo ∗fifo, void ∗queue_mem, u16 entries, u16 entry_size)
- bool sbi_fifo_is_empty (struct sbi_fifo ∗fifo)
- bool 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)

26.39.1 Enumeration Type Documentation

26.39.1.1 sbi_fifo_inplace_update_types

```
enum sbi_fifo_inplace_update_types
```

Enumerator

SBI_FIFO_SKIP	
SBI_FIFO_UPDATED	
SBI_FIFO_UNCHANGED	

26.39.2 Function Documentation

26.39.2.1 sbi_fifo_avail()

```
u16 sbi_fifo_avail (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

26.39.2.2 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:

26.39.2.3 sbi_fifo_enqueue()

```
int sbi_fifo_enqueue (
            struct sbi_fifo * fifo,
            void * data )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.39.2.4 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:

26.39.2.5 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:

26.39.2.6 sbi_fifo_is_empty()

```
bool sbi_fifo_is_empty (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

26.39.2.7 sbi_fifo_is_full()

```
bool sbi_fifo_is_full (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

26.40 include/sbi/sbi_hart.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_hart.h:

This graph shows which files directly or indirectly include this file:

Enumerations

- enum sbi_hart_features {
  SBI_HART_HAS_PMP = (1 << 0), SBI_HART_HAS_SCOUNTEREN = (1 << 1), SBI_HART_HAS_MCOUNTEREN = (1 << 2), SBI_HART_HAS_TIME = (1 << 3),
  SBI_HART_HAS_LAST_FEATURE = SBI_HART_HAS_TIME }

Functions

- int sbi_hart_init (struct sbi_scratch *scratch, u32 hartid, bool cold_boot)
- static ulong sbi_hart_expected_trap_addr (void)
- void sbi_hart_delegation_dump (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_count (struct sbi_scratch *scratch)
- int sbi_hart_pmp_get (struct sbi_scratch *scratch, unsigned int n, unsigned long *prot_out, unsigned long *addr_out, unsigned long *size)
- void sbi_hart_pmp_dump (struct sbi_scratch *scratch)
- int sbi_hart_pmp_check_addr (struct sbi_scratch *scratch, unsigned long daddr, unsigned long attr)
- bool sbi_hart_has_feature (struct sbi_scratch *scratch, unsigned long feature)
- void sbi_hart_get_features_str (struct sbi_scratch *scratch, char *features_str, int nfstr)
- void __attribute__ ((noreturn)) sbi_hart_hang(void)

Variables

- 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

26.40.1 Enumeration Type Documentation

26.40.1.1 sbi_hart_features

```
enum sbi_hart_features
```

Possible feature flags of a hart

Enumerator

SBI_HART_HAS_PMP	Hart has PMP support
SBI_HART_HAS_SCOUNTEREN	Hart has S-mode counter enable
SBI_HART_HAS_MCOUNTEREN	Hart has M-mode counter enable
SBI_HART_HAS_TIME	HART has timer csr implementation in hardware
SBI_HART_HAS_LAST_FEATURE	Last index of Hart features

26.40.2 Function Documentation

26.40.2.1 __attribute__()

```
void __attribute__ (
            (noreturn)  )
```

26.40.2.2 sbi_hart_delegation_dump()

```
void sbi_hart_delegation_dump (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.40.2.3 sbi_hart_expected_trap_addr()

```
static ulong sbi_hart_expected_trap_addr (
            void  )  [inline], [static]
```

Here is the caller graph for this function:

26.40.2.4 sbi_hart_get_features_str()

```
void sbi_hart_get_features_str (
            struct sbi_scratch * scratch,
            char * features_str,
            int nfstr )
```

Get the hart features in string format

Parameters

scratch	pointer to the HART scratch space
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:

26.40.2.5 sbi_hart_has_feature()

```
bool sbi_hart_has_feature (
            struct sbi_scratch * scratch,
            unsigned long feature )
```

Check whether a particular hart feature is available

Parameters

scratch	pointer to the HART scratch space
feature	the feature to check

Returns

true (feature available) or false (feature not available)

Here is the caller graph for this function:

26.40.2.6 sbi_hart_init()

```
int sbi_hart_init (
            struct sbi_scratch * scratch,
            u32 hartid,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.40.2.7 sbi_hart_pmp_check_addr()

```
int sbi_hart_pmp_check_addr (
            struct sbi_scratch * scratch,
            unsigned long daddr,
            unsigned long attr )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.40.2.8 sbi_hart_pmp_count()

```
unsigned int sbi_hart_pmp_count (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

26.40.2.9 sbi_hart_pmp_dump()

```
void sbi_hart_pmp_dump (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.40.2.10 sbi_hart_pmp_get()

```
int sbi_hart_pmp_get (
            struct sbi_scratch * scratch,
            unsigned int n,
            unsigned long * prot_out,
            unsigned long * addr_out,
            unsigned long * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.40.3 Variable Documentation

26.40.3.1 arg1

```
void unsigned long arg1
```

26.40.3.2 next_addr

```
void unsigned long unsigned long next_addr
```

26.40.3.3 next_mode

```
void unsigned long unsigned long unsigned long next_mode
```

26.40.3.4 next_virt

```
void unsigned long unsigned long unsigned long bool next_virt
```

26.40.3.5 sbi_hart_expected_trap

```
void(* sbi_hart_expected_trap) (void)
```

26.41 include/sbi/sbi_hartmask.h File Reference

```
#include <sbi/sbi_bitmap.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) bitmap_zero_except(((__m)->bits), (__h), SBI_HARTMASK_MAX_BITS)
- #define sbi_hartmask_bits(__m) ((__m)->bits)
- #define sbi_hartmask_for_each_hart(__h, __m) for_each_set_bit(__h, (__m)->bits, SBI_HARTMASK_MAX_BITS)

Functions

- static void sbi_hartmask_set_hart (u32 h, struct sbi_hartmask *m)
- static void sbi_hartmask_clear_hart (u32 h, struct sbi_hartmask *m)
- static int sbi_hartmask_test_hart (u32 h, 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_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)

26.41.1 Macro Definition Documentation

26.41.1.1 sbi_hartmask_bits

```
#define sbi_hartmask_bits(
            __m ) ((__m)->bits)
```

Get underlying bitmap of hartmask

Parameters

m	the hartmask pointer

26.41.1.2 sbi_hartmask_for_each_hart

```
#define sbi_hartmask_for_each_hart(
            __h,
            __m ) for_each_set_bit(__h, (__m)->bits, SBI_HARTMASK_MAX_BITS)
```

Iterate over each HART in hartmask

26.41.1.3 SBI_HARTMASK_INIT

```
#define SBI_HARTMASK_INIT(
            __m ) bitmap_zero(((__m)->bits), SBI_HARTMASK_MAX_BITS)
```

Initialize hartmask to zero

26.41.1.4 SBI_HARTMASK_INIT_EXCEPT

```
#define SBI_HARTMASK_INIT_EXCEPT(
            __m,
            __h ) bitmap_zero_except(((__m)->bits), (__h), SBI_HARTMASK_MAX_BITS)
```

Initialize hartmask to zero except a particular HART id

26.41.1.5 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.

26.41.2 Function Documentation

26.41.2.1 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:

26.41.2.2 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:

26.41.2.3 sbi_hartmask_clear_hart()

```
static void sbi_hartmask_clear_hart (
            u32 h,
            struct sbi_hartmask * m )  [inline], [static]
```

Clear a HART in hartmask

Parameters

h	HART id to clear
m	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:

26.41.2.4 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:

26.41.2.5 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:

26.41.2.6 sbi_hartmask_set_hart()

```
static void sbi_hartmask_set_hart (
            u32 h,
            struct sbi_hartmask * m )  [inline], [static]
```

Set a HART 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:

26.41.2.7 sbi_hartmask_test_hart()

```
static int sbi_hartmask_test_hart (
            u32 h,
            struct sbi_hartmask * m )  [inline], [static]
```

Test a HART in hartmask

Parameters

h	HART id to test
m	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:

26.41.2.8 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:

26.42 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, unsigned long vmid)
- void __sbi_hfence_gvma_vmid (unsigned long vmid)
- void __sbi_hfence_gvma_gpa (unsigned long gpa)
- 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)

26.42.1 Function Documentation

26.42.1.1 __sbi_hfence_gvma_all()

```
void __sbi_hfence_gvma_all (
            void  )
```

Invalidate all possible Stage2 TLBs Here is the caller graph for this function:

26.42.1.2 __sbi_hfence_gvma_gpa()

```
void __sbi_hfence_gvma_gpa (
            unsigned long gpa )
```

Invalidate Stage2 TLBs for given guest physical address Here is the caller graph for this function:

26.42.1.3 __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:

26.42.1.4 __sbi_hfence_gvma_vmid_gpa()

```
void __sbi_hfence_gvma_vmid_gpa (
            unsigned long gpa,
            unsigned long vmid )
```

Invalidate Stage2 TLBs for given VMID and guest physical address Here is the caller graph for this function:

26.42.1.5 __sbi_hfence_vvma_all()

```
void __sbi_hfence_vvma_all (
            void )
```

Invalidate all possible Stage2 TLBs Here is the caller graph for this function:

26.42.1.6 __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:

26.42.1.7 __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:

26.42.1.8 __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:

26.43 include/sbi/sbi_hsm.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_hsm.h:

This graph shows which files directly or indirectly include this file:

Macros

- #define SBI_HART_STOPPED 0
- #define SBI_HART_STOPPING 1
- #define SBI_HART_STARTING 2
- #define SBI_HART_STARTED 3
- #define SBI_HART_UNKNOWN 4

Functions

- int sbi_hsm_init (struct sbi_scratch ∗scratch, u32 hartid, bool cold_boot)
- void __noreturn sbi_hsm_exit (struct sbi_scratch ∗scratch)
- int sbi_hsm_hart_start (struct sbi_scratch ∗scratch, u32 hartid, ulong saddr, ulong priv)
- int sbi_hsm_hart_stop (struct sbi_scratch ∗scratch, bool exitnow)
- int sbi_hsm_hart_get_state (u32 hartid)
- int sbi_hsm_hart_state_to_status (int state)
- bool sbi_hsm_hart_started (u32 hartid)
- int sbi_hsm_hart_started_mask (ulong hbase, ulong ∗out_hmask)
- void sbi_hsm_prepare_next_jump (struct sbi_scratch ∗scratch, u32 hartid)

26.43.1 Macro Definition Documentation

26.43.1.1 SBI_HART_STARTED

```
#define SBI_HART_STARTED 3
```

26.43.1.2 SBI_HART_STARTING

```
#define SBI_HART_STARTING 2
```

26.43.1.3 SBI_HART_STOPPED

```
#define SBI_HART_STOPPED 0
```

Hart state values

26.43.1.4 SBI_HART_STOPPING

```
#define SBI_HART_STOPPING 1
```

26.43.1.5 SBI_HART_UNKNOWN

```
#define SBI_HART_UNKNOWN 4
```

26.43.2 Function Documentation

26.43.2.1 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:

26.43.2.2 sbi_hsm_hart_get_state()

```
int sbi_hsm_hart_get_state (
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.43.2.3 sbi_hsm_hart_start()

```
int sbi_hsm_hart_start (
            struct sbi_scratch * scratch,
            u32 hartid,
            ulong saddr,
            ulong priv )
```

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:

26.43.2.4 sbi_hsm_hart_started()

```
bool sbi_hsm_hart_started (
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.43.2.5 sbi_hsm_hart_started_mask()

```
int sbi_hsm_hart_started_mask (
            ulong hbase,
            ulong * out_hmask )
```

Get ulong HART mask for given HART base ID

Parameters

hbase	the HART base ID
out_hmask	the output ulong HART mask

Returns

0 on success and SBI_Exxx ($<$ 0) on failure Note: the output HART mask will be set to zero on failure as well.

Here is the call graph for this function:

Here is the caller graph for this function:

26.43.2.6 sbi_hsm_hart_state_to_status()

```
int sbi_hsm_hart_state_to_status (
            int state )
```

Here is the caller graph for this function:

26.43.2.7 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:

26.43.2.8 sbi_hsm_init()

```
int sbi_hsm_init (
            struct sbi_scratch * scratch,
            u32 hartid,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.43.2.9 sbi_hsm_prepare_next_jump()

```
void sbi_hsm_prepare_next_jump (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.44 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 (ulong insn, struct sbi_trap_regs *regs)

26.44.1 Function Documentation

26.44.1.1 sbi_illegal_insn_handler()

```
int sbi_illegal_insn_handler (
            ulong insn,
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.45 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_init_count (u32 hartid)
- void __noreturn sbi_exit (struct sbi_scratch *scratch)

26.45.1 Function Documentation

26.45.1.1 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:

26.45.1.2 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:

26.45.1.3 sbi_init_count()

```
unsigned long sbi_init_count (
            u32 hartid )
```

Here is the caller graph for this function:

26.46 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_event_ops

Macros

- #define SBI_IPI_EVENT_MAX __riscv_xlen

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_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_ipi_exit (struct sbi_scratch ∗scratch)

26.46.1 Macro Definition Documentation

26.46.1.1 SBI_IPI_EVENT_MAX

`#define SBI_IPI_EVENT_MAX __riscv_xlen`

26.46.2 Function Documentation

26.46.2.1 sbi_ipi_clear_smode()

```
void sbi_ipi_clear_smode (
            void  )
```

Here is the caller graph for this function:

26.46.2.2 sbi_ipi_event_create()

```
int sbi_ipi_event_create (
            const struct sbi_ipi_event_ops * ops )
```

Here is the caller graph for this function:

26.46.2.3 sbi_ipi_event_destroy()

```
void sbi_ipi_event_destroy (
            u32 event )
```

26.46.2.4 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:

26.46.2.5 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:

26.46.2.6 sbi_ipi_process()

```
void sbi_ipi_process (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.46.2.7 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:

26.46.2.8 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:

26.46.2.9 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:

26.47 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 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)

26.47.1 Macro Definition Documentation

26.47.1.1 SBI_INIT_LIST_HEAD

```
#define SBI_INIT_LIST_HEAD(
            ptr )
```

Value:

```
do { \
    (ptr)->next = ptr; (ptr)->prev = ptr; \
} while (0);
```

26.47.1.2 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.

26.47.1.3 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.

26.47.1.4 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.

26.47.1.5 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.

26.47.1.6 SBI_LIST_HEAD

```
#define SBI_LIST_HEAD(
            _lname ) struct sbi_dlist _lname = SBI_LIST_HEAD_INIT(_lname)
```

26.47.1.7 SBI_LIST_HEAD_INIT

```
#define SBI_LIST_HEAD_INIT(
```

```
__lname ) { &(__lname), &(__lname) }
```

26.47.1.8 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.

26.47.1.9 SBI_LIST_POISON_NEXT

```
#define SBI_LIST_POISON_NEXT 0xFADEBABE
```

26.47.1.10 SBI_LIST_POISON_PREV

```
#define SBI_LIST_POISON_PREV 0xDEADBEEF
```

26.47.2 Function Documentation

26.47.2.1 __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:

26.47.2.2 __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:

26.47.2.3 __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:

26.47.2.4 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:

26.47.2.5 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:

26.47.2.6 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:

26.47.2.7 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:

26.48 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)

26.48.1 Function Documentation

26.48.1.1 log2roundup()

```
unsigned long log2roundup (
            unsigned long x )
```

Here is the caller graph for this function:

26.49 include/sbi/sbi_misaligned_ldst.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_misaligned_ldst.h:

This graph shows which files directly or indirectly include this file:

Functions

- int sbi_misaligned_load_handler (ulong addr, ulong tval2, ulong tinst, struct sbi_trap_regs *regs)
- int sbi_misaligned_store_handler (ulong addr, ulong tval2, ulong tinst, struct sbi_trap_regs *regs)

26.49.1 Function Documentation

26.49.1.1 sbi_misaligned_load_handler()

```
int sbi_misaligned_load_handler (
            ulong addr,
            ulong tval2,
            ulong tinst,
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.49.1.2 sbi_misaligned_store_handler()

```
int sbi_misaligned_store_handler (
            ulong addr,
            ulong tval2,
            ulong tinst,
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.50 include/sbi/sbi_platform.h File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_version.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_OPS_OFFSET (0x58)
- #define SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET (0x58 + __SIZEOF_POINTER__)
- #define SBI_PLATFORM_HART_INDEX2ID_OFFSET (0x58 + (__SIZEOF_POINTER__ * 2))
- #define SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT (1UL << 12)
- #define SBI_PLATFORM_DEFAULT_FEATURES (SBI_PLATFORM_HAS_TIMER_VALUE | SBI_PLATFORM_HAS_MFAULT
- #define SBI_PLATFORM_RESET_SHUTDOWN 0
- #define SBI_PLATFORM_RESET_COLD 1
- #define SBI_PLATFORM_RESET_WARM 2
- #define SBI_PLATFORM_DEFAULT_HART_STACK_SIZE 8192
- #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_timer_value(__p) ((__p)->features & SBI_PLATFORM_HAS_TIMER_VALUE)
- #define sbi_platform_has_hart_hotplug(__p) ((__p)->features & SBI_PLATFORM_HAS_HART_HOTPLUG)
- #define sbi_platform_has_mfaults_delegation(__p) ((__p)->features & SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
- #define sbi_platform_has_hart_secondary_boot(__p) ((__p)->features & SBI_PLATFORM_HAS_HART_SECONDARY_BOOT

Enumerations

- enum sbi_platform_features {
  SBI_PLATFORM_HAS_TIMER_VALUE = (1 << 0), SBI_PLATFORM_HAS_HART_HOTPLUG = (1 << 1),
  SBI_PLATFORM_HAS_MFAULTS_DELEGATION = (1 << 2), SBI_PLATFORM_HAS_HART_SECONDARY_BOOT = (1 << 3),
  SBI_PLATFORM_HAS_LAST_FEATURE = SBI_PLATFORM_HAS_HART_SECONDARY_BOOT }

Functions

- u32 sbi_platform_hart_index (const struct sbi_platform *plat, u32 hartid)
- 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_hart_count (const struct sbi_platform *plat)
- static u32 sbi_platform_hart_stack_size (const struct sbi_platform *plat)
- static bool sbi_platform_hart_invalid (const struct sbi_platform *plat, u32 hartid)
- static int sbi_platform_hart_start (const struct sbi_platform *plat, u32 hartid, ulong saddr)
- static int sbi_platform_hart_stop (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 u32 sbi_platform_pmp_region_count (const struct sbi_platform *plat, u32 hartid)
- static int sbi_platform_pmp_region_info (const struct sbi_platform *plat, u32 hartid, u32 index, ulong *prot, ulong *addr, ulong *log2size)
- static void sbi_platform_console_putc (const struct sbi_platform *plat, char ch)
- static int sbi_platform_console_getc (const struct sbi_platform *plat)
- static int sbi_platform_console_init (const struct sbi_platform *plat)
- static int sbi_platform_irqchip_init (const struct sbi_platform *plat, bool cold_boot)
- static void sbi_platform_irqchip_exit (const struct sbi_platform *plat)
- static void sbi_platform_ipi_send (const struct sbi_platform *plat, u32 target_hart)
- static void sbi_platform_ipi_clear (const struct sbi_platform *plat, u32 target_hart)
- static int sbi_platform_ipi_init (const struct sbi_platform *plat, bool cold_boot)
- static void sbi_platform_ipi_exit (const struct sbi_platform *plat)
- static u64 sbi_platform_timer_value (const struct sbi_platform *plat)
- static void sbi_platform_timer_event_start (const struct sbi_platform *plat, u64 next_event)
- static void sbi_platform_timer_event_stop (const struct sbi_platform *plat)
- static int sbi_platform_timer_init (const struct sbi_platform *plat, bool cold_boot)
- static void sbi_platform_timer_exit (const struct sbi_platform *plat)
- static int sbi_platform_system_reset (const struct sbi_platform *plat, u32 reset_type)
- static int sbi_platform_vendor_ext_check (const struct sbi_platform *plat, long extid)
- static int sbi_platform_vendor_ext_provider (const struct sbi_platform *plat, long extid, long funcid, unsigned long *args, unsigned long *out_value, struct sbi_trap_info *out_trap)

Variables

- struct sbi_platform_operations __packed

26.50.1 Macro Definition Documentation

26.50.1.1 SBI_PLATFORM_DEFAULT_FEATURES

```
#define SBI_PLATFORM_DEFAULT_FEATURES (SBI_PLATFORM_HAS_TIMER_VALUE | SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
```

Default feature set for a platform

26.50.1.2 SBI_PLATFORM_DEFAULT_HART_STACK_SIZE

```
#define SBI_PLATFORM_DEFAULT_HART_STACK_SIZE 8192
```

Platform default per-HART stack size for exception/interrupt handling

26.50.1.3 SBI_PLATFORM_FEATURES_OFFSET

```
#define SBI_PLATFORM_FEATURES_OFFSET (0x48)
```

Offset of features in struct sbi_platform

26.50.1.4 SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET

```
#define SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET (0x58 + __SIZEOF_POINTER__)
```

Offset of firmware_context in struct sbi_platform

26.50.1.5 SBI_PLATFORM_HART_COUNT_OFFSET

```
#define SBI_PLATFORM_HART_COUNT_OFFSET (0x50)
```

Offset of hart_count in struct sbi_platform

26.50.1.6 SBI_PLATFORM_HART_INDEX2ID_OFFSET

```
#define SBI_PLATFORM_HART_INDEX2ID_OFFSET (0x58 + (__SIZEOF_POINTER__ * 2))
```

Offset of hart_index2id in struct sbi_platform

26.50.1.7 SBI_PLATFORM_HART_STACK_SIZE_OFFSET

```
#define SBI_PLATFORM_HART_STACK_SIZE_OFFSET (0x54)
```

Offset of hart_stack_size in struct sbi_platform

26.50.1.8 sbi_platform_has_hart_hotplug

```
#define sbi_platform_has_hart_hotplug(
            __p ) ((__p)->features & SBI_PLATFORM_HAS_HART_HOTPLUG)
```

Check whether the platform supports HART hotplug

26.50.1.9 sbi_platform_has_hart_secondary_boot

```
#define sbi_platform_has_hart_secondary_boot(
            __p ) ((__p)->features & SBI_PLATFORM_HAS_HART_SECONDARY_BOOT)
```

Check whether the platform supports custom secondary hart booting support

26.50.1.10 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

26.50.1.11 sbi_platform_has_timer_value

```
#define sbi_platform_has_timer_value(
            __p ) ((__p)->features & SBI_PLATFORM_HAS_TIMER_VALUE)
```

Check whether the platform supports timer value

26.50.1.12 SBI_PLATFORM_NAME_OFFSET

```
#define SBI_PLATFORM_NAME_OFFSET (0x08)
```

Offset of name in struct sbi_platform

26.50.1.13 SBI_PLATFORM_OPENSBI_VERSION_OFFSET

```
#define SBI_PLATFORM_OPENSBI_VERSION_OFFSET (0x00)
```

Offset of opensbi_version in struct sbi_platform

26.50.1.14 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

26.50.1.15 SBI_PLATFORM_OPS_OFFSET

```
#define SBI_PLATFORM_OPS_OFFSET (0x58)
```

Offset of platform_ops_addr in struct sbi_platform

26.50.1.16 sbi_platform_ptr

```
#define sbi_platform_ptr(
            __s ) ((const struct sbi_platform *)((__s)->platform_addr))
```

Get pointer to sbi_platform for sbi_scratch pointer

26.50.1.17 SBI_PLATFORM_RESET_COLD

```
#define SBI_PLATFORM_RESET_COLD 1
```

26.50.1.18 SBI_PLATFORM_RESET_SHUTDOWN

```
#define SBI_PLATFORM_RESET_SHUTDOWN 0
```

Reset the platform

26.50.1.19 SBI_PLATFORM_RESET_WARM

```
#define SBI_PLATFORM_RESET_WARM 2
```

26.50.1.20 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

26.50.1.21 SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT

```
#define SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT (1UL << 12)
```

26.50.1.22 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

26.50.1.23 SBI_PLATFORM_VERSION_OFFSET

```
#define SBI_PLATFORM_VERSION_OFFSET (0x04)
```

Offset of platform_version in struct sbi_platform

26.50.2 Enumeration Type Documentation

26.50.2.1 sbi_platform_features

```
enum sbi_platform_features
```

Possible feature flags of a platform

Enumerator

SBI_PLATFORM_HAS_TIMER_VALUE	Platform has timer value
SBI_PLATFORM_HAS_HART_HOTPLUG	Platform has HART hotplug support
SBI_PLATFORM_HAS_MFAULTS_DELEGATION	Platform has fault delegation support
SBI_PLATFORM_HAS_HART_SECONDARY_BOOT	Platform has custom secondary hart booting support
SBI_PLATFORM_HAS_LAST_FEATURE	Last index of Platform features

26.50.3 Function Documentation

26.50.3.1 sbi_platform_console_getc()

```
static int sbi_platform_console_getc (
            const struct sbi_platform * plat )  [inline], [static]
```

Read a character from the platform console input

Parameters

plat	pointer to struct sbi_platform

Returns

character read from console input

Here is the caller graph for this function:

26.50.3.2 sbi_platform_console_init()

```
static int sbi_platform_console_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Initialize the platform console

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:

26.50.3.3 sbi_platform_console_putc()

```
static void sbi_platform_console_putc (
            const struct sbi_platform * plat,
            char ch )  [inline], [static]
```

Write a character to the platform console output

Parameters

plat	pointer to struct sbi_platform
ch	character to write

Here is the caller graph for this function:

26.50.3.4 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:

26.50.3.5 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:

26.50.3.6 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:

26.50.3.7 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:

26.50.3.8 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:

26.50.3.9 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:

26.50.3.10 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:

26.50.3.11 sbi_platform_hart_index()

```
u32 sbi_platform_hart_index (
            const struct sbi_platform * plat,
            u32 hartid )
```

Get HART index for the given HART

Parameters

plat	pointer to struct sbi_platform
hartid	HART ID

Returns

0 <= value < hart_count for valid HART otherwise -1U

Here is the caller graph for this function:

26.50.3.12 sbi_platform_hart_invalid()

```
static bool sbi_platform_hart_invalid (
            const struct sbi_platform * plat,
            u32 hartid )  [inline], [static]
```

Check whether given HART is invalid

Parameters

plat	pointer to struct sbi_platform
hartid	HART ID

Returns

TRUE if HART is invalid and FALSE otherwise

Here is the call graph for this function:

Here is the caller graph for this function:

26.50.3.13 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

26.50.3.14 sbi_platform_hart_start()

```
static int sbi_platform_hart_start (
            const struct sbi_platform * plat,
            u32 hartid,
            ulong saddr )  [inline], [static]
```

Bringup a given hart from previous stage. Platform should implement this operation if they support a custom mechanism to start a hart. Otherwise, a generic WFI based approach will be used to start/stop a hart in OpenSBI.

Parameters

plat	pointer to struct sbi_platform
hartid	HART id
saddr	M-mode start physical address for the HART

Returns

0 if sucessful and negative error code on failure

Here is the caller graph for this function:

26.50.3.15 sbi_platform_hart_stop()

```
static int sbi_platform_hart_stop (
            const struct sbi_platform * plat )  [inline], [static]
```

Stop the current hart in OpenSBI.

Parameters

plat	pointer to struct sbi_platform

Returns

Negative error code on failure. It doesn't return on success.

Here is the caller graph for this function:

26.50.3.16 sbi_platform_ipi_clear()

```
static void sbi_platform_ipi_clear (
            const struct sbi_platform * plat,
            u32 target_hart )  [inline], [static]
```

Clear IPI for a target HART

Parameters

plat	pointer to struct sbi_platform
target_hart	HART ID of IPI target

Here is the caller graph for this function:

26.50.3.17 sbi_platform_ipi_exit()

```
static void sbi_platform_ipi_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Exit the platform IPI support for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:

26.50.3.18 sbi_platform_ipi_init()

```
static int sbi_platform_ipi_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Initialize the platform IPI support 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:

26.50.3.19 sbi_platform_ipi_send()

```
static void sbi_platform_ipi_send (
            const struct sbi_platform * plat,
            u32 target_hart )  [inline], [static]
```

Send IPI to a target HART

Parameters

plat	pointer to struct sbi_platform
target_hart	HART ID of IPI target

Here is the caller graph for this function:

26.50.3.20 sbi_platform_irqchip_exit()

```
static void sbi_platform_irqchip_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Exit the platform interrupt controller for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:

26.50.3.21 sbi_platform_irqchip_init()

```
static int sbi_platform_irqchip_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Initialize the platform interrupt controller 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:

26.50.3.22 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:

26.50.3.23 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:

26.50.3.24 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:

26.50.3.25 sbi_platform_pmp_region_count()

```
static u32 sbi_platform_pmp_region_count (
            const struct sbi_platform * plat,
            u32 hartid )  [inline], [static]
```

Get the number of PMP regions of a HART

Parameters

plat	pointer to struct sbi_platform
hartid	HART ID

Returns

number of PMP regions

Here is the caller graph for this function:

26.50.3.26 sbi_platform_pmp_region_info()

```
static int sbi_platform_pmp_region_info (
            const struct sbi_platform * plat,
            u32 hartid,
            u32 index,
            ulong * prot,
            ulong * addr,
            ulong * log2size )  [inline], [static]
```

Get PMP regions details (namely: protection, base address, and size) of a HART

Parameters

plat	pointer to struct sbi_platform
hartid	HART ID
index	index of PMP region for which we want details

Parameters

prot	output pointer for PMP region protection
addr	output pointer for PMP region base address
log2size	output pointer for log-of-2 PMP region size

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

26.50.3.27 sbi_platform_system_reset()

```
static int sbi_platform_system_reset (
            const struct sbi_platform * plat,
            u32 reset_type ) [inline], [static]
```

Reset the platform

Parameters

plat	pointer to struct sbi_platform
reset_type	type of reset

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

26.50.3.28 sbi_platform_timer_event_start()

```
static void sbi_platform_timer_event_start (
            const struct sbi_platform * plat,
            u64 next_event ) [inline], [static]
```

Start platform timer event for current HART

Parameters

plat	pointer to struct struct sbi_platform
next_event	timer value when timer event will happen

Here is the caller graph for this function:

26.50.3.29 sbi_platform_timer_event_stop()

```
static void sbi_platform_timer_event_stop (
            const struct sbi_platform * plat )  [inline], [static]
```

Stop platform timer event for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:

26.50.3.30 sbi_platform_timer_exit()

```
static void sbi_platform_timer_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Exit the platform timer for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:

26.50.3.31 sbi_platform_timer_init()

```
static int sbi_platform_timer_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Initialize the platform timer 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:

26.50.3.32 sbi_platform_timer_value()

```
static u64 sbi_platform_timer_value (
            const struct sbi_platform * plat )  [inline], [static]
```

Get platform timer value

Parameters

plat	pointer to struct sbi_platform

Returns

64-bit timer value

Here is the caller graph for this function:

26.50.3.33 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:

26.50.3.34 sbi_platform_vendor_ext_check()

```
static int sbi_platform_vendor_ext_check (
            const struct sbi_platform * plat,
            long extid )  [inline], [static]
```

Check if a vendor extension is implemented or not.

Parameters

plat	pointer to struct sbi_platform
extid	vendor SBI extension id

Returns

0 if extid is not implemented and 1 if implemented

Here is the caller graph for this function:

26.50.3.35 sbi_platform_vendor_ext_provider()

```
static int sbi_platform_vendor_ext_provider (
            const struct sbi_platform * plat,
            long extid,
            long funcid,
            unsigned long * args,
            unsigned long * out_value,
            struct sbi_trap_info * out_trap )  [inline], [static]
```

Invoke platform specific vendor SBI extension implementation.

Parameters

plat	pointer to struct sbi_platform
extid	vendor SBI extension id
funcid	SBI function id within the extension id
args	pointer to arguments 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:

26.50.4 Variable Documentation

26.50.4.1 __packed

```
struct sbi_platform __packed
```

26.51 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_NEXT_ARG1_OFFSET (2 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_NEXT_ADDR_OFFSET (3 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_NEXT_MODE_OFFSET (4 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_WARMBOOT_ADDR_OFFSET (5 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_PLATFORM_ADDR_OFFSET (6 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET (7 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_TMP0_OFFSET (8 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_OPTIONS_OFFSET (9 ∗ __SIZEOF_POINTER__)

- #define SBI_SCRATCH_EXTRA_SPACE_OFFSET (10 ∗ __SIZEOF_POINTER__)
- #define SBI_SCRATCH_SIZE (0x1000)
- #define sbi_scratch_thishart_ptr() ((struct sbi_scratch ∗)csr_read(CSR_MSCRATCH))
- #define sbi_scratch_thishart_arg1_ptr() ((void ∗)(sbi_scratch_thishart_ptr()->next_arg1))
- #define sbi_scratch_offset_ptr(scratch, offset) ((void ∗)scratch + (offset))
- #define sbi_scratch_thishart_offset_ptr(offset) ((void ∗)sbi_scratch_thishart_ptr() + (offset))
- #define sbi_hartid_to_scratch(__hartid) hartid_to_scratch_table[__hartid]
- #define sbi_scratch_last_hartid() last_hartid_having_scratch

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, const char ∗owner)
- void sbi_scratch_free_offset (unsigned long offset)

Variables

- struct sbi_scratch __packed
- struct sbi_scratch ∗ hartid_to_scratch_table [ ]
- u32 last_hartid_having_scratch

26.51.1 Macro Definition Documentation

26.51.1.1 sbi_hartid_to_scratch

```
#define sbi_hartid_to_scratch(
            __hartid ) hartid_to_scratch_table[__hartid]
```

Get sbi_scratch from HART id

26.51.1.2 SBI_SCRATCH_EXTRA_SPACE_OFFSET

```
#define SBI_SCRATCH_EXTRA_SPACE_OFFSET (10 * __SIZEOF_POINTER__)
```

Offset of extra space in sbi_scratch

26.51.1.3 SBI_SCRATCH_FW_SIZE_OFFSET

```
#define SBI_SCRATCH_FW_SIZE_OFFSET (1 * __SIZEOF_POINTER__)
```

Offset of fw_size member in sbi_scratch

26.51.1.4 SBI_SCRATCH_FW_START_OFFSET

```
#define SBI_SCRATCH_FW_START_OFFSET (0 * __SIZEOF_POINTER__)
```

Offset of fw_start member in sbi_scratch

26.51.1.5 SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET

```
#define SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET (7 * __SIZEOF_POINTER__)
```

Offset of hartid_to_scratch member in sbi_scratch

26.51.1.6 sbi_scratch_last_hartid

```
#define sbi_scratch_last_hartid( ) last_hartid_having_scratch
```

Get last HART id having a sbi_scratch pointer

26.51.1.7 SBI_SCRATCH_NEXT_ADDR_OFFSET

```
#define SBI_SCRATCH_NEXT_ADDR_OFFSET (3 * __SIZEOF_POINTER__)
```

Offset of next_addr member in sbi_scratch

26.51.1.8 SBI_SCRATCH_NEXT_ARG1_OFFSET

```
#define SBI_SCRATCH_NEXT_ARG1_OFFSET (2 * __SIZEOF_POINTER__)
```

Offset of next_arg1 member in sbi_scratch

26.51.1.9 SBI_SCRATCH_NEXT_MODE_OFFSET

```
#define SBI_SCRATCH_NEXT_MODE_OFFSET (4 * __SIZEOF_POINTER__)
```

Offset of next_mode member in sbi_scratch

26.51.1.10 sbi_scratch_offset_ptr

```
#define sbi_scratch_offset_ptr(
            scratch,
            offset ) ((void *)scratch + (offset))
```

Get pointer from offset in sbi_scratch

26.51.1.11 SBI_SCRATCH_OPTIONS_OFFSET

```
#define SBI_SCRATCH_OPTIONS_OFFSET (9 * __SIZEOF_POINTER__)
```

Offset of options member in sbi_scratch

26.51.1.12 SBI_SCRATCH_PLATFORM_ADDR_OFFSET

```
#define SBI_SCRATCH_PLATFORM_ADDR_OFFSET (6 * __SIZEOF_POINTER__)
```

Offset of platform_addr member in sbi_scratch

26.51.1.13 SBI_SCRATCH_SIZE

```
#define SBI_SCRATCH_SIZE (0x1000)
```

Maximum size of sbi_scratch (4KB)

26.51.1.14 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

26.51.1.15 sbi_scratch_thishart_offset_ptr

```
#define sbi_scratch_thishart_offset_ptr(
            offset ) ((void *)sbi_scratch_thishart_ptr() + (offset))
```

Get pointer from offset in sbi_scratch for current HART

26.51.1.16 sbi_scratch_thishart_ptr

```
#define sbi_scratch_thishart_ptr( ) ((struct sbi_scratch *)csr_read(CSR_MSCRATCH))
```

Get pointer to sbi_scratch for current HART

26.51.1.17 SBI_SCRATCH_TMP0_OFFSET

```
#define SBI_SCRATCH_TMP0_OFFSET (8 * __SIZEOF_POINTER__)
```

Offset of tmp0 member in sbi_scratch

26.51.1.18 SBI_SCRATCH_WARMBOOT_ADDR_OFFSET

```
#define SBI_SCRATCH_WARMBOOT_ADDR_OFFSET (5 * __SIZEOF_POINTER__)
```

Offset of warmboot_addr member in sbi_scratch

26.51.2 Enumeration Type Documentation

26.51.2.1 sbi_scratch_options

```
enum sbi_scratch_options
```

Possible options for OpenSBI library

Enumerator

SBI_SCRATCH_NO_BOOT_PRINTS	Disable prints during boot
SBI_SCRATCH_DEBUG_PRINTS	Enable runtime debug prints

26.51.3 Function Documentation

26.51.3.1 sbi_scratch_alloc_offset()

```
unsigned long sbi_scratch_alloc_offset (
            unsigned long size,
            const char * owner )
```

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:

26.51.3.2 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:

26.51.3.3 sbi_scratch_init()

```
int sbi_scratch_init (
            struct sbi_scratch * scratch )
```

Initialize scatch table and allocator Here is the call graph for this function:

Here is the caller graph for this function:

26.51.4 Variable Documentation

26.51.4.1 __packed

```
struct sbi_scratch __packed
```

26.51.4.2 hartid_to_scratch_table

```
struct sbi_scratch* hartid_to_scratch_table[]
```

HART id to scratch table

26.51.4.3 last_hartid_having_scratch

```
u32 last_hartid_having_scratch
```

Last HART id having a sbi_scratch pointer

26.52 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)
- 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)

26.52.1 Function Documentation

26.52.1.1 sbi_memchr()

```
void* sbi_memchr (
            const void * s,
            int c,
            size_t count )
```

26.52.1.2 sbi_memcmp()

```
int sbi_memcmp (
            const void * s1,
            const void * s2,
            size_t count )
```

26.52.1.3 sbi_memcpy()

```
void* sbi_memcpy (
            void * dest,
            const void * src,
            size_t count )
```

Here is the caller graph for this function:

26.52.1.4 sbi_memmove()

```
void* sbi_memmove (
            void * dest,
            const void * src,
            size_t count )
```

26.52.1.5 sbi_memset()

```
void* sbi_memset (
            void * s,
            int c,
            size_t count )
```

Here is the caller graph for this function:

26.52.1.6 sbi_strchr()

```
char* sbi_strchr (
            const char * s,
            int c )
```

26.52.1.7 sbi_strcmp()

```
int sbi_strcmp (
            const char * a,
            const char * b )
```

Here is the caller graph for this function:

26.52.1.8 sbi_strcpy()

```
char* sbi_strcpy (
            char * dest,
            const char * src )
```

26.52.1.9 sbi_strlen()

```
size_t sbi_strlen (
            const char * str )
```

Here is the caller graph for this function:

26.52.1.10 sbi_strncpy()

```
char* sbi_strncpy (
            char * dest,
            const char * src,
            size_t count )
```

Here is the caller graph for this function:

26.52.1.11 sbi_strnlen()

```
size_t sbi_strnlen (
            const char * str,
            size_t count )
```

26.52.1.12 sbi_strrchr()

```
char* sbi_strrchr (
            const char * s,
            int c )
```

Here is the call graph for this function:

26.53 include/sbi/sbi_system.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_system.h:

This graph shows which files directly or indirectly include this file:

Functions

- void __noreturn sbi_system_reset (u32 platform_reset_type)

26.53.1 Function Documentation

26.53.1.1 sbi_system_reset()

```
void __noreturn sbi_system_reset (
            u32 platform_reset_type )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.54 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:

Functions

- 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_set_delta_upper (ulong delta_upper)
- void sbi_timer_event_start (u64 next_event)
- void sbi_timer_process (void)
- int sbi_timer_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_timer_exit (struct sbi_scratch *scratch)

26.54.1 Function Documentation

26.54.1.1 sbi_timer_event_start()

```
void sbi_timer_event_start (
            u64 next_event )
```

Start timer event for current HART Here is the call graph for this function:

Here is the caller graph for this function:

26.54.1.2 sbi_timer_exit()

```
void sbi_timer_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.54.1.3 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:

26.54.1.4 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:

26.54.1.5 sbi_timer_process()

```
void sbi_timer_process (
            void  )
```

Process timer event for current HART Here is the caller graph for this function:

26.54.1.6 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:

26.54.1.7 sbi_timer_set_delta_upper()

```
void sbi_timer_set_delta_upper (
            ulong delta_upper )
```

Set upper 32-bits of timer delta value for current HART Here is the caller graph for this function:

26.54.1.8 sbi_timer_value()

```
u64 sbi_timer_value (
            void  )
```

Get timer value for current HART Here is the caller graph for this function:

26.54.1.9 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:

26.55 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_FIFO_NUM_ENTRIES 8
- #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_info_types {
  SBI_TLB_FLUSH_VMA, SBI_TLB_FLUSH_VMA_ASID, SBI_TLB_FLUSH_GVMA, SBI_TLB_FLUSH_GVMA_VMID, SBI_TLB_FLUSH_VVMA, SBI_TLB_FLUSH_VVMA_ASID, SBI_ITLB_FLUSH }

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)

26.55.1 Macro Definition Documentation

26.55.1.1 SBI_TLB_FIFO_NUM_ENTRIES

```
#define SBI_TLB_FIFO_NUM_ENTRIES 8
```

26.55.1.2 SBI_TLB_FLUSH_ALL

```
#define SBI_TLB_FLUSH_ALL ((unsigned long)-1)
```

26.55.1.3 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)
```

26.55.1.4 SBI_TLB_INFO_SIZE

```
#define SBI_TLB_INFO_SIZE sizeof(struct sbi_tlb_info)
```

26.55.2 Enumeration Type Documentation

26.55.2.1 sbi_tlb_info_types

```
enum sbi_tlb_info_types
```

Enumerator

SBI_TLB_FLUSH_VMA	
SBI_TLB_FLUSH_VMA_ASID	
SBI_TLB_FLUSH_GVMA	
SBI_TLB_FLUSH_GVMA_VMID	
SBI_TLB_FLUSH_VVMA	
SBI_TLB_FLUSH_VVMA_ASID	
SBI_ITLB_FLUSH	

26.55.3 Function Documentation

26.55.3.1 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:

26.55.3.2 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:

26.56 include/sbi/sbi_trap.h File Reference

```
#include <sbi/sbi_types.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

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_epc 0
- #define SBI_TRAP_INFO_cause 1
- #define SBI_TRAP_INFO_tval 2
- #define SBI_TRAP_INFO_tval2 3
- #define SBI_TRAP_INFO_tinst 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)

Functions

- int sbi_trap_redirect (struct sbi_trap_regs ∗regs, struct sbi_trap_info ∗trap)
- void sbi_trap_handler (struct sbi_trap_regs ∗regs)

Variables

- struct sbi_trap_regs __packed

26.56.1 Macro Definition Documentation

26.56.1.1 SBI_TRAP_INFO_cause

```
#define SBI_TRAP_INFO_cause 1
```

Index of cause member in sbi_trap_info

26.56.1.2 SBI_TRAP_INFO_epc

```
#define SBI_TRAP_INFO_epc 0
```

Index of epc member in sbi_trap_info

26.56.1.3 SBI_TRAP_INFO_last

```
#define SBI_TRAP_INFO_last 5
```

Last member index in sbi_trap_info

26.56.1.4 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

26.56.1.5 SBI_TRAP_INFO_SIZE

```
#define SBI_TRAP_INFO_SIZE SBI_TRAP_INFO_OFFSET(last)
```

Size (in bytes) of sbi_trap_info

26.56.1.6 SBI_TRAP_INFO_tinst

```
#define SBI_TRAP_INFO_tinst 4
```

Index of tinst member in sbi_trap_info

26.56.1.7 SBI_TRAP_INFO_tval

```
#define SBI_TRAP_INFO_tval 2
```

Index of tval member in sbi_trap_info

26.56.1.8 SBI_TRAP_INFO_tval2

```
#define SBI_TRAP_INFO_tval2 3
```

Index of tval2 member in sbi_trap_info

26.56.1.9 SBI_TRAP_REGS_a0

```
#define SBI_TRAP_REGS_a0 10
```

Index of a0 member in sbi_trap_regs

26.56.1.10 SBI_TRAP_REGS_a1

```
#define SBI_TRAP_REGS_a1 11
```

Index of a1 member in sbi_trap_regs

26.56.1.11 SBI_TRAP_REGS_a2

```
#define SBI_TRAP_REGS_a2 12
```

Index of a2 member in sbi_trap_regs

26.56.1.12 SBI_TRAP_REGS_a3

```
#define SBI_TRAP_REGS_a3 13
```

Index of a3 member in sbi_trap_regs

26.56.1.13 SBI_TRAP_REGS_a4

```
#define SBI_TRAP_REGS_a4 14
```

Index of a4 member in sbi_trap_regs

26.56.1.14 SBI_TRAP_REGS_a5

```
#define SBI_TRAP_REGS_a5 15
```

Index of a5 member in sbi_trap_regs

26.56.1.15 SBI_TRAP_REGS_a6

```
#define SBI_TRAP_REGS_a6 16
```

Index of a6 member in sbi_trap_regs

26.56.1.16 SBI_TRAP_REGS_a7

```
#define SBI_TRAP_REGS_a7 17
```

Index of a7 member in sbi_trap_regs

26.56.1.17 SBI_TRAP_REGS_gp

```
#define SBI_TRAP_REGS_gp 3
```

Index of gp member in sbi_trap_regs

26.56.1.18 SBI_TRAP_REGS_last

```
#define SBI_TRAP_REGS_last 35
```

Last member index in sbi_trap_regs

26.56.1.19 SBI_TRAP_REGS_mepc

```
#define SBI_TRAP_REGS_mepc 32
```

Index of mepc member in sbi_trap_regs

26.56.1.20 SBI_TRAP_REGS_mstatus

```
#define SBI_TRAP_REGS_mstatus 33
```

Index of mstatus member in sbi_trap_regs

26.56.1.21 SBI_TRAP_REGS_mstatusH

```
#define SBI_TRAP_REGS_mstatusH 34
```

Index of mstatusH member in sbi_trap_regs

26.56.1.22 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

26.56.1.23 SBI_TRAP_REGS_ra

```
#define SBI_TRAP_REGS_ra 1
```

Index of ra member in sbi_trap_regs

26.56.1.24 SBI_TRAP_REGS_s0

```
#define SBI_TRAP_REGS_s0 8
```

Index of s0 member in sbi_trap_regs

26.56.1.25 SBI_TRAP_REGS_s1

```
#define SBI_TRAP_REGS_s1 9
```

Index of s1 member in sbi_trap_regs

26.56.1.26 SBI_TRAP_REGS_s10

```
#define SBI_TRAP_REGS_s10 26
```

Index of s10 member in sbi_trap_regs

26.56.1.27 SBI_TRAP_REGS_s11

```
#define SBI_TRAP_REGS_s11 27
```

Index of s11 member in sbi_trap_regs

26.56.1.28 SBI_TRAP_REGS_s2

```
#define SBI_TRAP_REGS_s2 18
```

Index of s2 member in sbi_trap_regs

26.56.1.29 SBI_TRAP_REGS_s3

```
#define SBI_TRAP_REGS_s3 19
```

Index of s3 member in sbi_trap_regs

26.56.1.30 SBI_TRAP_REGS_s4

```
#define SBI_TRAP_REGS_s4 20
```

Index of s4 member in sbi_trap_regs

26.56.1.31 SBI_TRAP_REGS_s5

```
#define SBI_TRAP_REGS_s5 21
```

Index of s5 member in sbi_trap_regs

26.56.1.32 SBI_TRAP_REGS_s6

```
#define SBI_TRAP_REGS_s6 22
```

Index of s6 member in sbi_trap_regs

26.56.1.33 SBI_TRAP_REGS_s7

```
#define SBI_TRAP_REGS_s7 23
```

Index of s7 member in sbi_trap_regs

26.56.1.34 SBI_TRAP_REGS_s8

```
#define SBI_TRAP_REGS_s8 24
```

Index of s8 member in sbi_trap_regs

26.56.1.35 SBI_TRAP_REGS_s9

```
#define SBI_TRAP_REGS_s9 25
```

Index of s9 member in sbi_trap_regs

26.56.1.36 SBI_TRAP_REGS_SIZE

```
#define SBI_TRAP_REGS_SIZE SBI_TRAP_REGS_OFFSET(last)
```

Size (in bytes) of sbi_trap_regs

26.56.1.37 SBI_TRAP_REGS_sp

```
#define SBI_TRAP_REGS_sp 2
```

Index of sp member in sbi_trap_regs

26.56.1.38 SBI_TRAP_REGS_t0

```
#define SBI_TRAP_REGS_t0 5
```

Index of t0 member in sbi_trap_regs

26.56.1.39 SBI_TRAP_REGS_t1

```
#define SBI_TRAP_REGS_t1 6
```

Index of t1 member in sbi_trap_regs

26.56.1.40 SBI_TRAP_REGS_t2

```
#define SBI_TRAP_REGS_t2 7
```

Index of t2 member in sbi_trap_regs

26.56.1.41 SBI_TRAP_REGS_t3

```
#define SBI_TRAP_REGS_t3 28
```

Index of t3 member in sbi_trap_regs

26.56.1.42 SBI_TRAP_REGS_t4

```
#define SBI_TRAP_REGS_t4 29
```

Index of t4 member in sbi_trap_regs

26.56.1.43 SBI_TRAP_REGS_t5

```
#define SBI_TRAP_REGS_t5 30
```

Index of t5 member in sbi_trap_regs

26.56.1.44 SBI_TRAP_REGS_t6

```
#define SBI_TRAP_REGS_t6 31
```

Index of t6 member in sbi_trap_regs

26.56.1.45 SBI_TRAP_REGS_tp

```
#define SBI_TRAP_REGS_tp 4
```

Index of tp member in sbi_trap_regs

26.56.1.46 SBI_TRAP_REGS_zero

```
#define SBI_TRAP_REGS_zero 0
```

Index of zero member in sbi_trap_regs

26.56.2 Function Documentation

26.56.2.1 sbi_trap_handler()

```
void sbi_trap_handler (
            struct sbi_trap_regs * regs )
```

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

regs	pointer to register state

Here is the call graph for this function:

26.56.2.2 sbi_trap_redirect()

```
int sbi_trap_redirect (
            struct sbi_trap_regs * regs,
            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 caller graph for this function:

26.56.3 Variable Documentation

26.56.3.1 __packed

```
struct sbi_trap_regs __packed
```

26.57 include/sbi/sbi_types.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define TRUE 1
- #define FALSE 0
- #define true TRUE
- #define false FALSE
- #define NULL ((void ∗)0)
- #define __packed __attribute__((packed))
- #define __noreturn __attribute__((noreturn))
- #define likely(x) __builtin_expect((x), 1)
- #define unlikely(x) __builtin_expect((x), 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 int 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

26.57.1 Macro Definition Documentation

26.57.1.1 __noreturn

```
#define __noreturn __attribute__((noreturn))
```

26.57.1.2 __packed

```
struct sbi_platform __packed __attribute__((packed))
```

26.57.1.3 array_size

```
#define array_size(
            x ) (sizeof(x) / sizeof((x)[0]))
```

26.57.1.4 CLAMP

```
#define CLAMP(
            a,
            lo,
            hi ) MIN(MAX(a, lo), hi)
```

26.57.1.5 container_of

```
#define container_of(
            ptr,
            type,
            member )
```

Value:

```
({          \
const typeof(((type *)0)->member) * __mptr = (ptr); \
(type *)((char *)__mptr - offsetof(type, member)); })
```

26.57.1.6 FALSE

```
#define FALSE 0
```

26.57.1.7 false

```
#define false FALSE
```

26.57.1.8 likely

```
#define likely(
            x ) __builtin_expect((x), 1)
```

26.57.1.9 MAX

```
#define MAX(
            a,
            b ) ((a) > (b) ?  (a) :  (b))
```

26.57.1.10 MIN

```
#define MIN(
            a,
            b ) ((a) < (b) ? (a) : (b))
```

26.57.1.11 NULL

```
#define NULL ((void *)0)
```

26.57.1.12 offsetof

```
#define offsetof(
            TYPE,
            MEMBER ) ((size_t) &((TYPE *)0)->MEMBER)
```

26.57.1.13 ROUNDDOWN

```
#define ROUNDDOWN(
            a,
            b ) ((a) / (b) * (b))
```

26.57.1.14 ROUNDUP

```
#define ROUNDUP(
            a,
            b ) ((((a)-1) / (b) + 1) * (b))
```

26.57.1.15 STR

```
#define STR(
            x ) XSTR(x)
```

26.57.1.16 TRUE

```
#define TRUE 1
```

26.57.1.17 true

```
#define true TRUE
```

26.57.1.18 unlikely

```
#define unlikely(
            x ) __builtin_expect((x), 0)
```

26.57.1.19 XSTR

```
#define XSTR(
            x ) #x
```

26.57.2 Typedef Documentation

26.57.2.1 bool

```
typedef int bool
```

26.57.2.2 int16_t

```
typedef short int16_t
```

26.57.2.3 int32_t

```
typedef int int32_t
```

26.57.2.4 physical_addr_t

```
typedef unsigned long physical_addr_t
```

26.57.2.5 physical_size_t

```
typedef unsigned long physical_size_t
```

26.57.2.6 s16

```
typedef short s16
```

26.57.2.7 s32

```
typedef int s32
```

26.57.2.8 s8

```
typedef char s8
```

26.57.2.9 size_t

```
typedef unsigned long size_t
```

26.57.2.10 ssize_t

```
typedef long ssize_t
```

26.57.2.11 u16

```
typedef unsigned short u16
```

26.57.2.12 u32

```
typedef unsigned int u32
```

26.57.2.13 u8

```
typedef unsigned char u8
```

26.57.2.14 uint16_t

```
typedef unsigned short uint16_t
```

26.57.2.15 uint32_t

```
typedef unsigned int uint32_t
```

26.57.2.16 uint8_t

```
typedef unsigned char uint8_t
```

26.57.2.17 uintptr_t

```
typedef unsigned long uintptr_t
```

26.57.2.18 ulong

```
typedef unsigned long ulong
```

26.57.2.19 virtual_addr_t

```
typedef unsigned long virtual_addr_t
```

26.57.2.20 virtual_size_t

```
typedef unsigned long virtual_size_t
```

26.58 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)

26.58.1 Macro Definition Documentation

26.58.1.1 DECLARE_UNPRIVILEGED_LOAD_FUNCTION

```
#define DECLARE_UNPRIVILEGED_LOAD_FUNCTION(
            type )
```

Value:

```
type sbi_load_##type(const type *addr,            \
            struct sbi_trap_info *trap);
```

26.58.1.2 DECLARE_UNPRIVILEGED_STORE_FUNCTION

```
#define DECLARE_UNPRIVILEGED_STORE_FUNCTION(
            type )
```

Value:

```
void sbi_store_##type(type *addr, type val,       \
            struct sbi_trap_info *trap);
```

26.58.2 Function Documentation

26.58.2.1 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:

26.59 include/sbi/sbi_version.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define OPENSBI_VERSION_MAJOR 0
- #define OPENSBI_VERSION_MINOR 8
- #define OPENSBI_VERSION

26.59.1 Macro Definition Documentation

26.59.1.1 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

26.59.1.2 OPENSBI_VERSION_MAJOR

```
#define OPENSBI_VERSION_MAJOR 0
```

26.59.1.3 OPENSBI_VERSION_MINOR

```
#define OPENSBI_VERSION_MINOR 8
```

26.60 include/sbi_utils/fdt/fdt_fixup.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void fdt_cpu_fixup (void ∗fdt)
- void fdt_plic_fixup (void ∗fdt, const char ∗compat)
- int fdt_reserved_memory_fixup (void ∗fdt)
- int fdt_reserved_memory_nomap_fixup (void ∗fdt)
- void fdt_fixups (void ∗fdt)

26.60.1 Function Documentation

26.60.1.1 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:

26.60.1.2 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 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:

26.60.1.3 fdt_plic_fixup()

```
void fdt_plic_fixup (
```

```
void * fdt,
const char * compat )
```

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
compat	PLIC node compatible string

Here is the call graph for this function:

Here is the caller graph for this function:

26.60.1.4 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:

26.60.1.5 fdt_reserved_memory_nomap_fixup()

```
int fdt_reserved_memory_nomap_fixup (
            void * fdt )
```

Fix up the reserved memory subnodes in the device tree
This routine adds the no-map property to the reserved memory subnodes so that the OS does not map those PMP protected memory regions.
Platform codes must call this helper in their final_init() after fdt_fixups() if the OS should not map the PMP protected reserved regions.

Parameters

fdt	device tree blob

Returns

zero on success and -ve on failure

Here is the call graph for this function:

26.61 include/sbi_utils/fdt/fdt_helper.h File Reference

```
#include <sbi/sbi_types.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 platform_uart_data

Functions

- const struct fdt_match ∗ fdt_match_node (void ∗fdt, int nodeoff, const struct fdt_match ∗match_table)
- int fdt_find_match (void ∗fdt, int startoff, const struct fdt_match ∗match_table, const struct fdt_match ∗∗out↩ _match)
- int fdt_get_node_addr_size (void ∗fdt, int node, unsigned long ∗addr, unsigned long ∗size)
- int fdt_parse_hart_id (void ∗fdt, int cpu_offset, u32 ∗hartid)
- int fdt_parse_max_hart_id (void ∗fdt, u32 ∗max_hartid)
- int fdt_parse_shakti_uart_node (void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_sifive_uart_node (void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_uart8250_node (void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)

- int fdt_parse_uart8250 (void ∗fdt, struct platform_uart_data ∗uart, const char ∗compatible)
- int fdt_parse_plic_node (void ∗fdt, int nodeoffset, struct plic_data ∗plic)
- int fdt_parse_plic (void ∗fdt, struct plic_data ∗plic, const char ∗compat)
- int fdt_parse_clint_node (void ∗fdt, int nodeoffset, bool for_timer, struct clint_data ∗clint)
- int fdt_parse_compat_addr (void ∗fdt, unsigned long ∗addr, const char ∗compatible)

26.61.1 Function Documentation

26.61.1.1 fdt_find_match()

```
int fdt_find_match (
            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:

26.61.1.2 fdt_get_node_addr_size()

```
int fdt_get_node_addr_size (
            void * fdt,
            int node,
            unsigned long * addr,
            unsigned long * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.61.1.3 fdt_match_node()

```
const struct fdt_match* fdt_match_node (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match_table )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.61.1.4 fdt_parse_clint_node()

```
int fdt_parse_clint_node (
            void * fdt,
            int nodeoffset,
            bool for_timer,
            struct clint_data * clint )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.61.1.5 fdt_parse_compat_addr()

```
int fdt_parse_compat_addr (
            void * fdt,
            unsigned long * addr,
            const char * compatible )
```

Here is the call graph for this function:

26.61.1.6 fdt_parse_hart_id()

```
int fdt_parse_hart_id (
            void * fdt,
            int cpu_offset,
            u32 * hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.61.1.7 fdt_parse_max_hart_id()

```
int fdt_parse_max_hart_id (
            void * fdt,
            u32 * max_hartid )
```

Here is the call graph for this function:

26.61.1.8 fdt_parse_plic()

```
int fdt_parse_plic (
            void * fdt,
            struct plic_data * plic,
            const char * compat )
```

Here is the call graph for this function:

26.61.1.9 fdt_parse_plic_node()

```
int fdt_parse_plic_node (
            void * fdt,
            int nodeoffset,
            struct plic_data * plic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.61.1.10 fdt_parse_shakti_uart_node()

```
int fdt_parse_shakti_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

UART address is mandaotry. 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:

26.61.1.11 fdt_parse_sifive_uart_node()

```
int fdt_parse_sifive_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

UART address is mandaotry. 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:

26.61.1.12 fdt_parse_uart8250()

```
int fdt_parse_uart8250 (
            void * fdt,
            struct platform_uart_data * uart,
            const char * compatible )
```

Here is the call graph for this function:

26.61.1.13 fdt_parse_uart8250_node()

```
int fdt_parse_uart8250_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

UART address is mandaotry. 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:

26.62 include/sbi_utils/ipi/fdt_ipi.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_ipi.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_ipi

Functions

- void fdt_ipi_send (u32 target_hart)
- void fdt_ipi_clear (u32 target_hart)
- void fdt_ipi_exit (void)
- int fdt_ipi_init (bool cold_boot)

26.62.1 Function Documentation

26.62.1.1 fdt_ipi_clear()

```
void fdt_ipi_clear (
            u32 target_hart )
```

26.62.1.2 fdt_ipi_exit()

```
void fdt_ipi_exit (
            void  )
```

26.62.1.3 fdt_ipi_init()

```
int fdt_ipi_init (
            bool cold_boot )
```

Here is the call graph for this function:

26.62.1.4 fdt_ipi_send()

```
void fdt_ipi_send (
            u32 target_hart )
```

26.63 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:

Data Structures

- struct fdt_irqchip

Functions

- void fdt_irqchip_exit (void)
- int fdt_irqchip_init (bool cold_boot)

26.63.1 Function Documentation

26.63.1.1 fdt_irqchip_exit()

```
void fdt_irqchip_exit (
            void  )
```

26.63.1.2 fdt_irqchip_init()

```
int fdt_irqchip_init (
            bool cold_boot )
```

Here is the call graph for this function:

26.64 include/sbi_utils/irqchip/plic.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for plic.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct plic_data

Functions

- int plic_warm_irqchip_init (struct plic_data *plic, int m_cntx_id, int s_cntx_id)
- int plic_cold_irqchip_init (struct plic_data *plic)
- void plic_set_thresh (struct plic_data *plic, u32 cntxid, u32 val)
- void plic_set_ie (struct plic_data *plic, u32 cntxid, u32 word_index, u32 val)

26.64.1 Function Documentation

26.64.1.1 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:

26.64.1.2 plic_set_ie()

```
void plic_set_ie (
            struct plic_data * plic,
            u32 cntxid,
            u32 word_index,
            u32 val )
```

Here is the caller graph for this function:

26.64.1.3 plic_set_thresh()

```
void plic_set_thresh (
            struct plic_data * plic,
            u32 cntxid,
            u32 val )
```

Here is the caller graph for this function:

26.64.1.4 plic_warm_irqchip_init()

```
int plic_warm_irqchip_init (
            struct plic_data * plic,
            int m_cntx_id,
            int s_cntx_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.65 include/sbi_utils/reset/fdt_reset.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_reset.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_reset

Functions

- int fdt_system_reset (u32 reset_type)
- int fdt_reset_init (void)

26.65.1 Function Documentation

26.65.1.1 fdt_reset_init()

```
int fdt_reset_init (
            void  )
```

Here is the call graph for this function:

26.65.1.2 fdt_system_reset()

```
int fdt_system_reset (
            u32 reset_type )
```

26.66 include/sbi_utils/serial/fdt_serial.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_serial.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_serial

Functions

- void fdt_serial_putc (char ch)
- int fdt_serial_getc (void)
- int fdt_serial_init (void)

26.66.1 Function Documentation

26.66.1.1 fdt_serial_getc()

```
int fdt_serial_getc (
            void  )
```

26.66.1.2 fdt_serial_init()

```
int fdt_serial_init (
            void  )
```

Here is the call graph for this function:

26.66.1.3 fdt_serial_putc()

```
void fdt_serial_putc (
            char ch )
```

26.67 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

- void shakti_uart_putc (char ch)
- int shakti_uart_getc (void)
- int shakti_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

26.67.1 Function Documentation

26.67.1.1 shakti_uart_getc()

```
int shakti_uart_getc (
            void  )
```

26.67.1.2 shakti_uart_init()

```
int shakti_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the caller graph for this function:

26.67.1.3 shakti_uart_putc()

```
void shakti_uart_putc (
            char ch )
```

26.68 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

- void sifive_uart_putc (char ch)
- int sifive_uart_getc (void)
- int sifive_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

26.68.1 Function Documentation

26.68.1.1 sifive_uart_getc()

```
int sifive_uart_getc (
            void  )
```

Here is the call graph for this function:

26.68.1.2 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:

26.68.1.3 sifive_uart_putc()

```
void sifive_uart_putc (
            char ch )
```

Here is the call graph for this function:

26.69 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

- void uart8250_putc (char ch)
- int uart8250_getc (void)
- int uart8250_init (unsigned long base, u32 in_freq, u32 baudrate, u32 reg_shift, u32 reg_width)

26.69.1 Function Documentation

26.69.1.1 uart8250_getc()

```
int uart8250_getc (
            void  )
```

Here is the call graph for this function:

26.69.1.2 uart8250_init()

```
int uart8250_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate,
            u32 reg_shift,
            u32 reg_width )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.69.1.3 uart8250_putc()

```
void uart8250_putc (
            char ch )
```

Here is the call graph for this function:

26.70 include/sbi_utils/sys/clint.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for clint.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct clint_data

Functions

- void clint_ipi_send (u32 target_hart)
- void clint_ipi_clear (u32 target_hart)
- int clint_warm_ipi_init (void)
- int clint_cold_ipi_init (struct clint_data ∗clint)
- u64 clint_timer_value (void)
- void clint_timer_event_stop (void)
- void clint_timer_event_start (u64 next_event)
- int clint_warm_timer_init (void)
- int clint_cold_timer_init (struct clint_data ∗clint, struct clint_data ∗reference)

26.70.1 Function Documentation

26.70.1.1 clint_cold_ipi_init()

```
int clint_cold_ipi_init (
            struct clint_data * clint )
```

Here is the caller graph for this function:

26.70.1.2 clint_cold_timer_init()

```
int clint_cold_timer_init (
            struct clint_data * clint,
            struct clint_data * reference )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.70.1.3 clint_ipi_clear()

```
void clint_ipi_clear (
            u32 target_hart )
```

Here is the caller graph for this function:

26.70.1.4 clint_ipi_send()

```
void clint_ipi_send (
            u32 target_hart )
```

26.70.1.5 clint_timer_event_start()

```
void clint_timer_event_start (
            u64 next_event )
```

26.70.1.6 clint_timer_event_stop()

```
void clint_timer_event_stop (
            void  )
```

26.70.1.7 clint_timer_value()

```
u64 clint_timer_value (
            void  )
```

26.70.1.8 clint_warm_ipi_init()

```
int clint_warm_ipi_init (
            void  )
```

Here is the call graph for this function:

26.70.1.9 clint_warm_timer_init()

```
int clint_warm_timer_init (
            void  )
```

Here is the call graph for this function:

26.71 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

- void htif_putc (char ch)
- int htif_getc (void)
- int htif_system_reset (u32 type)

26.71.1 Function Documentation

26.71.1.1 htif_getc()

```
int htif_getc (
            void  )
```

Here is the call graph for this function:

26.71.1.2 htif_putc()

```
void htif_putc (
            char ch )
```

Here is the call graph for this function:

26.71.1.3 htif_system_reset()

```
int htif_system_reset (
            u32 type )
```

26.72 include/sbi_utils/sys/sifive_test.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sifive_test.h:

This graph shows which files directly or indirectly include this file:

Functions

- int sifive_test_system_reset (u32 type)
- int sifive_test_init (unsigned long base)

26.72.1 Function Documentation

26.72.1.1 sifive_test_init()

```
int sifive_test_init (
            unsigned long base )
```

Here is the caller graph for this function:

26.72.1.2 sifive_test_system_reset()

```
int sifive_test_system_reset (
            u32 type )
```

26.73 include/sbi_utils/timer/fdt_timer.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for fdt_timer.h:

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_timer

Functions

- u64 fdt_timer_value (void)
- void fdt_timer_event_stop (void)
- void fdt_timer_event_start (u64 next_event)
- void fdt_timer_exit (void)
- int fdt_timer_init (bool cold_boot)

26.73.1 Function Documentation

26.73.1.1 fdt_timer_event_start()

```
void fdt_timer_event_start (
            u64 next_event )
```

26.73.1.2 fdt_timer_event_stop()

```
void fdt_timer_event_stop (
            void )
```

26.73.1.3 fdt_timer_exit()

```
void fdt_timer_exit (
            void )
```

26.73.1.4 fdt_timer_init()

```
int fdt_timer_init (
            bool cold_boot )
```

Here is the call graph for this function:

26.73.1.5 fdt_timer_value()

```
u64 fdt_timer_value (
            void )
```

26.74 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 dependency graph for riscv_asm.c:

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_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 ∗size)

26.74.1 Function Documentation

26.74.1.1 csr_read_num()

```
unsigned long csr_read_num (
            int csr_num )
```

Here is the caller graph for this function:

26.74.1.2 csr_write_num()

```
void csr_write_num (
            int csr_num,
            unsigned long val )
```

Here is the caller graph for this function:

26.74.1.3 ctz()

```
static unsigned long ctz (
            unsigned long x )  [static]
```

Here is the caller graph for this function:

26.74.1.4 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:

26.74.1.5 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:

26.74.1.6 misa_xlen()

```
int misa_xlen (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.74.1.7 pmp_get()

```
int pmp_get (
            unsigned int n,
```

```
            unsigned long * prot_out,
            unsigned long * addr_out,
            unsigned long * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.74.1.8 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:

26.75 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)
- #define __xchg(ptr, new, size)
- #define xchg(ptr, n)
- #define __cmpxchg(ptr, old, new, size)
- #define cmpxchg(ptr, o, n)
- #define __atomic_op_bit_ord(op, mod, nr, addr, ord)
- #define __atomic_op_bit(op, mod, nr, addr) __atomic_op_bit_ord(op, mod, nr, addr, .aqrl)
- #define __NOP(x) (x)
- #define __NOT(x) (~(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)

26.75.1 Macro Definition Documentation

26.75.1.1 __atomic_op_bit

```
#define __atomic_op_bit(
            op,
            mod,
            nr,
            addr ) __atomic_op_bit_ord(op, mod, nr, addr, .aqrl)
```

26.75.1.2 __atomic_op_bit_ord

```
#define __atomic_op_bit_ord(
            op,
            mod,
            nr,
            addr,
            ord )
```

Value:

```
    ({                                                                  \
        unsigned long __res, __mask;                                    \
        __mask = BIT_MASK(nr);                                          \
        __asm__ __volatile__(__AMO(op) #ord " %0, %2, %1"               \
                    : "=r"(__res), "+A"(addr[BIT_WORD(nr)]) \
                    : "r"(mod(__mask))                      \
                    : "memory");                            \
        __res;                                                          \
    })
```

26.75.1.3 __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;                                  \
    })
```

26.75.1.4 __cmpxchg

```
#define __cmpxchg(
                ptr,
                old,
                new,
                size )
```

26.75.1.5 __NOP

```
#define __NOP(
                x ) (x)
```

26.75.1.6 __NOT

```
#define __NOT(
                x ) (~(x))
```

26.75.1.7 __xchg

```
#define __xchg(
                ptr,
                new,
                size )
```

26.75.1.8 axchg

```
#define axchg(
                ptr,
                x )
```

Value:

```
    ({                                          \
        __typeof__(*(ptr)) _x_ = (x);                       \
        (__typeof__(*(ptr))) __axchg((ptr), _x_, sizeof(*(ptr)));    \
    })
```

26.75.1.9 cmpxchg

```
#define cmpxchg(
                ptr,
                o,
                n )
```

Value:

```
    ({                                                          \
        __typeof__(*(ptr)) _o_ = (o);                       \
        __typeof__(*(ptr)) _n_ = (n);                       \
        (__typeof__(*(ptr)))                                \
            __cmpxchg((ptr), _o_, _n_, sizeof(*(ptr))); \
    })
```

26.75.1.10 xchg

```
#define xchg(
            ptr,
            n )
```

Value:

```
({                                                          \
    __typeof__(*(ptr)) _n_ = (n);                           \
    (__typeof__(*(ptr))) __xchg((ptr), _n_, sizeof(*(ptr))); \
})
```

26.75.2 Function Documentation

26.75.2.1 atomic_add_return()

```
long atomic_add_return (
            atomic_t * atom,
            long value )
```

26.75.2.2 atomic_clear_bit()

```
int atomic_clear_bit (
            int nr,
            atomic_t * atom )  [inline]
```

Clear a bit in an atomic variable and return the new value. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:

26.75.2.3 atomic_cmpxchg()

```
long atomic_cmpxchg (
            atomic_t * atom,
            long oldval,
            long newval )
```

Here is the caller graph for this function:

26.75.2.4 atomic_raw_clear_bit()

```
int atomic_raw_clear_bit (
            int nr,
            volatile unsigned long * addr )  [inline]
```

Clear a bit in any address and return the new value . @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:

26.75.2.5 atomic_raw_set_bit()

```
int atomic_raw_set_bit (
            int nr,
            volatile unsigned long * addr )  [inline]
```

Set a bit in any address and return the new value . @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:

26.75.2.6 atomic_raw_xchg_uint()

```
unsigned int atomic_raw_xchg_uint (
            volatile unsigned int * ptr,
            unsigned int newval )
```

26.75.2.7 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:

26.75.2.8 atomic_read()

```
long atomic_read (
            atomic_t * atom )
```

Here is the caller graph for this function:

26.75.2.9 atomic_set_bit()

```
int atomic_set_bit (
            int nr,
            atomic_t * atom )  [inline]
```

Set a bit in an atomic variable and return the new value. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:

26.75.2.10 atomic_sub_return()

```
long atomic_sub_return (
            atomic_t * atom,
```

```
                long value )
```

26.75.2.11 atomic_write()

```
void atomic_write (
                atomic_t * atom,
                long value )
```

26.75.2.12 atomic_xchg()

```
long atomic_xchg (
                atomic_t * atom,
                long newval )
```

Here is the caller graph for this function:

26.76 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

- int spin_lock_check (spinlock_t *lock)
- int spin_trylock (spinlock_t *lock)
- void spin_lock (spinlock_t *lock)
- void spin_unlock (spinlock_t *lock)

26.76.1 Function Documentation

26.76.1.1 spin_lock()

```
void spin_lock (
            spinlock_t * lock )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.76.1.2 spin_lock_check()

```
int spin_lock_check (
            spinlock_t * lock )
```

Here is the caller graph for this function:

26.76.1.3 spin_trylock()

```
int spin_trylock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

26.76.1.4 spin_unlock()

```
void spin_unlock (
            spinlock_t * lock )
```

Here is the caller graph for this function:

26.77 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)

26.77.1 Function Documentation

26.77.1.1 __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:

26.77.1.2 __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:

26.77.1.3 __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:

26.78 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)

26.78.1 Macro Definition Documentation

26.78.1.1 BITOP_WORD

```
#define BITOP_WORD(
            nr ) ((nr) / BITS_PER_LONG)
```

26.78.2 Function Documentation

26.78.2.1 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:

26.78.2.2 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.

26.78.2.3 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:

26.78.2.4 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:

26.78.2.5 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

26.79 lib/sbi/sbi_console.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
```

Include dependency graph for sbi_console.c:

Macros

- #define PAD_RIGHT 1
- #define PAD_ZERO 2
- #define PAD_ALTERNATE 4

- #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

- bool sbi_isprintable (char c)
- int sbi_getc (void)
- void sbi_putc (char ch)
- void sbi_puts (const char *str)
- void sbi_gets (char *s, int maxwidth, char endchar)
- static void printc (char **out, u32 *out_len, char ch)
- 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 b, int sg, int width, int flags, int letbase)
- 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,...)
- int sbi_console_init (struct sbi_scratch *scratch)

Variables

- static const struct sbi_platform * console_plat = NULL
- static spinlock_t console_out_lock = SPIN_LOCK_INITIALIZER

26.79.1 Macro Definition Documentation

26.79.1.1 PAD_ALTERNATE

```
#define PAD_ALTERNATE 4
```

26.79.1.2 PAD_RIGHT

```
#define PAD_RIGHT 1
```

26.79.1.3 PAD_ZERO

```
#define PAD_ZERO 2
```

26.79.1.4 PRINT_BUF_LEN

```
#define PRINT_BUF_LEN 64
```

26.79.1.5 va_arg

```
#define va_arg __builtin_va_arg
```

26.79.1.6 va_end

```
#define va_end __builtin_va_end
```

26.79.1.7 va_start

```
#define va_start(
            v,
            l ) __builtin_va_start((v), l)
```

26.79.2 Typedef Documentation

26.79.2.1 va_list

```
typedef __builtin_va_list va_list
```

26.79.3 Function Documentation

26.79.3.1 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:

26.79.3.2 printc()

```
static void printc (
            char ** out,
            u32 * out_len,
            char ch )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.79.3.3 printi()

```
static int printi (
            char ** out,
            u32 * out_len,
            long long i,
            int b,
            int sg,
            int width,
            int flags,
            int letbase )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.79.3.4 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:

26.79.3.5 sbi_console_init()

```
int sbi_console_init (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.79.3.6 sbi_dprintf()

```
int sbi_dprintf (
            const char * format,
             ...  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.79.3.7 sbi_getc()

```
int sbi_getc (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.79.3.8 sbi_gets()

```
void sbi_gets (
            char * s,
            int maxwidth,
            char endchar )
```

Here is the call graph for this function:

26.79.3.9 sbi_isprintable()

```
bool sbi_isprintable (
            char c )
```

26.79.3.10 sbi_printf()

```
int sbi_printf (
            const char * format,
             ...  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.79.3.11 sbi_putc()

```
void sbi_putc (
            char ch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.79.3.12 sbi_puts()

```
void sbi_puts (
            const char * str )
```

Here is the call graph for this function:

26.79.3.13 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:

26.79.3.14 sbi_sprintf()

```
int sbi_sprintf (
            char * out,
            const char * format,
             ...  )
```

Here is the call graph for this function:

26.79.4 Variable Documentation

26.79.4.1 console_out_lock

```
spinlock_t console_out_lock = SPIN_LOCK_INITIALIZER  [static]
```

26.79.4.2 console_plat

```
const struct sbi_platform* console_plat = NULL  [static]
```

26.80 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_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)
- 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_regs ∗regs)
- int sbi_ecall_init (void)

Variables

- static unsigned long ecall_impid = SBI_OPENSBI_IMPID

26.80.1 Function Documentation

26.80.1.1 sbi_ecall_find_extension()

```
struct sbi_ecall_extension* sbi_ecall_find_extension (
            unsigned long extid )
```

Here is the caller graph for this function:

26.80.1.2 sbi_ecall_get_impid()

```
unsigned long sbi_ecall_get_impid (
            void  )
```

Here is the caller graph for this function:

26.80.1.3 sbi_ecall_handler()

```
int sbi_ecall_handler (
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.80.1.4 sbi_ecall_init()

```
int sbi_ecall_init (
            void  )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.80.1.5 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:

26.80.1.6 sbi_ecall_set_impid()

```
void sbi_ecall_set_impid (
            unsigned long impid )
```

26.80.1.7 sbi_ecall_unregister_extension()

```
void sbi_ecall_unregister_extension (
            struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:

26.80.1.8 sbi_ecall_version_major()

```
u16 sbi_ecall_version_major (
            void  )
```

Here is the caller graph for this function:

26.80.1.9 sbi_ecall_version_minor()

```
u16 sbi_ecall_version_minor (
            void  )
```

Here is the caller graph for this function:

26.80.1.10 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            ecall_exts_list  )  [static]
```

26.80.2 Variable Documentation

26.80.2.1 ecall_impid

```
unsigned long ecall_impid = SBI_OPENSBI_IMPID  [static]
```

26.81 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_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, unsigned long *args, unsigned long *out_val, struct sbi_trap_info *out_trap)

Variables

- struct sbi_ecall_extension ecall_base

26.81.1 Function Documentation

26.81.1.1 sbi_ecall_base_handler()

```
static int sbi_ecall_base_handler (
            unsigned long extid,
            unsigned long funcid,
            unsigned long * args,
            unsigned long * out_val,
            struct sbi_trap_info * out_trap )  [static]
```

Here is the call graph for this function:

26.81.1.2 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:

26.81.2 Variable Documentation

26.81.2.1 ecall_base

```
struct sbi_ecall_extension ecall_base
```

Initial value:

```
= {
    .extid_start = SBI_EXT_BASE,
    .extid_end = SBI_EXT_BASE,
    .handle = sbi_ecall_base_handler,
}
```

26.82 lib/sbi/sbi_ecall_hsm.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_version.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, unsigned long ∗args, unsigned long ∗out_val, struct sbi_trap_info ∗out_trap)

Variables

- struct sbi_ecall_extension ecall_hsm

26.82.1 Function Documentation

26.82.1.1 sbi_ecall_hsm_handler()

```
static int sbi_ecall_hsm_handler (
            unsigned long extid,
            unsigned long funcid,
            unsigned long * args,
            unsigned long * out_val,
            struct sbi_trap_info * out_trap )  [static]
```

Here is the call graph for this function:

26.82.2 Variable Documentation

26.82.2.1 ecall_hsm

```
struct sbi_ecall_extension ecall_hsm
```

Initial value:

```
= {
    .extid_start = SBI_EXT_HSM,
    .extid_end = SBI_EXT_HSM,
    .handle = sbi_ecall_hsm_handler,
}
```

26.83 lib/sbi/sbi_ecall_legacy.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hsm.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 int sbi_load_hart_mask_unpriv (ulong ∗pmask, ulong ∗hmask, struct sbi_trap_info ∗uptrap)
- static int sbi_ecall_legacy_handler (unsigned long extid, unsigned long funcid, unsigned long ∗args, unsigned long ∗out_val, struct sbi_trap_info ∗out_trap)

Variables

- struct sbi_ecall_extension ecall_legacy

26.83.1 Function Documentation

26.83.1.1 sbi_ecall_legacy_handler()

```
static int sbi_ecall_legacy_handler (
            unsigned long extid,
            unsigned long funcid,
            unsigned long * args,
```

```
unsigned long * out_val,
struct sbi_trap_info * out_trap )  [static]
```

Here is the call graph for this function:

26.83.1.2 sbi_load_hart_mask_unpriv()

```
static int sbi_load_hart_mask_unpriv (
            ulong * pmask,
            ulong * hmask,
            struct sbi_trap_info * uptrap )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.83.2 Variable Documentation

26.83.2.1 ecall_legacy

```
struct sbi_ecall_extension ecall_legacy
```

Initial value:

```
= {
    .extid_start = SBI_EXT_0_1_SET_TIMER,
    .extid_end = SBI_EXT_0_1_SHUTDOWN,
    .handle = sbi_ecall_legacy_handler,
}
```

26.84 lib/sbi/sbi_ecall_replace.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_tlb.h>
```

Include dependency graph for sbi_ecall_replace.c:

Functions

- static int sbi_ecall_time_handler (unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct sbi_trap_info *out_trap)
- static int sbi_ecall_rfence_handler (unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct sbi_trap_info *out_trap)
- static int sbi_ecall_ipi_handler (unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct sbi_trap_info *out_trap)

Variables

- struct sbi_ecall_extension ecall_time
- struct sbi_ecall_extension ecall_rfence
- struct sbi_ecall_extension ecall_ipi

26.84.1 Function Documentation

26.84.1.1 sbi_ecall_ipi_handler()

```
static int sbi_ecall_ipi_handler (
            unsigned long extid,
            unsigned long funcid,
```

```
            unsigned long * args,
            unsigned long * out_val,
            struct sbi_trap_info * out_trap )  [static]
```

Here is the call graph for this function:

26.84.1.2 sbi_ecall_rfence_handler()

```
static int sbi_ecall_rfence_handler (
            unsigned long extid,
            unsigned long funcid,
            unsigned long * args,
            unsigned long * out_val,
            struct sbi_trap_info * out_trap )  [static]
```

Here is the call graph for this function:

26.84.1.3 sbi_ecall_time_handler()

```
static int sbi_ecall_time_handler (
            unsigned long extid,
            unsigned long funcid,
            unsigned long * args,
            unsigned long * out_val,
            struct sbi_trap_info * out_trap )  [static]
```

Here is the call graph for this function:

26.84.2 Variable Documentation

26.84.2.1 ecall_ipi

```
struct sbi_ecall_extension ecall_ipi
```

Initial value:

```
= {
    .extid_start = SBI_EXT_IPI,
    .extid_end = SBI_EXT_IPI,
    .handle = sbi_ecall_ipi_handler,
}
```

26.84.2.2 ecall_rfence

```
struct sbi_ecall_extension ecall_rfence
```

Initial value:

```
= {
    .extid_start = SBI_EXT_RFENCE,
    .extid_end = SBI_EXT_RFENCE,
    .handle = sbi_ecall_rfence_handler,
}
```

26.84.2.3 ecall_time

```
struct sbi_ecall_extension ecall_time
```

Initial value:

```
= {
    .extid_start = SBI_EXT_TIME,
    .extid_end = SBI_EXT_TIME,
    .handle = sbi_ecall_time_handler,
}
```

26.85 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 dependency graph for sbi_ecall_vendor.c:

Functions

- static int sbi_ecall_vendor_probe (unsigned long extid, unsigned long *out_val)
- static int sbi_ecall_vendor_handler (unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct sbi_trap_info *out_trap)

Variables

- struct sbi_ecall_extension ecall_vendor

26.85.1 Function Documentation

26.85.1.1 sbi_ecall_vendor_handler()

```
static int sbi_ecall_vendor_handler (
            unsigned long extid,
            unsigned long funcid,
            unsigned long * args,
            unsigned long * out_val,
            struct sbi_trap_info * out_trap )  [static]
```

Here is the call graph for this function:

26.85.1.2 sbi_ecall_vendor_probe()

```
static int sbi_ecall_vendor_probe (
            unsigned long extid,
            unsigned long * out_val )  [static]
```

Here is the call graph for this function:

26.85.2 Variable Documentation

26.85.2.1 ecall_vendor

```
struct sbi_ecall_extension ecall_vendor
```

Initial value:

```
= {
    .extid_start = SBI_EXT_VENDOR_START,
    .extid_end = SBI_EXT_VENDOR_END,
    .probe = sbi_ecall_vendor_probe,
```

```
    .handle = sbi_ecall_vendor_handler,
}
```

26.86 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_console.h>
#include <sbi/sbi_emulate_csr.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_emulate_csr.c:

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)

26.86.1 Function Documentation

26.86.1.1 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:

26.86.1.2 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:

26.87 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)
- bool sbi_fifo_is_full (struct sbi_fifo *fifo)
- static void __sbi_fifo_enqueue (struct sbi_fifo *fifo, void *data)
- static bool __sbi_fifo_is_empty (struct sbi_fifo *fifo)
- bool 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)
- int sbi_fifo_dequeue (struct sbi_fifo *fifo, void *data)

26.87.1 Function Documentation

26.87.1.1 __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:

26.87.1.2 __sbi_fifo_is_empty()

```
static bool __sbi_fifo_is_empty (
            struct sbi_fifo * fifo )  [inline], [static]
```

Here is the caller graph for this function:

26.87.1.3 __sbi_fifo_is_full()

```
static bool __sbi_fifo_is_full (
            struct sbi_fifo * fifo )  [inline], [static]
```

Here is the caller graph for this function:

26.87.1.4 __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:

26.87.1.5 sbi_fifo_avail()

```
u16 sbi_fifo_avail (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

26.87.1.6 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:

26.87.1.7 sbi_fifo_enqueue()

```
int sbi_fifo_enqueue (
            struct sbi_fifo * fifo,
            void * data )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.87.1.8 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:

26.87.1.9 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:

26.87.1.10 sbi_fifo_is_empty()

```
bool sbi_fifo_is_empty (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

26.87.1.11 sbi_fifo_is_full()

```
bool sbi_fifo_is_full (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

26.87.1.12 sbi_fifo_reset()

```
bool sbi_fifo_reset (
            struct sbi_fifo * fifo )
```

Here is the call graph for this function:

26.88 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_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_string.h>
```

Include dependency graph for sbi_hart.c:

Macros

- #define __detect_pmp(__pmp_csr)

Functions

- void __sbi_expected_trap (void)
- void __sbi_expected_trap_hext (void)
- static void mstatus_init (struct sbi_scratch *scratch, u32 hartid)
- static int fp_init (u32 hartid)
- static int delegate_traps (struct sbi_scratch *scratch, u32 hartid)
- void sbi_hart_delegation_dump (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_count (struct sbi_scratch *scratch)
- int sbi_hart_pmp_get (struct sbi_scratch *scratch, unsigned int n, unsigned long *prot_out, unsigned long *addr_out, unsigned long *size)
- void sbi_hart_pmp_dump (struct sbi_scratch *scratch)
- int sbi_hart_pmp_check_addr (struct sbi_scratch *scratch, unsigned long addr, unsigned long attr)
- static int pmp_init (struct sbi_scratch *scratch, u32 hartid)
- bool sbi_hart_has_feature (struct sbi_scratch *scratch, unsigned long feature)
- static unsigned long hart_get_features (struct sbi_scratch *scratch)
- static char * sbi_hart_feature_id2string (unsigned long feature)
- void sbi_hart_get_features_str (struct sbi_scratch *scratch, char *features_str, int nfstr)
- static void hart_detect_features (struct sbi_scratch *scratch)
- int sbi_hart_init (struct sbi_scratch *scratch, u32 hartid, bool cold_boot)
- void __attribute__ ((noreturn))

Variables

- void(* sbi_hart_expected_trap)(void) = &__sbi_expected_trap
- static unsigned long hart_features_offset

26.88.1 Macro Definition Documentation

26.88.1.1 __detect_pmp

```
#define __detect_pmp(
            __pmp_csr )
```

Value:

```
val = csr_read_allowed(__pmp_csr, (ulong)&trap);   \
if (!trap.cause) {                                 \
    csr_write_allowed(__pmp_csr, (ulong)&trap, val);\
    if (!trap.cause)                               \
        hfeatures->pmp_count++;                    \
}
```

26.88.2 Function Documentation

26.88.2.1 __attribute__()

```
void __attribute__ (
            (noreturn)  )
```

26.88.2.2 __sbi_expected_trap()

```
void __sbi_expected_trap (
            void  )
```

26.88.2.3 __sbi_expected_trap_hext()

```
void __sbi_expected_trap_hext (
            void  )
```

Here is the caller graph for this function:

26.88.2.4 delegate_traps()

```
static int delegate_traps (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the caller graph for this function:

26.88.2.5 fp_init()

```
static int fp_init (
            u32 hartid )  [static]
```

Here is the caller graph for this function:

26.88.2.6 hart_detect_features()

```
static void hart_detect_features (
            struct sbi_scratch * scratch )  [static]
```

Here is the caller graph for this function:

26.88.2.7 hart_get_features()

```
static unsigned long hart_get_features (
            struct sbi_scratch * scratch )  [static]
```

Here is the caller graph for this function:

26.88.2.8 mstatus_init()

```
static void mstatus_init (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.88.2.9 pmp_init()

```
static int pmp_init (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.88.2.10 sbi_hart_delegation_dump()

```
void sbi_hart_delegation_dump (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.88.2.11 sbi_hart_feature_id2string()

```
static char* sbi_hart_feature_id2string (
            unsigned long feature )  [inline], [static]
```

Here is the caller graph for this function:

26.88.2.12 sbi_hart_get_features_str()

```
void sbi_hart_get_features_str (
            struct sbi_scratch * scratch,
```

```
char * features_str,
int nfstr )
```

Get the hart features in string format

Parameters

scratch	pointer to the HART scratch space
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:

26.88.2.13 sbi_hart_has_feature()

```
bool sbi_hart_has_feature (
            struct sbi_scratch * scratch,
            unsigned long feature )
```

Check whether a particular hart feature is available

Parameters

scratch	pointer to the HART scratch space
feature	the feature to check

Returns

true (feature available) or false (feature not available)

Here is the caller graph for this function:

26.88.2.14 sbi_hart_init()

```
int sbi_hart_init (
            struct sbi_scratch * scratch,
            u32 hartid,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.88.2.15 sbi_hart_pmp_check_addr()

```
int sbi_hart_pmp_check_addr (
            struct sbi_scratch * scratch,
            unsigned long addr,
            unsigned long attr )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.88.2.16 sbi_hart_pmp_count()

```
unsigned int sbi_hart_pmp_count (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

26.88.2.17 sbi_hart_pmp_dump()

```
void sbi_hart_pmp_dump (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.88.2.18 sbi_hart_pmp_get()

```
int sbi_hart_pmp_get (
            struct sbi_scratch * scratch,
            unsigned int n,
            unsigned long * prot_out,
            unsigned long * addr_out,
            unsigned long * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.88.3 Variable Documentation

26.88.3.1 hart_features_offset

```
unsigned long hart_features_offset  [static]
```

26.88.3.2 sbi_hart_expected_trap

```
void(* sbi_hart_expected_trap) (void) = &__sbi_expected_trap
```

26.89 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_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_platform.h>
#include <sbi/sbi_system.h>
```

```
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_hsm.c:

Functions

- int sbi_hsm_hart_state_to_status (int state)
- int sbi_hsm_hart_get_state (u32 hartid)
- bool sbi_hsm_hart_started (u32 hartid)
- int sbi_hsm_hart_started_mask (ulong hbase, ulong *out_hmask)
- void sbi_hsm_prepare_next_jump (struct sbi_scratch *scratch, u32 hartid)
- static void sbi_hsm_hart_wait (struct sbi_scratch *scratch, u32 hartid)
- int sbi_hsm_init (struct sbi_scratch *scratch, u32 hartid, bool cold_boot)
- void __noreturn sbi_hsm_exit (struct sbi_scratch *scratch)
- int sbi_hsm_hart_start (struct sbi_scratch *scratch, u32 hartid, ulong saddr, ulong priv)
- int sbi_hsm_hart_stop (struct sbi_scratch *scratch, bool exitnow)

Variables

- static unsigned long hart_data_offset

26.89.1 Function Documentation

26.89.1.1 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:

26.89.1.2 sbi_hsm_hart_get_state()

```
int sbi_hsm_hart_get_state (
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.89.1.3 sbi_hsm_hart_start()

```
int sbi_hsm_hart_start (
            struct sbi_scratch * scratch,
            u32 hartid,
            ulong saddr,
            ulong priv )
```

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:

26.89.1.4 sbi_hsm_hart_started()

```
bool sbi_hsm_hart_started (
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.89.1.5 sbi_hsm_hart_started_mask()

```
int sbi_hsm_hart_started_mask (
            ulong hbase,
            ulong * out_hmask )
```

Get ulong HART mask for given HART base ID

Parameters

hbase	the HART base ID
out_hmask	the output ulong HART mask

Returns

0 on success and SBI_Exxx ($<$ 0) on failure Note: the output HART mask will be set to zero on failure as well.

Here is the call graph for this function:

Here is the caller graph for this function:

26.89.1.6 sbi_hsm_hart_state_to_status()

```
int sbi_hsm_hart_state_to_status (
            int state )
```

Here is the caller graph for this function:

26.89.1.7 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:

26.89.1.8 sbi_hsm_hart_wait()

```
static void sbi_hsm_hart_wait (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.89.1.9 sbi_hsm_init()

```
int sbi_hsm_init (
            struct sbi_scratch * scratch,
            u32 hartid,
            bool cold_boot )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.89.1.10 sbi_hsm_prepare_next_jump()

```
void sbi_hsm_prepare_next_jump (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.89.2 Variable Documentation

26.89.2.1 hart_data_offset

```
unsigned long hart_data_offset  [static]
```

26.90 lib/sbi/sbi_illegal_insn.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_illegal_insn.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.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 system_opcode_insn (ulong insn, struct sbi_trap_regs *regs)
- int sbi_illegal_insn_handler (ulong insn, struct sbi_trap_regs *regs)

Variables

- static illegal_insn_func illegal_insn_table [32]

26.90.1 Typedef Documentation

26.90.1.1 illegal_insn_func

```
typedef int(* illegal_insn_func) (ulong insn, struct sbi_trap_regs *regs)
```

26.90.2 Function Documentation

26.90.2.1 sbi_illegal_insn_handler()

```
int sbi_illegal_insn_handler (
            ulong insn,
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.90.2.2 system_opcode_insn()

```
static int system_opcode_insn (
            ulong insn,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

26.90.2.3 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:

26.90.3 Variable Documentation

26.90.3.1 illegal_insn_table

```
illegal_insn_func illegal_insn_table[32]  [static]
```

26.91 lib/sbi/sbi_init.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_locks.h>
#include <sbi/sbi_console.h>
```

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_platform.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 dependency graph for sbi_init.c:

Macros

- #define BANNER

Functions

- static void sbi_boot_prints (struct sbi_scratch *scratch, u32 hartid)
- static void wait_for_coldboot (struct sbi_scratch *scratch, u32 hartid)
- static void wake_coldboot_harts (struct sbi_scratch *scratch, u32 hartid)
- static void __noreturn init_coldboot (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_init_count (u32 hartid)
- void __noreturn sbi_exit (struct sbi_scratch *scratch)

Variables

- static spinlock_t coldboot_lock = SPIN_LOCK_INITIALIZER
- static unsigned long coldboot_done = 0
- static struct sbi_hartmask coldboot_wait_hmask = { 0 }
- static unsigned long init_count_offset
- static atomic_t coldboot_lottery = ATOMIC_INITIALIZER(0)

26.91.1 Macro Definition Documentation

26.91.1.1 BANNER

```
#define BANNER
```

Value:

```
"   ____                    _____ ____ _____\n"     \
"  / __ \\                  / ____|  _ \\_   _|\n"  \
" | |  | |_ __   ___ _ __ | (___ | |_) || |\n"      \
" | |  | | '_ \\ / _ \\ '_ \\ \\___ \\|  _ < | |\n" \
" | |__| | |_) |  __/ | | |____) | |_) || |_\n"    \
"  \\____/| .__/ \\___|_| |_|_____/|____/_____|\n"  \
"        | |\n"                                    \
"        |_|\n\n"
```

26.91.2 Function Documentation

26.91.2.1 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:

26.91.2.2 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:

26.91.2.3 sbi_boot_prints()

```
static void sbi_boot_prints (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.91.2.4 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:

26.91.2.5 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:

26.91.2.6 sbi_init_count()

```
unsigned long sbi_init_count (
            u32 hartid )
```

Here is the caller graph for this function:

26.91.2.7 wait_for_coldboot()

```
static void wait_for_coldboot (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.91.2.8 wake_coldboot_harts()

```
static void wake_coldboot_harts (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.91.3 Variable Documentation

26.91.3.1 coldboot_done

```
unsigned long coldboot_done = 0  [static]
```

26.91.3.2 coldboot_lock

```
spinlock_t coldboot_lock = SPIN_LOCK_INITIALIZER  [static]
```

26.91.3.3 coldboot_lottery

```
atomic_t coldboot_lottery = ATOMIC_INITIALIZER(0)  [static]
```

26.91.3.4 coldboot_wait_hmask

```
struct sbi_hartmask coldboot_wait_hmask = { 0 }  [static]
```

26.91.3.5 init_count_offset

```
unsigned long init_count_offset  [static]
```

26.92 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_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 dependency graph for sbi_ipi.c:

Functions

- static int sbi_ipi_send (struct sbi_scratch *scratch, u32 remote_hartid, u32 event, void *data)
- 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 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_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_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

26.92.1 Function Documentation

26.92.1.1 sbi_ipi_clear_smode()

```
void sbi_ipi_clear_smode (
            void  )
```

Here is the caller graph for this function:

26.92.1.2 sbi_ipi_event_create()

```
int sbi_ipi_event_create (
            const struct sbi_ipi_event_ops * ops )
```

Here is the caller graph for this function:

26.92.1.3 sbi_ipi_event_destroy()

```
void sbi_ipi_event_destroy (
            u32 event )
```

26.92.1.4 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:

26.92.1.5 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:

26.92.1.6 sbi_ipi_process()

```
void sbi_ipi_process (
            void )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.92.1.7 sbi_ipi_process_halt()

```
static void sbi_ipi_process_halt (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

26.92.1.8 sbi_ipi_process_smode()

```
static void sbi_ipi_process_smode (
            struct sbi_scratch * scratch )  [static]
```

26.92.1.9 sbi_ipi_send()

```
static int sbi_ipi_send (
            struct sbi_scratch * scratch,
            u32 remote_hartid,
```

```
            u32 event,
            void * data )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.92.1.10 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:

26.92.1.11 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:

26.92.1.12 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:

26.92.2 Variable Documentation

26.92.2.1 ipi_data_off

```
unsigned long ipi_data_off [static]
```

26.92.2.2 ipi_halt_event

```
u32 ipi_halt_event = SBI_IPI_EVENT_MAX [static]
```

26.92.2.3 ipi_halt_ops

```
struct sbi_ipi_event_ops ipi_halt_ops [static]
```

Initial value:

```
= {
    .name = "IPI_HALT",
    .process = sbi_ipi_process_halt,
}
```

26.92.2.4 ipi_ops_array

```
const struct sbi_ipi_event_ops* ipi_ops_array[SBI_IPI_EVENT_MAX] [static]
```

26.92.2.5 ipi_smode_event

```
u32 ipi_smode_event = SBI_IPI_EVENT_MAX [static]
```

26.92.2.6 ipi_smode_ops

```
struct sbi_ipi_event_ops ipi_smode_ops [static]
```

Initial value:

```
= {
    .name = "IPI_SMODE",
    .process = sbi_ipi_process_smode,
}
```

26.93 lib/sbi/sbi_math.c File Reference

Functions

- unsigned long log2roundup (unsigned long x)

26.93.1 Function Documentation

26.93.1.1 log2roundup()

```
unsigned long log2roundup (
            unsigned long x )
```

Here is the caller graph for this function:

26.94 lib/sbi/sbi_misaligned_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_misaligned_ldst.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
```

Include dependency graph for sbi_misaligned_ldst.c:

Functions

- int sbi_misaligned_load_handler (ulong addr, ulong tval2, ulong tinst, struct sbi_trap_regs *regs)
- int sbi_misaligned_store_handler (ulong addr, ulong tval2, ulong tinst, struct sbi_trap_regs *regs)

26.94.1 Function Documentation

26.94.1.1 sbi_misaligned_load_handler()

```
int sbi_misaligned_load_handler (
            ulong addr,
            ulong tval2,
            ulong tinst,
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.94.1.2 sbi_misaligned_store_handler()

```
int sbi_misaligned_store_handler (
            ulong addr,
            ulong tval2,
            ulong tinst,
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.95 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)
- u32 sbi_platform_hart_index (const struct sbi_platform ∗plat, u32 hartid)

26.95.1 Function Documentation

26.95.1.1 sbi_platform_feature_id2string()

```
static char* sbi_platform_feature_id2string (
            unsigned long feature )  [inline], [static]
```

Here is the caller graph for this function:

26.95.1.2 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:

26.95.1.3 sbi_platform_hart_index()

```
u32 sbi_platform_hart_index (
            const struct sbi_platform * plat,
            u32 hartid )
```

Get HART index for the given HART

Parameters

plat	pointer to struct sbi_platform
hartid	HART ID

Returns

0 <= value < hart_count for valid HART otherwise -1U

Here is the caller graph for this function:

26.96 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

- int sbi_scratch_init (struct sbi_scratch *scratch)
- unsigned long sbi_scratch_alloc_offset (unsigned long size, const char *owner)
- void sbi_scratch_free_offset (unsigned long offset)

Variables

- u32 last_hartid_having_scratch = SBI_HARTMASK_MAX_BITS
- struct sbi_scratch * hartid_to_scratch_table [SBI_HARTMASK_MAX_BITS] = { 0 }
- static spinlock_t extra_lock = SPIN_LOCK_INITIALIZER
- static unsigned long extra_offset = SBI_SCRATCH_EXTRA_SPACE_OFFSET

26.96.1 Typedef Documentation

26.96.1.1 hartid2scratch

```
typedef struct sbi_scratch*(* hartid2scratch) (ulong hartid, ulong hartindex)
```

26.96.2 Function Documentation

26.96.2.1 sbi_scratch_alloc_offset()

```
unsigned long sbi_scratch_alloc_offset (
            unsigned long size,
            const char * owner )
```

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:

26.96.2.2 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:

26.96.2.3 sbi_scratch_init()

```
int sbi_scratch_init (
            struct sbi_scratch * scratch )
```

Initialize scatch table and allocator Here is the call graph for this function:

Here is the caller graph for this function:

26.96.3 Variable Documentation

26.96.3.1 extra_lock

```
spinlock_t extra_lock = SPIN_LOCK_INITIALIZER  [static]
```

26.96.3.2 extra_offset

```
unsigned long extra_offset = SBI_SCRATCH_EXTRA_SPACE_OFFSET  [static]
```

26.96.3.3 hartid_to_scratch_table

```
struct sbi_scratch* hartid_to_scratch_table[SBI_HARTMASK_MAX_BITS] = { 0 }
```

HART id to scratch table

26.96.3.4 last_hartid_having_scratch

```
u32 last_hartid_having_scratch = SBI_HARTMASK_MAX_BITS
```

Last HART id having a sbi_scratch pointer

26.97 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)
- 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)

26.97.1 Function Documentation

26.97.1.1 sbi_memchr()

```
void* sbi_memchr (
            const void * s,
            int c,
            size_t count )
```

26.97.1.2 sbi_memcmp()

```
int sbi_memcmp (
            const void * s1,
            const void * s2,
            size_t count )
```

26.97.1.3 sbi_memcpy()

```
void* sbi_memcpy (
            void * dest,
            const void * src,
            size_t count )
```

Here is the caller graph for this function:

26.97.1.4 sbi_memmove()

```
void* sbi_memmove (
            void * dest,
            const void * src,
            size_t count )
```

26.97.1.5 sbi_memset()

```
void* sbi_memset (
            void * s,
            int c,
            size_t count )
```

Here is the caller graph for this function:

26.97.1.6 sbi_strchr()

```
char* sbi_strchr (
            const char * s,
            int c )
```

26.97.1.7 sbi_strcmp()

```
int sbi_strcmp (
            const char * a,
            const char * b )
```

Here is the caller graph for this function:

26.97.1.8 sbi_strcpy()

```
char* sbi_strcpy (
            char * dest,
            const char * src )
```

26.97.1.9 sbi_strlen()

```
size_t sbi_strlen (
            const char * str )
```

Here is the caller graph for this function:

26.97.1.10 sbi_strncpy()

```
char* sbi_strncpy (
            char * dest,
            const char * src,
            size_t count )
```

Here is the caller graph for this function:

26.97.1.11 sbi_strnlen()

```
size_t sbi_strnlen (
            const char * str,
            size_t count )
```

26.97.1.12 sbi_strrchr()

```
char* sbi_strrchr (
            const char * s,
            int c )
```

Here is the call graph for this function:

26.98 lib/sbi/sbi_system.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_bitops.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 dependency graph for sbi_system.c:

Functions

- void __noreturn sbi_system_reset (u32 platform_reset_type)

26.98.1 Function Documentation

26.98.1.1 sbi_system_reset()

```
void __noreturn sbi_system_reset (
            u32 platform_reset_type )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.99 lib/sbi/sbi_timer.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_timer.c:

Functions

- static u64 get_ticks (const struct sbi_platform *plat)
- 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_set_delta_upper (ulong delta_upper)
- void sbi_timer_event_start (u64 next_event)
- void sbi_timer_process (void)
- 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)(const struct sbi_platform *plat)

26.99.1 Function Documentation

26.99.1.1 get_ticks()

```
static u64 get_ticks (
            const struct sbi_platform * plat )  [static]
```

Here is the caller graph for this function:

26.99.1.2 sbi_timer_event_start()

```
void sbi_timer_event_start (
            u64 next_event )
```

Start timer event for current HART Here is the call graph for this function:

Here is the caller graph for this function:

26.99.1.3 sbi_timer_exit()

```
void sbi_timer_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.99.1.4 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:

26.99.1.5 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:

26.99.1.6 sbi_timer_process()

```
void sbi_timer_process (
            void  )
```

Process timer event for current HART Here is the caller graph for this function:

26.99.1.7 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:

26.99.1.8 sbi_timer_set_delta_upper()

```
void sbi_timer_set_delta_upper (
            ulong delta_upper )
```

Set upper 32-bits of timer delta value for current HART Here is the caller graph for this function:

26.99.1.9 sbi_timer_value()

```
u64 sbi_timer_value (
            void  )
```

Get timer value for current HART Here is the caller graph for this function:

26.99.1.10 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:

26.99.2 Variable Documentation

26.99.2.1 get_time_val

```
u64(* get_time_val) (const struct sbi_platform *plat)  [static]
```

26.99.2.2 time_delta_off

```
unsigned long time_delta_off  [static]
```

26.100 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_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 dependency graph for sbi_tlb.c:

Functions

- static void sbi_tlb_flush_all (void)
- static void sbi_tlb_hfence_vvma (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_hfence_gvma (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_sfence_vma (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_hfence_vvma_asid (struct sbi_tlb_info *tinfo)

- static void sbi_tlb_hfence_gvma_vmid (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_sfence_vma_asid (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_local_flush (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_entry_process (struct sbi_tlb_info *tinfo)
- static void sbi_tlb_process_count (struct sbi_scratch *scratch, int count)
- static void sbi_tlb_process (struct sbi_scratch *scratch)
- static void sbi_tlb_sync (struct sbi_scratch *scratch)
- static int __sbi_tlb_range_check (struct sbi_tlb_info *curr, struct sbi_tlb_info *next)
- static int sbi_tlb_update_cb (void *in, void *data)
- static int sbi_tlb_update (struct sbi_scratch *scratch, struct sbi_scratch *remote_scratch, u32 remote_hartid, 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

26.100.1 Function Documentation

26.100.1.1 __sbi_tlb_range_check()

```
static int __sbi_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:

26.100.1.2 sbi_tlb_entry_process()

```
static void sbi_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:

26.100.1.3 sbi_tlb_flush_all()

```
static void sbi_tlb_flush_all (
            void  )  [static]
```

Here is the caller graph for this function:

26.100.1.4 sbi_tlb_hfence_gvma()

```
static void sbi_tlb_hfence_gvma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.100.1.5 sbi_tlb_hfence_gvma_vmid()

```
static void sbi_tlb_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:

26.100.1.6 sbi_tlb_hfence_vvma()

```
static void sbi_tlb_hfence_vvma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.100.1.7 sbi_tlb_hfence_vvma_asid()

```
static void sbi_tlb_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:

26.100.1.8 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:

26.100.1.9 sbi_tlb_local_flush()

```
static void sbi_tlb_local_flush (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.100.1.10 sbi_tlb_process()

```
static void sbi_tlb_process (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

26.100.1.11 sbi_tlb_process_count()

```
static void sbi_tlb_process_count (
            struct sbi_scratch * scratch,
            int count )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.100.1.12 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:

26.100.1.13 sbi_tlb_sfence_vma()

```
static void sbi_tlb_sfence_vma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.100.1.14 sbi_tlb_sfence_vma_asid()

```
static void sbi_tlb_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:

26.100.1.15 sbi_tlb_sync()

```
static void sbi_tlb_sync (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:

26.100.1.16 sbi_tlb_update()

```
static int sbi_tlb_update (
            struct sbi_scratch * scratch,
            struct sbi_scratch * remote_scratch,
            u32 remote_hartid,
            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:

26.100.1.17 sbi_tlb_update_cb()

```
static int sbi_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:

26.100.2 Variable Documentation

26.100.2.1 tlb_event

```
u32 tlb_event = SBI_IPI_EVENT_MAX  [static]
```

26.100.2.2 tlb_fifo_mem_off

```
unsigned long tlb_fifo_mem_off  [static]
```

26.100.2.3 tlb_fifo_off

```
unsigned long tlb_fifo_off  [static]
```

26.100.2.4 tlb_ops

```
struct sbi_ipi_event_ops tlb_ops  [static]
```

Initial value:

```
= {
    .name = "IPI_TLB",
    .update = sbi_tlb_update,
    .sync = sbi_tlb_sync,
    .process = sbi_tlb_process,
}
```

26.100.2.5 tlb_range_flush_limit

```
unsigned long tlb_range_flush_limit  [static]
```

26.100.2.6 tlb_sync_off

```
unsigned long tlb_sync_off  [static]
```

26.101 lib/sbi/sbi_trap.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.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_misaligned_ldst.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_trap.c:

Functions

- static void __noreturn sbi_trap_error (const char *msg, int rc, ulong mcause, ulong mtval, ulong mtval2, ulong mtinst, struct sbi_trap_regs *regs)
- int sbi_trap_redirect (struct sbi_trap_regs *regs, struct sbi_trap_info *trap)

- void sbi_trap_handler (struct sbi_trap_regs *regs)

26.101.1 Function Documentation

26.101.1.1 sbi_trap_error()

```
static void __noreturn sbi_trap_error (
            const char * msg,
            int rc,
            ulong mcause,
            ulong mtval,
            ulong mtval2,
            ulong mtinst,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.101.1.2 sbi_trap_handler()

```
void sbi_trap_handler (
            struct sbi_trap_regs * regs )
```

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

regs	pointer to register state

Here is the call graph for this function:

26.101.1.3 sbi_trap_redirect()

```
int sbi_trap_redirect (
            struct sbi_trap_regs * regs,
            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 caller graph for this function:

26.102 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

- u64 sbi_load_u64 (const u64 ∗addr, struct sbi_trap_info ∗trap)
- void sbi_store_u64 (u64 ∗addr, u64 val, struct sbi_trap_info ∗trap)
- ulong sbi_get_insn (ulong mepc, struct sbi_trap_info ∗trap)

26.102.1 Macro Definition Documentation

26.102.1.1 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.

26.102.1.2 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");                                           \
}
```

26.102.2 Function Documentation

26.102.2.1 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:

26.102.2.2 sbi_load_u64()

```
u64 sbi_load_u64 (
            const u64 * addr,
            struct sbi_trap_info * trap )
```

26.102.2.3 sbi_store_u64()

```
void sbi_store_u64 (
            u64 * addr,
            u64 val,
            struct sbi_trap_info * trap )
```

26.103 lib/utils/fdt/fdt_fixup.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_math.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi_utils/fdt/fdt_fixup.h>
#include <sbi_utils/fdt/fdt_helper.h>
```

Include dependency graph for fdt_fixup.c:

Functions

- void fdt_cpu_fixup (void ∗fdt)
- void fdt_plic_fixup (void ∗fdt, const char ∗compat)
- static int fdt_resv_memory_update_node (void ∗fdt, unsigned long addr, unsigned long size, int index, int parent, bool no_map)
- int fdt_reserved_memory_fixup (void ∗fdt)
- int fdt_reserved_memory_nomap_fixup (void ∗fdt)
- void fdt_fixups (void ∗fdt)

26.103.1 Function Documentation

26.103.1.1 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:

26.103.1.2 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 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:

26.103.1.3 fdt_plic_fixup()

```
void fdt_plic_fixup (
            void * fdt,
            const char * compat )
```

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
compat	PLIC node compatible string

Here is the call graph for this function:

Here is the caller graph for this function:

26.103.1.4 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:

26.103.1.5 fdt_reserved_memory_nomap_fixup()

```
int fdt_reserved_memory_nomap_fixup (
            void * fdt )
```

Fix up the reserved memory subnodes in the device tree
This routine adds the no-map property to the reserved memory subnodes so that the OS does not map those PMP protected memory regions.
Platform codes must call this helper in their final_init() after fdt_fixups() if the OS should not map the PMP protected reserved regions.

Parameters

fdt	device tree blob

Returns

zero on success and -ve on failure

Here is the call graph for this function:

26.103.1.6 fdt_resv_memory_update_node()

```
static int fdt_resv_memory_update_node (
            void * fdt,
            unsigned long addr,
            unsigned long size,
            int index,
            int parent,
            bool no_map )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.104 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_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/plic.h>
```

```
#include <sbi_utils/sys/clint.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_SIFIVE_UART_FREQ 0
- #define DEFAULT_SIFIVE_UART_BAUD 115200
- #define DEFAULT_SIFIVE_UART_REG_SHIFT 0
- #define DEFAULT_SIFIVE_UART_REG_IO_WIDTH 4
- #define DEFAULT_SHAKTI_UART_FREQ 50000000
- #define DEFAULT_SHAKTI_UART_BAUD 115200

Functions

- const struct fdt_match ∗ fdt_match_node (void ∗fdt, int nodeoff, const struct fdt_match ∗match_table)
- int fdt_find_match (void ∗fdt, int startoff, const struct fdt_match ∗match_table, const struct fdt_match ∗∗out↩ _match)
- int fdt_get_node_addr_size (void ∗fdt, int node, unsigned long ∗addr, unsigned long ∗size)
- int fdt_parse_hart_id (void ∗fdt, int cpu_offset, u32 ∗hartid)
- int fdt_parse_max_hart_id (void ∗fdt, u32 ∗max_hartid)
- int fdt_parse_shakti_uart_node (void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_sifive_uart_node (void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_uart8250_node (void ∗fdt, int nodeoffset, struct platform_uart_data ∗uart)
- int fdt_parse_uart8250 (void ∗fdt, struct platform_uart_data ∗uart, const char ∗compatible)
- int fdt_parse_plic_node (void ∗fdt, int nodeoffset, struct plic_data ∗plic)
- int fdt_parse_plic (void ∗fdt, struct plic_data ∗plic, const char ∗compat)
- int fdt_parse_clint_node (void ∗fdt, int nodeoffset, bool for_timer, struct clint_data ∗clint)
- int fdt_parse_compat_addr (void ∗fdt, unsigned long ∗addr, const char ∗compatible)

26.104.1 Macro Definition Documentation

26.104.1.1 DEFAULT_SHAKTI_UART_BAUD

```
#define DEFAULT_SHAKTI_UART_BAUD 115200
```

26.104.1.2 DEFAULT_SHAKTI_UART_FREQ

```
#define DEFAULT_SHAKTI_UART_FREQ 50000000
```

26.104.1.3 DEFAULT_SIFIVE_UART_BAUD

```
#define DEFAULT_SIFIVE_UART_BAUD 115200
```

26.104.1.4 DEFAULT_SIFIVE_UART_FREQ

```
#define DEFAULT_SIFIVE_UART_FREQ 0
```

26.104.1.5 DEFAULT_SIFIVE_UART_REG_IO_WIDTH

```
#define DEFAULT_SIFIVE_UART_REG_IO_WIDTH 4
```

26.104.1.6 DEFAULT_SIFIVE_UART_REG_SHIFT

```
#define DEFAULT_SIFIVE_UART_REG_SHIFT 0
```

26.104.1.7 DEFAULT_UART_BAUD

```
#define DEFAULT_UART_BAUD 115200
```

26.104.1.8 DEFAULT_UART_FREQ

```
#define DEFAULT_UART_FREQ 0
```

26.104.1.9 DEFAULT_UART_REG_IO_WIDTH

```
#define DEFAULT_UART_REG_IO_WIDTH 1
```

26.104.1.10 DEFAULT_UART_REG_SHIFT

```
#define DEFAULT_UART_REG_SHIFT 0
```

26.104.2 Function Documentation

26.104.2.1 fdt_find_match()

```
int fdt_find_match (
            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:

26.104.2.2 fdt_get_node_addr_size()

```
int fdt_get_node_addr_size (
            void * fdt,
            int node,
            unsigned long * addr,
            unsigned long * size )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.104.2.3 fdt_match_node()

```
const struct fdt_match* fdt_match_node (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match_table )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.104.2.4 fdt_parse_clint_node()

```
int fdt_parse_clint_node (
            void * fdt,
```

```
            int nodeoffset,
            bool for_timer,
            struct clint_data * clint )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.104.2.5 fdt_parse_compat_addr()

```
int fdt_parse_compat_addr (
            void * fdt,
            unsigned long * addr,
            const char * compatible )
```

Here is the call graph for this function:

26.104.2.6 fdt_parse_hart_id()

```
int fdt_parse_hart_id (
            void * fdt,
            int cpu_offset,
            u32 * hartid )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.104.2.7 fdt_parse_max_hart_id()

```
int fdt_parse_max_hart_id (
            void * fdt,
            u32 * max_hartid )
```

Here is the call graph for this function:

26.104.2.8 fdt_parse_plic()

```
int fdt_parse_plic (
            void * fdt,
            struct plic_data * plic,
            const char * compat )
```

Here is the call graph for this function:

26.104.2.9 fdt_parse_plic_node()

```
int fdt_parse_plic_node (
            void * fdt,
            int nodeoffset,
            struct plic_data * plic )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.104.2.10 fdt_parse_shakti_uart_node()

```
int fdt_parse_shakti_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

UART address is mandaotry. 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:

26.104.2.11 fdt_parse_sifive_uart_node()

```
int fdt_parse_sifive_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

UART address is mandaotry. 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:

26.104.2.12 fdt_parse_uart8250()

```
int fdt_parse_uart8250 (
            void * fdt,
            struct platform_uart_data * uart,
            const char * compatible )
```

Here is the call graph for this function:

26.104.2.13 fdt_parse_uart8250_node()

```
int fdt_parse_uart8250_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

UART address is mandaotry. 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:

26.105 lib/utils/ipi/fdt_ipi.c File Reference

```
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/ipi/fdt_ipi.h>
```

Include dependency graph for fdt_ipi.c:

Functions

- static void dummy_send (u32 target_hart)
- static void dummy_clear (u32 target_hart)
- void fdt_ipi_send (u32 target_hart)
- void fdt_ipi_clear (u32 target_hart)
- void fdt_ipi_exit (void)
- static int fdt_ipi_warm_init (void)
- static int fdt_ipi_cold_init (void)
- int fdt_ipi_init (bool cold_boot)

Variables

- struct fdt_ipi fdt_ipi_clint
- static struct fdt_ipi * ipi_drivers [ ]
- static struct fdt_ipi dummy
- static struct fdt_ipi * current_driver = &dummy

26.105.1 Function Documentation

26.105.1.1 dummy_clear()

```
static void dummy_clear (
            u32 target_hart )  [static]
```

26.105.1.2 dummy_send()

```
static void dummy_send (
            u32 target_hart )  [static]
```

26.105.1.3 fdt_ipi_clear()

```
void fdt_ipi_clear (
            u32 target_hart )
```

26.105.1.4 fdt_ipi_cold_init()

```
static int fdt_ipi_cold_init (
            void ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.105.1.5 fdt_ipi_exit()

```
void fdt_ipi_exit (
            void )
```

26.105.1.6 fdt_ipi_init()

```
int fdt_ipi_init (
            bool cold_boot )
```

Here is the call graph for this function:

26.105.1.7 fdt_ipi_send()

```
void fdt_ipi_send (
            u32 target_hart )
```

26.105.1.8 fdt_ipi_warm_init()

```
static int fdt_ipi_warm_init (
            void ) [static]
```

Here is the caller graph for this function:

26.105.2 Variable Documentation

26.105.2.1 current_driver

```
struct fdt_ipi* current_driver = &dummy [static]
```

26.105.2.2 dummy

```
struct fdt_ipi dummy [static]
```

Initial value:

```
= {
    .match_table = NULL,
    .cold_init = NULL,
    .warm_init = NULL,
    .exit = NULL,
    .send = dummy_send,
    .clear = dummy_clear
}
```

26.105.2.3 fdt_ipi_clint

```
struct fdt_ipi fdt_ipi_clint
```

26.105.2.4 ipi_drivers

```
struct fdt_ipi* ipi_drivers[] [static]
```

Initial value:

```
= {
    &fdt_ipi_clint
}
```

26.106 lib/utils/ipi/fdt_ipi_clint.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/ipi/fdt_ipi.h>
#include <sbi_utils/sys/clint.h>
```

Include dependency graph for fdt_ipi_clint.c:

Macros

- #define CLINT_IPI_MAX_NR 16

Functions

- static int ipi_clint_cold_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static unsigned long clint_ipi_count = 0
- static struct clint_data clint_ipi [CLINT_IPI_MAX_NR]
- static const struct fdt_match ipi_clint_match [ ]
- struct fdt_ipi fdt_ipi_clint

26.106.1 Macro Definition Documentation

26.106.1.1 CLINT_IPI_MAX_NR

```
#define CLINT_IPI_MAX_NR 16
```

26.106.2 Function Documentation

26.106.2.1 ipi_clint_cold_init()

```
static int ipi_clint_cold_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

26.106.3 Variable Documentation

26.106.3.1 clint_ipi

```
struct clint_data clint_ipi[CLINT_IPI_MAX_NR]  [static]
```

26.106.3.2 clint_ipi_count

```
unsigned long clint_ipi_count = 0  [static]
```

26.106.3.3 fdt_ipi_clint

```
struct fdt_ipi fdt_ipi_clint
```

Initial value:

```
= {
    .match_table = ipi_clint_match,
    .cold_init = ipi_clint_cold_init,
    .warm_init = clint_warm_ipi_init,
    .exit = NULL,
    .send = clint_ipi_send,
    .clear = clint_ipi_clear,
}
```

26.106.3.4 ipi_clint_match

```
const struct fdt_match ipi_clint_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,clint0" },
    { },
}
```

26.107 lib/utils/irqchip/fdt_irqchip.c File Reference

```
#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

- void fdt_irqchip_exit (void)
- static int fdt_irqchip_warm_init (void)
- static int fdt_irqchip_cold_init (void)
- int fdt_irqchip_init (bool cold_boot)

Variables

- struct fdt_irqchip fdt_irqchip_plic
- static struct fdt_irqchip ∗ irqchip_drivers [ ]
- static struct fdt_irqchip ∗ current_driver = NULL

26.107.1 Function Documentation

26.107.1.1 fdt_irqchip_cold_init()

```
static int fdt_irqchip_cold_init (
            void ) [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.107.1.2 fdt_irqchip_exit()

```
void fdt_irqchip_exit (
            void  )
```

26.107.1.3 fdt_irqchip_init()

```
int fdt_irqchip_init (
            bool cold_boot )
```

Here is the call graph for this function:

26.107.1.4 fdt_irqchip_warm_init()

```
static int fdt_irqchip_warm_init (
            void  ) [static]
```

Here is the caller graph for this function:

26.107.2 Variable Documentation

26.107.2.1 current_driver

```
struct fdt_irqchip* current_driver = NULL [static]
```

26.107.2.2 fdt_irqchip_plic

```
struct fdt_irqchip fdt_irqchip_plic
```

26.107.2.3 irqchip_drivers

```
struct fdt_irqchip* irqchip_drivers[] [static]
```

Initial value:

```
= {
    &fdt_irqchip_plic
}
```

26.108 lib/utils/irqchip/fdt_irqchip_plic.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hartmask.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:

Macros

- #define PLIC_MAX_NR 16

Functions

- static int irqchip_plic_warm_init (void)
- static int irqchip_plic_update_hartid_table (void ∗fdt, int nodeoff, struct plic_data ∗pd)
- static int irqchip_plic_cold_init (void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static unsigned long plic_count = 0
- static struct plic_data plic [PLIC_MAX_NR]
- static struct plic_data ∗ plic_hartid2data [SBI_HARTMASK_MAX_BITS]
- static int plic_hartid2context [SBI_HARTMASK_MAX_BITS][2]
- static const struct fdt_match irqchip_plic_match [ ]
- struct fdt_irqchip fdt_irqchip_plic

26.108.1 Macro Definition Documentation

26.108.1.1 PLIC_MAX_NR

```
#define PLIC_MAX_NR 16
```

26.108.2 Function Documentation

26.108.2.1 irqchip_plic_cold_init()

```
static int irqchip_plic_cold_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

26.108.2.2 irqchip_plic_update_hartid_table()

```
static int irqchip_plic_update_hartid_table (
            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:

26.108.2.3 irqchip_plic_warm_init()

```
static int irqchip_plic_warm_init (
            void ) [static]
```

Here is the call graph for this function:

26.108.3 Variable Documentation

26.108.3.1 fdt_irqchip_plic

```
struct fdt_irqchip fdt_irqchip_plic
```

Initial value:

```
= {
    .match_table = irqchip_plic_match,
    .cold_init = irqchip_plic_cold_init,
    .warm_init = irqchip_plic_warm_init,
    .exit = NULL,
}
```

26.108.3.2 irqchip_plic_match

```
const struct fdt_match irqchip_plic_match[] [static]
```

Initial value:

```
= {
    { .compatible = "riscv,plic0" },
    { .compatible = "sifive,plic-1.0.0" },
    { },
}
```

26.108.3.3 plic

```
struct plic_data plic[PLIC_MAX_NR] [static]
```

26.108.3.4 plic_count

```
unsigned long plic_count = 0  [static]
```

26.108.3.5 plic_hartid2context

```
int plic_hartid2context[SBI_HARTMASK_MAX_BITS][2]  [static]
```

26.108.3.6 plic_hartid2data

```
struct plic_data* plic_hartid2data[SBI_HARTMASK_MAX_BITS]  [static]
```

26.109 lib/utils/irqchip/plic.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.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

Functions

- static void plic_set_priority (struct plic_data *plic, u32 source, u32 val)
- void plic_set_thresh (struct plic_data *plic, u32 cntxid, u32 val)
- void plic_set_ie (struct plic_data *plic, u32 cntxid, u32 word_index, u32 val)
- int plic_warm_irqchip_init (struct plic_data *plic, int m_cntx_id, int s_cntx_id)
- int plic_cold_irqchip_init (struct plic_data *plic)

26.109.1 Macro Definition Documentation

26.109.1.1 PLIC_CONTEXT_BASE

```
#define PLIC_CONTEXT_BASE 0x200000
```

26.109.1.2 PLIC_CONTEXT_STRIDE

```
#define PLIC_CONTEXT_STRIDE 0x1000
```

26.109.1.3 PLIC_ENABLE_BASE

```
#define PLIC_ENABLE_BASE 0x2000
```

26.109.1.4 PLIC_ENABLE_STRIDE

```
#define PLIC_ENABLE_STRIDE 0x80
```

26.109.1.5 PLIC_PENDING_BASE

```
#define PLIC_PENDING_BASE 0x1000
```

26.109.1.6 PLIC_PRIORITY_BASE

```
#define PLIC_PRIORITY_BASE 0x0
```

26.109.2 Function Documentation

26.109.2.1 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:

26.109.2.2 plic_set_ie()

```
void plic_set_ie (
            struct plic_data * plic,
            u32 cntxid,
            u32 word_index,
            u32 val )
```

Here is the caller graph for this function:

26.109.2.3 plic_set_priority()

```
static void plic_set_priority (
            struct plic_data * plic,
            u32 source,
            u32 val )  [static]
```

Here is the caller graph for this function:

26.109.2.4 plic_set_thresh()

```
void plic_set_thresh (
            struct plic_data * plic,
            u32 cntxid,
            u32 val )
```

Here is the caller graph for this function:

26.109.2.5 plic_warm_irqchip_init()

```
int plic_warm_irqchip_init (
            struct plic_data * plic,
```

```
            int m_cntx_id,
            int s_cntx_id )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.110 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)
- 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)

26.110.1 Function Documentation

26.110.1.1 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:

26.110.1.2 check_off_()

```
static int check_off_ (
            uint32_t hdrsize,
            uint32_t totalsize,
            uint32_t off )  [static]
```

Here is the caller graph for this function:

26.110.1.3 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:

26.110.1.4 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:

26.110.1.5 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:

26.110.1.6 fdt_find_string_()

```
const char* fdt_find_string_ (
            const char * strtab,
            int tabsize,
            const char * s )
```

Here is the caller graph for this function:

26.110.1.7 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

Returns

offset of first subnode, or -FDT_ERR_NOTFOUND if there is none

Here is the call graph for this function:

26.110.1.8 fdt_header_size_()

```
size_t fdt_header_size_ (
            uint32_t version )
```

fdt_header_size - return the size of the tree's header @fdt: pointer to a flattened device tree Here is the caller graph for this function:

26.110.1.9 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_BAD↩ MAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the caller graph for this function:

26.110.1.10 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:

26.110.1.11 fdt_next_subnode()

```
int fdt_next_subnode (
            const void * fdt,
            int offset )
```

fdt_next_subnode() - get offset of next direct subnode

After first calling fdt_first_subnode(), call this function repeatedly to get direct subnodes of a parent node.

@fdt: FDT blob @offset: Offset of previous subnode

Returns

offset of next subnode, or -FDT_ERR_NOTFOUND if there are no more subnodes

Here is the call graph for this function:

26.110.1.12 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:

26.110.1.13 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:

26.110.1.14 fdt_ro_probe_()

```
int32_t fdt_ro_probe_ (
            const void * fdt )
```

Here is the caller graph for this function:

26.111 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))

26.111.1 Macro Definition Documentation

26.111.1.1 FDT_BEGIN_NODE

```
#define FDT_BEGIN_NODE 0x1 /* Start node:  full name */
```

26.111.1.2 FDT_END

```
#define FDT_END 0x9
```

26.111.1.3 FDT_END_NODE

```
#define FDT_END_NODE 0x2 /* End node */
```

26.111.1.4 FDT_MAGIC

```
#define FDT_MAGIC 0xd00dfeed /* 4:  version, 4:  total size */
```

26.111.1.5 FDT_NOP

```
#define FDT_NOP 0x4 /* nop */
```

26.111.1.6 FDT_PROP

```
#define FDT_PROP
```

Value:

```
0x3      /* Property: name off,
size, content */
```

26.111.1.7 FDT_TAGSIZE

```
#define FDT_TAGSIZE sizeof(fdt32_t)
```

26.111.1.8 FDT_V16_SIZE

```
#define FDT_V16_SIZE FDT_V3_SIZE
```

26.111.1.9 FDT_V17_SIZE

```
#define FDT_V17_SIZE (FDT_V16_SIZE + sizeof(fdt32_t))
```

26.111.1.10 FDT_V1_SIZE

```
#define FDT_V1_SIZE (7*sizeof(fdt32_t))
```

26.111.1.11 FDT_V2_SIZE

```
#define FDT_V2_SIZE (FDT_V1_SIZE + sizeof(fdt32_t))
```

26.111.1.12 FDT_V3_SIZE

```
#define FDT_V3_SIZE (FDT_V2_SIZE + sizeof(fdt32_t))
```

26.112 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)

26.112.1 Function Documentation

26.112.1.1 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:

26.112.1.2 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:

26.112.1.3 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:

26.112.1.4 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_B↩ ADNCELLS, if the node has a badly formatted or invalid #size-cells property -FDT_ERR_BADMAGIC, -FDT_E↩ RR_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:

26.113 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)

26.113.1 Function Documentation

26.113.1.1 fdt_create_empty_tree()

```
int fdt_create_empty_tree (
            void * buf,
            int bufsize )
```

Here is the call graph for this function:

26.114 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)

26.114.1 Function Documentation

26.114.1.1 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_BADPHAN↩ DLE, -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:

26.114.1.2 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:

26.114.1.3 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:

26.114.1.4 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:

26.114.1.5 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:

26.114.1.6 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:

26.114.1.7 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:

26.114.1.8 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:

26.114.1.9 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:

26.114.1.10 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:

26.114.1.11 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:

26.114.1.12 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:

26.114.1.13 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:

26.114.1.14 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:

26.114.1.15 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:

26.115 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)
- int fdt_check_full (const void *fdt, size_t bufsize)

26.115.1 Function Documentation

26.115.1.1 fdt_check_full()

```
int fdt_check_full (
            const void * fdt,
            size_t bufsize )
```

Here is the call graph for this function:

26.115.1.2 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:

26.115.1.3 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_ER↩R_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ER↩R_TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

26.115.1.4 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.

Returns: 0 on success or a negative error-code on failure Here is the call graph for this function:

26.115.1.5 fdt_get_alias()

```
const char* fdt_get_alias (
            const void * fdt,
            const char * name )
```

Here is the call graph for this function:

26.115.1.6 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:

26.115.1.7 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 @address, @size: pointers to 64-bit variables

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:

26.115.1.8 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_BADSTA↩ TE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

26.115.1.9 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. -F↩ DT_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, -FD↩ T_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:

26.115.1.10 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:

26.115.1.11 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:

26.115.1.12 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_BADO↩ FFSET, 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:

26.115.1.13 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:

26.115.1.14 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:

26.115.1.15 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:

26.115.1.16 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:

26.115.1.17 fdt_getprop()

```
const void* fdt_getprop (
            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:

26.115.1.18 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:

26.115.1.19 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:

26.115.1.20 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:

26.115.1.21 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_BA↩DMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TR↩UNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

26.115.1.22 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_B↩ADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

26.115.1.23 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:

26.115.1.24 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:

26.115.1.25 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, -F↩ DT_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:

26.115.1.26 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_E↩

RR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

26.115.1.27 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:

26.115.1.28 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:

26.115.1.29 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_E↩RR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

26.115.1.30 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_ER↩R_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ER↩R_TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

26.115.1.31 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. Here is the call graph for this function:

Here is the caller graph for this function:

26.115.1.32 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:

26.115.1.33 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:

26.115.1.34 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.

Returns

: 1 if the string is found in the list, 0 not found, or invalid list

Here is the caller graph for this function:

26.115.1.35 fdt_stringlist_count()

```
int fdt_stringlist_count (
```

```
            const void * fdt,
            int nodeoffset,
            const char * property )
```

Here is the call graph for this function:

26.115.1.36 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:

26.115.1.37 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:

26.115.1.38 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:

26.115.1.39 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:

26.115.1.40 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_ER↩R_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

26.115.1.41 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:

26.116 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 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 fdt_open_into (const void ∗fdt, void ∗buf, int bufsize)
- int fdt_pack (void ∗fdt)

26.116.1 Macro Definition Documentation

26.116.1.1 FDT_RW_PROBE

```
#define FDT_RW_PROBE(
                fdt )
```

Value:

```
{ \
    int err_; \
    if ((err_ = fdt_rw_probe_(fdt)) != 0) \
        return err_; \
}
```

26.116.2 Function Documentation

26.116.2.1 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, @size: 64-bit values (native endian)

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:

26.116.2.2 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:

26.116.2.3 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:

26.116.2.4 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:

26.116.2.5 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:

26.116.2.6 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:

26.116.2.7 fdt_data_size_()

```
static int fdt_data_size_ (
            void * fdt )  [inline], [static]
```

Here is the caller graph for this function:

26.116.2.8 fdt_del_last_string_()

```
static void fdt_del_last_string_ (
            void * fdt,
            const char * s )  [static]
```

Here is the caller graph for this function:

26.116.2.9 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_ER↩R_BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

26.116.2.10 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_ER↩R_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_B↩ADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

26.116.2.11 fdt_delprop()

```
int fdt_delprop (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

26.116.2.12 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:

26.116.2.13 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:

26.116.2.14 fdt_pack()

```
int fdt_pack (
            void * fdt )
```

Here is the call graph for this function:

26.116.2.15 fdt_packblocks_()

```
static void fdt_packblocks_ (
            const char * old,
            char * new,
            int mem_rsv_size,
            int struct_size )  [static]
```

Here is the caller graph for this function:

26.116.2.16 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:

26.116.2.17 fdt_rw_probe_()

```
static int fdt_rw_probe_ (
            void * fdt )  [static]
```

Here is the call graph for this function:

26.116.2.18 fdt_set_name()

```
int fdt_set_name (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

26.116.2.19 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:

26.116.2.20 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:

26.116.2.21 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:

26.116.2.22 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:

26.116.2.23 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:

26.116.2.24 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:

26.117 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 (sizeof(fdt_errtable) / sizeof(fdt_errtable[0]))

Functions

- const char ∗ fdt_strerror (int errval)

Variables

- static struct fdt_errtabent fdt_errtable [ ]

26.117.1 Macro Definition Documentation

26.117.1.1 FDT_ERRTABENT

```
#define FDT_ERRTABENT(
            val ) [(val)] = { .str = #val, }
```

26.117.1.2 FDT_ERRTABSIZE

```
#define FDT_ERRTABSIZE (sizeof(fdt_errtable) / sizeof(fdt_errtable[0]))
```

26.117.2 Function Documentation

26.117.2.1 fdt_strerror()

```
const char* fdt_strerror (
            int errval )
```

26.117.3 Variable Documentation

26.117.3.1 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),
}
```

26.118 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)

26.118.1 Macro Definition Documentation

26.118.1.1 FDT_SW_PROBE

```
#define FDT_SW_PROBE(
                fdt )
```

Value:

```
{ \
    int err; \
    if ((err = fdt_sw_probe_(fdt)) != 0) \
        return err; \
}
```

26.118.1.2 FDT_SW_PROBE_MEMRSV

```
#define FDT_SW_PROBE_MEMRSV(
                fdt )
```

Value:

```
{ \
    int err; \
    if ((err = fdt_sw_probe_memrsv_(fdt)) != 0) \
        return err; \
}
```

26.118.1.3 FDT_SW_PROBE_STRUCT

```
#define FDT_SW_PROBE_STRUCT(
                fdt )
```

Value:

```
{ \
    int err; \
    if ((err = fdt_sw_probe_struct_(fdt)) != 0) \
        return err; \
}
```

26.118.2 Function Documentation

26.118.2.1 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:

26.118.2.2 fdt_add_string_()

```
static int fdt_add_string_ (
            void * fdt,
            const char * s )  [static]
```

Here is the caller graph for this function:

26.118.2.3 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:

26.118.2.4 fdt_create()

```
int fdt_create (
            void * buf,
            int bufsize )
```

fdt_create - begin creation of a new fdt @fdt: 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:

26.118.2.5 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 @fdt: 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:

26.118.2.6 fdt_del_last_string_()

```
static void fdt_del_last_string_ (
            void * fdt,
            const char * s )  [static]
```

Here is the caller graph for this function:

26.118.2.7 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:

26.118.2.8 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:

26.118.2.9 fdt_finish()

```
int fdt_finish (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.118.2.10 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:

26.118.2.11 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:

26.118.2.12 fdt_property()

```
int fdt_property (
            void * fdt,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.118.2.13 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:

26.118.2.14 fdt_resize()

```
int fdt_resize (
            void * fdt,
            void * buf,
            int bufsize )
```

26.118.2.15 fdt_sw_probe_()

```
static int fdt_sw_probe_ (
            void * fdt )  [static]
```

Here is the caller graph for this function:

26.118.2.16 fdt_sw_probe_memrsv_()

```
static int fdt_sw_probe_memrsv_ (
            void * fdt )  [static]
```

Here is the call graph for this function:

26.118.2.17 fdt_sw_probe_struct_()

```
static int fdt_sw_probe_struct_ (
            void * fdt )  [static]
```

Here is the call graph for this function:

26.118.2.18 sw_flags()

```
static uint32_t sw_flags (
            void * fdt )  [inline], [static]
```

Here is the caller graph for this function:

26.119 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)

26.119.1 Function Documentation

26.119.1.1 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:

26.119.1.2 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_ER↩R_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ER↩R_TRUNCATED, standard meanings Here is the call graph for this function:

26.119.1.3 fdt_nop_property()

```
int fdt_nop_property (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

26.119.1.4 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:

26.119.1.5 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:

26.119.1.6 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:

26.120 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_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_MAX 18
- #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_ (uint32_t version)
- static size_t fdt_header_size (const void ∗fdt)
- 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 'name' 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 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)
- 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)

: 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_BADVERS↩ ION, -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)

: 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_NOTFO↩ UND, if the requested subnode does not exist -FDT_ERR_BADOFFSET, if parentoffset did not point to an FD↩ T_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_NO↩ SPACE -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_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)
- int fdt_overlay_apply (void ∗fdt, void ∗fdto)
- const char ∗ fdt_strerror (int errval)

: name of the alias th look up

fdt_get_alias - retrieve the path referenced by a given alias @fdt: pointer to the device tree blob
fdt_get_alias() retrieves the value of a given alias. That is, the value of the property named 'name' 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_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_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 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_BADLAYO↩UT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTU↩RE, -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)
- static int fdt_setprop_inplace_cell (void *fdt, int nodeoffset, const char *name, uint32_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)

: 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_BADO↩FFSET, 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)

26.120.1 Macro Definition Documentation

26.120.1.1 fdt_appendprop_string

```
#define fdt_appendprop_string(
            fdt,
            nodeoffset,
            name,
            str ) fdt_appendprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
```

26.120.1.2 fdt_boot_cpuid_phys

```
#define fdt_boot_cpuid_phys(
            fdt ) (fdt_get_header(fdt, boot_cpuid_phys))
```

26.120.1.3 FDT_CREATE_FLAG_NO_NAME_DEDUP

```
#define FDT_CREATE_FLAG_NO_NAME_DEDUP 0x1
```

26.120.1.4 FDT_CREATE_FLAGS_ALL

```
#define FDT_CREATE_FLAGS_ALL (FDT_CREATE_FLAG_NO_NAME_DEDUP)
```

26.120.1.5 FDT_ERR_BADFLAGS

```
#define FDT_ERR_BADFLAGS 18
```

26.120.1.6 FDT_ERR_BADLAYOUT

```
#define FDT_ERR_BADLAYOUT 12
```

26.120.1.7 FDT_ERR_BADMAGIC

```
#define FDT_ERR_BADMAGIC 9
```

26.120.1.8 FDT_ERR_BADNCELLS

```
#define FDT_ERR_BADNCELLS 14
```

26.120.1.9 FDT_ERR_BADOFFSET

```
#define FDT_ERR_BADOFFSET 4
```

26.120.1.10 FDT_ERR_BADOVERLAY

```
#define FDT_ERR_BADOVERLAY 16
```

26.120.1.11 FDT_ERR_BADPATH

```
#define FDT_ERR_BADPATH 5
```

26.120.1.12 FDT_ERR_BADPHANDLE

```
#define FDT_ERR_BADPHANDLE 6
```

26.120.1.13 FDT_ERR_BADSTATE

```
#define FDT_ERR_BADSTATE 7
```

26.120.1.14 FDT_ERR_BADSTRUCTURE

```
#define FDT_ERR_BADSTRUCTURE 11
```

26.120.1.15 FDT_ERR_BADVALUE

```
#define FDT_ERR_BADVALUE 15
```

26.120.1.16 FDT_ERR_BADVERSION

```
#define FDT_ERR_BADVERSION 10
```

26.120.1.17 FDT_ERR_EXISTS

```
#define FDT_ERR_EXISTS 2
```

26.120.1.18 FDT_ERR_INTERNAL

```
#define FDT_ERR_INTERNAL 13
```

26.120.1.19 FDT_ERR_MAX

```
#define FDT_ERR_MAX 18
```

26.120.1.20 FDT_ERR_NOPHANDLES

```
#define FDT_ERR_NOPHANDLES 17
```

26.120.1.21 FDT_ERR_NOSPACE

```
#define FDT_ERR_NOSPACE 3
```

26.120.1.22 FDT_ERR_NOTFOUND

```
#define FDT_ERR_NOTFOUND 1
```

26.120.1.23 FDT_ERR_TRUNCATED

```
#define FDT_ERR_TRUNCATED 8
```

26.120.1.24 FDT_FIRST_SUPPORTED_VERSION

```
#define FDT_FIRST_SUPPORTED_VERSION 0x02
```

26.120.1.25 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))
```

fdt_for_each_property_offset - iterate over all properties of a node
@property_offset: property offset (int, lvalue) @fdt: FDT blob (const void $*$) @node: node offset (int)
This is actually a wrapper around a for loop and would be used like so:
fdt_for_each_property_offset(property, fdt, node) { Use property ... }
if ((property $<$ 0) && (property != -FDT_ERR_NOTFOUND)) { Error handling }
Note that this is implemented as a macro and property is used as iterator in the loop. The node variable can be constant or even a literal.

26.120.1.26 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.

26.120.1.27 fdt_get_header

```
#define fdt_get_header(
            fdt,
            field ) (fdt32_ld(&((const struct fdt_header *)(fdt))->field))
```

26.120.1.28 fdt_last_comp_version

```
#define fdt_last_comp_version(
            fdt ) (fdt_get_header(fdt, last_comp_version))
```

26.120.1.29 FDT_LAST_SUPPORTED_VERSION

```
#define FDT_LAST_SUPPORTED_VERSION 0x11
```

26.120.1.30 fdt_magic

```
#define fdt_magic(
            fdt ) (fdt_get_header(fdt, magic))
```

26.120.1.31 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.

26.120.1.32 FDT_MAX_PHANDLE

```
#define FDT_MAX_PHANDLE 0xfffffffe
```

26.120.1.33 fdt_off_dt_strings

```
#define fdt_off_dt_strings(
            fdt ) (fdt_get_header(fdt, off_dt_strings))
```

26.120.1.34 fdt_off_dt_struct

```
#define fdt_off_dt_struct(
            fdt ) (fdt_get_header(fdt, off_dt_struct))
```

26.120.1.35 fdt_off_mem_rsvmap

```
#define fdt_off_mem_rsvmap(
            fdt ) (fdt_get_header(fdt, off_mem_rsvmap))
```

26.120.1.36 fdt_property_string

```
#define fdt_property_string(
            fdt,
            name,
            str ) fdt_property(fdt, name, str, strlen(str)+1)
```

26.120.1.37 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); \
}
```

26.120.1.38 fdt_setprop_empty

```
#define fdt_setprop_empty(
            fdt,
            nodeoffset,
            name ) fdt_setprop((fdt), (nodeoffset), (name), NULL, 0)
```

26.120.1.39 fdt_setprop_string

```
#define fdt_setprop_string(
            fdt,
            nodeoffset,
            name,
            str ) fdt_setprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
```

26.120.1.40 fdt_size_dt_strings

```
#define fdt_size_dt_strings(
            fdt ) (fdt_get_header(fdt, size_dt_strings))
```

26.120.1.41 fdt_size_dt_struct

```
#define fdt_size_dt_struct(
            fdt ) (fdt_get_header(fdt, size_dt_struct))
```

26.120.1.42 fdt_totalsize

```
#define fdt_totalsize(
            fdt ) (fdt_get_header(fdt, totalsize))
```

26.120.1.43 fdt_version

```
#define fdt_version(
            fdt ) (fdt_get_header(fdt, version))
```

26.120.2 Function Documentation

26.120.2.1 fdt32_ld()

```
static uint32_t fdt32_ld (
            const fdt32_t * p )  [inline], [static]
```

Here is the caller graph for this function:

26.120.2.2 fdt32_st()

```
static void fdt32_st (
            void * property,
            uint32_t value )  [inline], [static]
```

Here is the caller graph for this function:

26.120.2.3 fdt64_ld()

```
static uint64_t fdt64_ld (
            const fdt64_t * p )  [inline], [static]
```

Here is the caller graph for this function:

26.120.2.4 fdt64_st()

```
static void fdt64_st (
            void * property,
            uint64_t value )  [inline], [static]
```

Here is the caller graph for this function:

26.120.2.5 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, @size: 64-bit values (native endian)

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:

26.120.2.6 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:

26.120.2.7 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:

26.120.2.8 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:

26.120.2.9 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_BA↩ DNCELLS, if the node has a badly formatted or invalid #address-cells property -FDT_ERR_BADMAGIC, -FDT_E↩ RR_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:

26.120.2.10 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:

26.120.2.11 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:

26.120.2.12 fdt_appendprop_cell()

```
static int fdt_appendprop_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

fdt_appendprop_cell - append a single cell value to a property

This is an alternative name for fdt_appendprop_u32() Here is the call graph for this function:

26.120.2.13 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:

26.120.2.14 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:

26.120.2.15 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:

26.120.2.16 fdt_check_full()

```
int fdt_check_full (
                const void * fdt,
                size_t bufsize )
```

Here is the call graph for this function:

26.120.2.17 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:

26.120.2.18 fdt_create()

```
int fdt_create (
            void * buf,
            int bufsize )
```

fdt_create - begin creation of a new fdt @fdt: 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:

26.120.2.19 fdt_create_empty_tree()

```
int fdt_create_empty_tree (
            void * buf,
            int bufsize )
```

Here is the call graph for this function:

26.120.2.20 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 @fdt: 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:

26.120.2.21 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↩ R_BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

26.120.2.22 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_ER↩ R_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_B↩ ADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

26.120.2.23 fdt_delprop()

```
int fdt_delprop (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

26.120.2.24 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:

26.120.2.25 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:

26.120.2.26 fdt_finish()

```
int fdt_finish (
            void * fdt )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.27 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:

26.120.2.28 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_ER↩R_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ER↩R_TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.29 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

Returns

offset of first subnode, or -FDT_ERR_NOTFOUND if there is none

Here is the call graph for this function:

26.120.2.30 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.

Returns: 0 on success or a negative error-code on failure Here is the call graph for this function:

26.120.2.31 fdt_get_alias()

```
const char* fdt_get_alias (
            const void * fdt,
            const char * name )
```

Here is the call graph for this function:

26.120.2.32 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:

26.120.2.33 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:

26.120.2.34 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 @address, @size: pointers to 64-bit variables

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:

26.120.2.35 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_BADSTA↩TE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.36 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. -F↩DT_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, -FD↩T_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:

26.120.2.37 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:

26.120.2.38 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:

26.120.2.39 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_BADO↩FFSET, 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:

26.120.2.40 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:

26.120.2.41 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:

26.120.2.42 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:

26.120.2.43 fdt_getprop()

```
const void* fdt_getprop (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.44 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:

26.120.2.45 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:

26.120.2.46 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:

26.120.2.47 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:

26.120.2.48 fdt_header_size()

```
static size_t fdt_header_size (
            const void * fdt )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.49 fdt_header_size_()

```
size_t fdt_header_size_ (
            uint32_t version )
```

fdt_header_size - return the size of the tree's header @fdt: pointer to a flattened device tree Here is the caller graph

for this function:

26.120.2.50 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_BAD↩ MAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the caller graph for this function:

26.120.2.51 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:

26.120.2.52 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_BA↩DMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TR↩UNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.53 fdt_next_subnode()

```
int fdt_next_subnode (
            const void * fdt,
            int offset )
```

fdt_next_subnode() - get offset of next direct subnode
After first calling fdt_first_subnode(), call this function repeatedly to get direct subnodes of a parent node.
@fdt: FDT blob @offset: Offset of previous subnode

Returns

offset of next subnode, or -FDT_ERR_NOTFOUND if there are no more subnodes

Here is the call graph for this function:

26.120.2.54 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:

26.120.2.55 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_B↩ADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph

for this function:

Here is the caller graph for this function:

26.120.2.56 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:

26.120.2.57 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_E↩ RR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.58 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, -F↩DT_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:

26.120.2.59 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_E↩ RR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

26.120.2.60 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_ER↩ R_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ER↩ R_TRUNCATED, standard meanings Here is the call graph for this function:

26.120.2.61 fdt_nop_property()

```
int fdt_nop_property (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

26.120.2.62 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:

26.120.2.63 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:

26.120.2.64 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:

26.120.2.65 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:

26.120.2.66 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:

26.120.2.67 fdt_pack()

```
int fdt_pack (
                void * fdt )
```

Here is the call graph for this function:

26.120.2.68 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_E↩RR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.69 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_ER↩R_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ER↩

R_TRUNCATED, standard meanings. Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.70 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. Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.71 fdt_property()

```
int fdt_property (
            void * fdt,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.120.2.72 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:

26.120.2.73 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:

26.120.2.74 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:

26.120.2.75 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:

26.120.2.76 fdt_resize()

```
int fdt_resize (
            void * fdt,
```

```
            void * buf,
            int bufsize )
```

26.120.2.77 fdt_set_hdr_() [1/10]

```
fdt_set_hdr_ (
            boot_cpuid_phys  )
```

26.120.2.78 fdt_set_hdr_() [2/10]

```
fdt_set_hdr_ (
            last_comp_version  )
```

26.120.2.79 fdt_set_hdr_() [3/10]

```
fdt_set_hdr_ (
            magic  )
```

26.120.2.80 fdt_set_hdr_() [4/10]

```
fdt_set_hdr_ (
            off_dt_strings  )
```

26.120.2.81 fdt_set_hdr_() [5/10]

```
fdt_set_hdr_ (
            off_dt_struct  )
```

26.120.2.82 fdt_set_hdr_() [6/10]

```
fdt_set_hdr_ (
            off_mem_rsvmap  )
```

26.120.2.83 fdt_set_hdr_() [7/10]

```
fdt_set_hdr_ (
            size_dt_strings  )
```

26.120.2.84 fdt_set_hdr_() [8/10]

```
fdt_set_hdr_ (
            size_dt_struct  )
```

26.120.2.85 fdt_set_hdr_() [9/10]

```
fdt_set_hdr_ (
            totalsize  )
```

26.120.2.86 fdt_set_hdr_() [10/10]

```
fdt_set_hdr_ (
            version  )
```

26.120.2.87 fdt_set_name()

```
int fdt_set_name (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

26.120.2.88 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:

26.120.2.89 fdt_setprop_cell()

```
static int fdt_setprop_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

fdt_setprop_cell - set a property to a single cell value

This is an alternative name for fdt_setprop_u32() Here is the call graph for this function:

26.120.2.90 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:

26.120.2.91 fdt_setprop_inplace_cell()

```
static int fdt_setprop_inplace_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

fdt_setprop_inplace_cell - change the value of a single-cell property

This is an alternative name for fdt_setprop_inplace_u32() Here is the call graph for this function:

26.120.2.92 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:

26.120.2.93 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:

26.120.2.94 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:

26.120.2.95 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:

26.120.2.96 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:

26.120.2.97 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:

26.120.2.98 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_B↩ ADNCELLS, if the node has a badly formatted or invalid #size-cells property -FDT_ERR_BADMAGIC, -FDT_E↩ RR_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:

26.120.2.99 fdt_strerror()

```
const char* fdt_strerror (
            int errval )
```

26.120.2.100 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:

26.120.2.101 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.

Returns

: 1 if the string is found in the list, 0 not found, or invalid list

Here is the caller graph for this function:

26.120.2.102 fdt_stringlist_count()

```
int fdt_stringlist_count (
            const void * fdt,
            int nodeoffset,
            const char * property )
```

Here is the call graph for this function:

26.120.2.103 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:

26.120.2.104 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:

26.120.2.105 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:

26.120.2.106 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:

26.120.2.107 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:

26.121 lib/utils/libfdt/libfdt_env.h File Reference

```
#include <sbi/sbi_string.h>
#include <sbi/sbi_types.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 strcmp sbi_strcmp
- #define strlen sbi_strlen
- #define strnlen sbi_strnlen
- #define EXTRACT_BYTE(x, n) ((unsigned long long)((uint8_t *)&x)[n])
- #define CPU_TO_FDT16(x) ((EXTRACT_BYTE(x, 0) << 8) | EXTRACT_BYTE(x, 1))
- #define CPU_TO_FDT32(x)
- #define CPU_TO_FDT64(x)

Typedefs

- typedef uint16_t FDT_BITWISE fdt16_t
- typedef uint32_t FDT_BITWISE fdt32_t
- typedef uint64_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)

26.121.1 Macro Definition Documentation

26.121.1.1 CPU_TO_FDT16

```
#define CPU_TO_FDT16(
            x ) ((EXTRACT_BYTE(x, 0) << 8) | EXTRACT_BYTE(x, 1))
```

26.121.1.2 CPU_TO_FDT32

```
#define CPU_TO_FDT32(
            x )
```

Value:

```
((EXTRACT_BYTE(x, 0) « 24) | (EXTRACT_BYTE(x, 1) « 16) | \
(EXTRACT_BYTE(x, 2) « 8) | EXTRACT_BYTE(x, 3))
```

26.121.1.3 CPU_TO_FDT64

```
#define CPU_TO_FDT64(
            x )
```

Value:

```
((EXTRACT_BYTE(x, 0) « 56) | (EXTRACT_BYTE(x, 1) « 48) | \
(EXTRACT_BYTE(x, 2) « 40) | (EXTRACT_BYTE(x, 3) « 32) | \
(EXTRACT_BYTE(x, 4) « 24) | (EXTRACT_BYTE(x, 5) « 16) | \
(EXTRACT_BYTE(x, 6) « 8) | EXTRACT_BYTE(x, 7))
```

26.121.1.4 EXTRACT_BYTE

```
#define EXTRACT_BYTE(
            x,
            n ) ((unsigned long long)((uint8_t *)&x)[n])
```

26.121.1.5 FDT_BITWISE

```
#define FDT_BITWISE
```

26.121.1.6 FDT_FORCE

```
#define FDT_FORCE
```

26.121.1.7 INT32_MAX

```
#define INT32_MAX INT_MAX
```

26.121.1.8 INT_MAX

```
#define INT_MAX ((int)(~0U >> 1))
```

26.121.1.9 memchr

```
#define memchr sbi_memchr
```

26.121.1.10 memcmp

```
#define memcmp sbi_memcmp
```

26.121.1.11 memcpy

```
#define memcpy sbi_memcpy
```

26.121.1.12 memmove

```
#define memmove sbi_memmove
```

26.121.1.13 memset

```
#define memset sbi_memset
```

26.121.1.14 strchr

```
#define strchr sbi_strchr
```

26.121.1.15 strcmp

```
#define strcmp sbi_strcmp
```

26.121.1.16 strcpy

```
#define strcpy sbi_strcpy
```

26.121.1.17 strlen

```
#define strlen sbi_strlen
```

26.121.1.18 strnlen

```
#define strnlen sbi_strnlen
```

26.121.1.19 strrchr

```
#define strrchr sbi_strrchr
```

26.121.1.20 UINT32_MAX

```
#define UINT32_MAX UINT_MAX
```

26.121.1.21 UINT_MAX

```
#define UINT_MAX ((unsigned int)~0U)
```

26.121.2 Typedef Documentation

26.121.2.1 fdt16_t

```
typedef uint16_t FDT_BITWISE fdt16_t
```

26.121.2.2 fdt32_t

```
typedef uint32_t FDT_BITWISE fdt32_t
```

26.121.2.3 fdt64_t

```
typedef uint64_t FDT_BITWISE fdt64_t
```

26.121.3 Function Documentation

26.121.3.1 cpu_to_fdt16()

```
static fdt16_t cpu_to_fdt16 (
            uint16_t x )  [inline], [static]
```

26.121.3.2 cpu_to_fdt32()

```
static fdt32_t cpu_to_fdt32 (
            uint32_t x )  [inline], [static]
```

Here is the caller graph for this function:

26.121.3.3 cpu_to_fdt64()

```
static fdt64_t cpu_to_fdt64 (
            uint64_t x )  [inline], [static]
```

Here is the caller graph for this function:

26.121.3.4 fdt16_to_cpu()

```
static uint16_t fdt16_to_cpu (
            fdt16_t x )  [inline], [static]
```

26.121.3.5 fdt32_to_cpu()

```
static uint32_t fdt32_to_cpu (
            fdt32_t x )  [inline], [static]
```

Here is the caller graph for this function:

26.121.3.6 fdt64_to_cpu()

```
static uint64_t fdt64_to_cpu (
            fdt64_t x )  [inline], [static]
```

26.122 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)

Functions

- int 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)

26.122.1 Macro Definition Documentation

26.122.1.1 FDT_ALIGN

```
#define FDT_ALIGN(
            x,
            a ) (((x) + (a) - 1) & ∼((a) - 1))
```

26.122.1.2 FDT_RO_PROBE

```
#define FDT_RO_PROBE(
                fdt )
```

Value:

```
    {                                   \
        int totalsize_;                 \
        if ((totalsize_ = fdt_ro_probe_(fdt)) < 0) \
            return totalsize_;          \
    }
```

26.122.1.3 FDT_SW_MAGIC

```
#define FDT_SW_MAGIC (~FDT_MAGIC)
```

26.122.1.4 FDT_TAGALIGN

```
#define FDT_TAGALIGN(
                x ) (FDT_ALIGN((x), FDT_TAGSIZE))
```

26.122.2 Function Documentation

26.122.2.1 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:

26.122.2.2 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:

26.122.2.3 fdt_find_string_()

```
const char* fdt_find_string_ (
            const char * strtab,
            int tabsize,
            const char * s )
```

Here is the caller graph for this function:

26.122.2.4 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:

26.122.2.5 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:

26.122.2.6 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:

26.122.2.7 fdt_offset_ptr_()

```
static const void* fdt_offset_ptr_ (
            const void * fdt,
            int offset )  [inline], [static]
```

Here is the caller graph for this function:

26.122.2.8 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:

26.122.2.9 fdt_ro_probe_()

```
int fdt_ro_probe_ (
            const void * fdt )
```

Here is the caller graph for this function:

26.123 lib/utils/reset/fdt_reset.c File Reference

```
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset.c:

Functions

- int fdt_system_reset (u32 reset_type)
- int fdt_reset_init (void)

Variables

- struct fdt_reset fdt_reset_sifive
- struct fdt_reset fdt_reset_htif
- static struct fdt_reset * reset_drivers [ ]
- static struct fdt_reset * current_driver = NULL

26.123.1 Function Documentation

26.123.1.1 fdt_reset_init()

```
int fdt_reset_init (
            void )
```

Here is the call graph for this function:

26.123.1.2 fdt_system_reset()

```
int fdt_system_reset (
            u32 reset_type )
```

26.123.2 Variable Documentation

26.123.2.1 current_driver

```
struct fdt_reset* current_driver = NULL  [static]
```

26.123.2.2 fdt_reset_htif

```
struct fdt_reset fdt_reset_htif
```

26.123.2.3 fdt_reset_sifive

```
struct fdt_reset fdt_reset_sifive
```

26.123.2.4 reset_drivers

```
struct fdt_reset* reset_drivers[]  [static]
```

Initial value:

```
= {
    &fdt_reset_sifive,
    &fdt_reset_htif,
}
```

26.124 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:

Variables

- static const struct fdt_match htif_reset_match [ ]
- struct fdt_reset fdt_reset_htif

26.124.1 Variable Documentation

26.124.1.1 fdt_reset_htif

```
struct fdt_reset fdt_reset_htif
```

Initial value:

```
= {
    .match_table = htif_reset_match,
    .system_reset = htif_system_reset
}
```

26.124.1.2 htif_reset_match

```
const struct fdt_match htif_reset_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "ucb,htif0" },
    { },
}
```

26.125 lib/utils/reset/fdt_reset_sifive.c File Reference

```
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
#include <sbi_utils/sys/sifive_test.h>
```

Include dependency graph for fdt_reset_sifive.c:

Functions

- static int sifive_test_reset_init (void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match sifive_test_reset_match [ ]
- struct fdt_reset fdt_reset_sifive

26.125.1 Function Documentation

26.125.1.1 sifive_test_reset_init()

```
static int sifive_test_reset_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

26.125.2 Variable Documentation

26.125.2.1 fdt_reset_sifive

```
struct fdt_reset fdt_reset_sifive
```

Initial value:

```
= {
    .match_table = sifive_test_reset_match,
    .init = sifive_test_reset_init,
    .system_reset = sifive_test_system_reset
}
```

26.125.2.2 sifive_test_reset_match

```
const struct fdt_match sifive_test_reset_match[] [static]
```

Initial value:

```
= {
    { .compatible = "sifive,test1" },
    { },
}
```

26.126 lib/utils/serial/fdt_serial.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
```

Include dependency graph for fdt_serial.c:

Functions

- static void dummy_putc (char ch)
- static int dummy_getc (void)
- void fdt_serial_putc (char ch)
- int fdt_serial_getc (void)
- int fdt_serial_init (void)

Variables

- struct fdt_serial fdt_serial_uart8250
- struct fdt_serial fdt_serial_sifive
- struct fdt_serial fdt_serial_htif
- struct fdt_serial fdt_serial_shakti
- static struct fdt_serial * serial_drivers [ ]

- static struct fdt_serial dummy
- static struct fdt_serial ∗ current_driver = &dummy

26.126.1 Function Documentation

26.126.1.1 dummy_getc()

```
static int dummy_getc (
            void ) [static]
```

26.126.1.2 dummy_putc()

```
static void dummy_putc (
            char ch ) [static]
```

26.126.1.3 fdt_serial_getc()

```
int fdt_serial_getc (
            void )
```

26.126.1.4 fdt_serial_init()

```
int fdt_serial_init (
            void )
```

Here is the call graph for this function:

26.126.1.5 fdt_serial_putc()

```
void fdt_serial_putc (
            char ch )
```

26.126.2 Variable Documentation

26.126.2.1 current_driver

```
struct fdt_serial* current_driver = &dummy [static]
```

26.126.2.2 dummy

```
struct fdt_serial dummy  [static]
```

Initial value:

```
= {
    .match_table = NULL,
    .init = NULL,
    .putc = dummy_putc,
    .getc = dummy_getc,
}
```

26.126.2.3 fdt_serial_htif

```
struct fdt_serial fdt_serial_htif
```

26.126.2.4 fdt_serial_shakti

```
struct fdt_serial fdt_serial_shakti
```

26.126.2.5 fdt_serial_sifive

```
struct fdt_serial fdt_serial_sifive
```

26.126.2.6 fdt_serial_uart8250

```
struct fdt_serial fdt_serial_uart8250
```

26.126.2.7 serial_drivers

```
struct fdt_serial* serial_drivers[]  [static]
```

Initial value:

```
= {
    &fdt_serial_uart8250,
    &fdt_serial_sifive,
    &fdt_serial_htif,
    &fdt_serial_shakti,
}
```

26.127 lib/utils/serial/fdt_serial_htif.c File Reference

```
#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:

Variables

- static const struct fdt_match serial_htif_match [ ]
- struct fdt_serial fdt_serial_htif

26.127.1 Variable Documentation

26.127.1.1 fdt_serial_htif

```
struct fdt_serial fdt_serial_htif
```

Initial value:

```
= {
    .match_table = serial_htif_match,
    .init = NULL,
    .getc = htif_getc,
    .putc = htif_putc
}
```

26.127.1.2 serial_htif_match

```
const struct fdt_match serial_htif_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "ucb,htif0" },
    { },
}
```

26.128 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 (void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match serial_shakti_match [ ]
- struct fdt_serial fdt_serial_shakti

26.128.1 Function Documentation

26.128.1.1 serial_shakti_init()

```
static int serial_shakti_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

26.128.2 Variable Documentation

26.128.2.1 fdt_serial_shakti

```
struct fdt_serial fdt_serial_shakti
```

Initial value:

```
= {
    .match_table = serial_shakti_match,
    .init = serial_shakti_init,
    .getc = shakti_uart_getc,
```

```
    .putc = shakti_uart_putc
}
```

26.128.2.2 serial_shakti_match

```
const struct fdt_match serial_shakti_match[] [static]
```

Initial value:

```
= {
    { .compatible = "shakti,uart0" },
    { },
}
```

26.129 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 (void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match serial_sifive_match [ ]
- struct fdt_serial fdt_serial_sifive

26.129.1 Function Documentation

26.129.1.1 serial_sifive_init()

```
static int serial_sifive_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match ) [static]
```

Here is the call graph for this function:

26.129.2 Variable Documentation

26.129.2.1 fdt_serial_sifive

```
struct fdt_serial fdt_serial_sifive
```

Initial value:

```
= {
    .match_table = serial_sifive_match,
    .init = serial_sifive_init,
    .getc = sifive_uart_getc,
    .putc = sifive_uart_putc
}
```

26.129.2.2 serial_sifive_match

```
const struct fdt_match serial_sifive_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "sifive,fu540-c000-uart" },
    { .compatible = "sifive,uart0" },
    { },
}
```

26.130 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 (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_uart8250_match [ ]
- struct fdt_serial fdt_serial_uart8250

26.130.1 Function Documentation

26.130.1.1 serial_uart8250_init()

```
static int serial_uart8250_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

26.130.2 Variable Documentation

26.130.2.1 fdt_serial_uart8250

```
struct fdt_serial fdt_serial_uart8250
```

Initial value:

```
= {
    .match_table = serial_uart8250_match,
    .init = serial_uart8250_init,
    .getc = uart8250_getc,
    .putc = uart8250_putc
}
```

26.130.2.2 serial_uart8250_match

```
const struct fdt_match serial_uart8250_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "ns16550" },
    { .compatible = "ns16550a" },
    { },
}
```

26.131 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

Functions

- void shakti_uart_putc (char ch)
- int shakti_uart_getc (void)
- int shakti_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile void * uart_base

26.131.1 Macro Definition Documentation

26.131.1.1 REG_BAUD

```
#define REG_BAUD 0x00
```

26.131.1.2 REG_CONTROL

```
#define REG_CONTROL 0x14
```

26.131.1.3 REG_DELAY

```
#define REG_DELAY 0x10
```

26.131.1.4 REG_INT_EN

```
#define REG_INT_EN 0x18
```

26.131.1.5 REG_IQ_CYCLES

```
#define REG_IQ_CYCLES 0x1C
```

26.131.1.6 REG_RX

```
#define REG_RX 0x08
```

26.131.1.7 REG_RX_THRES

```
#define REG_RX_THRES 0x20
```

26.131.1.8 REG_STATUS

```
#define REG_STATUS 0x0C
```

26.131.1.9 REG_TX

```
#define REG_TX 0x04
```

26.131.2 Function Documentation

26.131.2.1 shakti_uart_getc()

```
int shakti_uart_getc (
            void  )
```

26.131.2.2 shakti_uart_init()

```
int shakti_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the caller graph for this function:

26.131.2.3 shakti_uart_putc()

```
void shakti_uart_putc (
            char ch )
```

26.131.3 Variable Documentation

26.131.3.1 uart_base

```
volatile void* uart_base  [static]
```

26.132 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)
- void sifive_uart_putc (char ch)
- int sifive_uart_getc (void)
- int sifive_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

26.132.1 Macro Definition Documentation

26.132.1.1 UART_REG_DIV

```
#define UART_REG_DIV 6
```

26.132.1.2 UART_REG_IE

```
#define UART_REG_IE 4
```

26.132.1.3 UART_REG_IP

```
#define UART_REG_IP 5
```

26.132.1.4 UART_REG_RXCTRL

```
#define UART_REG_RXCTRL 3
```

26.132.1.5 UART_REG_RXFIFO

```
#define UART_REG_RXFIFO 1
```

26.132.1.6 UART_REG_TXCTRL

```
#define UART_REG_TXCTRL 2
```

26.132.1.7 UART_REG_TXFIFO

```
#define UART_REG_TXFIFO 0
```

26.132.1.8 UART_RXCTRL_RXEN

```
#define UART_RXCTRL_RXEN 0x1
```

26.132.1.9 UART_RXFIFO_DATA

```
#define UART_RXFIFO_DATA 0x000000ff
```

26.132.1.10 UART_RXFIFO_EMPTY

```
#define UART_RXFIFO_EMPTY 0x80000000
```

26.132.1.11 UART_TXCTRL_TXEN

```
#define UART_TXCTRL_TXEN 0x1
```

26.132.1.12 UART_TXFIFO_FULL

```
#define UART_TXFIFO_FULL 0x80000000
```

26.132.2 Function Documentation

26.132.2.1 get_reg()

```
static u32 get_reg (
            u32 num )  [static]
```

Here is the caller graph for this function:

26.132.2.2 set_reg()

```
static void set_reg (
            u32 num,
            u32 val )  [static]
```

Here is the caller graph for this function:

26.132.2.3 sifive_uart_getc()

```
int sifive_uart_getc (
            void  )
```

Here is the call graph for this function:

26.132.2.4 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:

26.132.2.5 sifive_uart_putc()

```
void sifive_uart_putc (
            char ch )
```

Here is the call graph for this function:

26.132.2.6 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:

26.132.3 Variable Documentation

26.132.3.1 uart_base

```
volatile void* uart_base [static]
```

26.132.3.2 uart_baudrate

```
u32 uart_baudrate [static]
```

26.132.3.3 uart_in_freq

```
u32 uart_in_freq [static]
```

26.133 lib/utils/serial/uart8250.c File Reference

```
#include <sbi/riscv_io.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)
- void uart8250_putc (char ch)
- int uart8250_getc (void)
- int uart8250_init (unsigned long base, u32 in_freq, u32 baudrate, u32 reg_shift, u32 reg_width)

Variables

- static volatile void ∗ uart8250_base
- static u32 uart8250_in_freq
- static u32 uart8250_baudrate
- static u32 uart8250_reg_width
- static u32 uart8250_reg_shift

26.133.1 Macro Definition Documentation

26.133.1.1 UART_DLL_OFFSET

```
#define UART_DLL_OFFSET 0 /* Out:  Divisor Latch Low */
```

26.133.1.2 UART_DLM_OFFSET

```
#define UART_DLM_OFFSET 1 /* Out:  Divisor Latch High */
```

26.133.1.3 UART_FCR_OFFSET

```
#define UART_FCR_OFFSET 2 /* Out:  FIFO Control Register */
```

26.133.1.4 UART_IER_OFFSET

```
#define UART_IER_OFFSET 1 /* I/O: Interrupt Enable Register */
```

26.133.1.5 UART_IIR_OFFSET

```
#define UART_IIR_OFFSET 2 /* I/O: Interrupt Identification Register */
```

26.133.1.6 UART_LCR_OFFSET

```
#define UART_LCR_OFFSET 3 /* Out:  Line Control Register */
```

26.133.1.7 UART_LSR_BI

```
#define UART_LSR_BI 0x10 /* Break interrupt indicator */
```

26.133.1.8 UART_LSR_BRK_ERROR_BITS

```
#define UART_LSR_BRK_ERROR_BITS 0x1E /* BI, FE, PE, OE bits */
```

26.133.1.9 UART_LSR_DR

```
#define UART_LSR_DR 0x01 /* Receiver data ready */
```

26.133.1.10 UART_LSR_FE

```
#define UART_LSR_FE 0x08 /* Frame error indicator */
```

26.133.1.11 UART_LSR_FIFOE

```
#define UART_LSR_FIFOE 0x80 /* Fifo error */
```

26.133.1.12 UART_LSR_OE

```
#define UART_LSR_OE 0x02 /* Overrun error indicator */
```

26.133.1.13 UART_LSR_OFFSET

```
#define UART_LSR_OFFSET 5 /* In:  Line Status Register */
```

26.133.1.14 UART_LSR_PE

```
#define UART_LSR_PE 0x04 /* Parity error indicator */
```

26.133.1.15 UART_LSR_TEMT

```
#define UART_LSR_TEMT 0x40 /* Transmitter empty */
```

26.133.1.16 UART_LSR_THRE

```
#define UART_LSR_THRE 0x20 /* Transmit-hold-register empty */
```

26.133.1.17 UART_MCR_OFFSET

```
#define UART_MCR_OFFSET 4 /* Out:  Modem Control Register */
```

26.133.1.18 UART_MDR1_OFFSET

```
#define UART_MDR1_OFFSET 8 /* I/O: Mode Register */
```

26.133.1.19 UART_MSR_OFFSET

```
#define UART_MSR_OFFSET 6 /* In:  Modem Status Register */
```

26.133.1.20 UART_RBR_OFFSET

```
#define UART_RBR_OFFSET 0 /* In:  Recieve Buffer Register */
```

26.133.1.21 UART_SCR_OFFSET

```
#define UART_SCR_OFFSET 7 /* I/O: Scratch Register */
```

26.133.1.22 UART_THR_OFFSET

```
#define UART_THR_OFFSET 0 /* Out:  Transmitter Holding Register */
```

26.133.2 Function Documentation

26.133.2.1 get_reg()

```
static u32 get_reg (
            u32 num )  [static]
```

Here is the caller graph for this function:

26.133.2.2 set_reg()

```
static void set_reg (
            u32 num,
            u32 val )  [static]
```

Here is the caller graph for this function:

26.133.2.3 uart8250_getc()

```
int uart8250_getc (
            void  )
```

Here is the call graph for this function:

26.133.2.4 uart8250_init()

```
int uart8250_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate,
            u32 reg_shift,
            u32 reg_width )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.133.2.5 uart8250_putc()

```
void uart8250_putc (
            char ch )
```

Here is the call graph for this function:

26.133.3 Variable Documentation

26.133.3.1 uart8250_base

```
volatile void* uart8250_base [static]
```

26.133.3.2 uart8250_baudrate

```
u32 uart8250_baudrate [static]
```

26.133.3.3 uart8250_in_freq

```
u32 uart8250_in_freq [static]
```

26.133.3.4 uart8250_reg_shift

```
u32 uart8250_reg_shift [static]
```

26.133.3.5 uart8250_reg_width

```
u32 uart8250_reg_width [static]
```

26.134 lib/utils/sys/clint.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hartmask.h>
#include <sbi_utils/sys/clint.h>
```

Include dependency graph for clint.c:

Macros

- #define CLINT_IPI_OFF 0
- #define CLINT_TIME_CMP_OFF 0x4000
- #define CLINT_TIME_VAL_OFF 0xbff8

Functions

- void clint_ipi_send (u32 target_hart)
- void clint_ipi_clear (u32 target_hart)
- int clint_warm_ipi_init (void)
- int clint_cold_ipi_init (struct clint_data *clint)
- static u64 clint_time_rd64 (volatile u64 *addr)
- static void clint_time_wr64 (u64 value, volatile u64 *addr)
- static u64 clint_time_rd32 (volatile u64 *addr)
- static void clint_time_wr32 (u64 value, volatile u64 *addr)
- u64 clint_timer_value (void)
- void clint_timer_event_stop (void)
- void clint_timer_event_start (u64 next_event)
- int clint_warm_timer_init (void)
- int clint_cold_timer_init (struct clint_data *clint, struct clint_data *reference)

Variables

- static struct clint_data * clint_ipi_hartid2data [SBI_HARTMASK_MAX_BITS]
- static struct clint_data * clint_timer_hartid2data [SBI_HARTMASK_MAX_BITS]

26.134.1 Macro Definition Documentation

26.134.1.1 CLINT_IPI_OFF

```
#define CLINT_IPI_OFF 0
```

26.134.1.2 CLINT_TIME_CMP_OFF

```
#define CLINT_TIME_CMP_OFF 0x4000
```

26.134.1.3 CLINT_TIME_VAL_OFF

```
#define CLINT_TIME_VAL_OFF 0xbff8
```

26.134.2 Function Documentation

26.134.2.1 clint_cold_ipi_init()

```
int clint_cold_ipi_init (
            struct clint_data * clint )
```

Here is the caller graph for this function:

26.134.2.2 clint_cold_timer_init()

```
int clint_cold_timer_init (
            struct clint_data * clint,
            struct clint_data * reference )
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.134.2.3 clint_ipi_clear()

```
void clint_ipi_clear (
            u32 target_hart )
```

Here is the caller graph for this function:

26.134.2.4 clint_ipi_send()

```
void clint_ipi_send (
            u32 target_hart )
```

26.134.2.5 clint_time_rd32()

```
static u64 clint_time_rd32 (
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:

26.134.2.6 clint_time_rd64()

```
static u64 clint_time_rd64 (
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:

26.134.2.7 clint_time_wr32()

```
static void clint_time_wr32 (
            u64 value,
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:

26.134.2.8 clint_time_wr64()

```
static void clint_time_wr64 (
            u64 value,
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:

26.134.2.9 clint_timer_event_start()

```
void clint_timer_event_start (
            u64 next_event )
```

26.134.2.10 clint_timer_event_stop()

```
void clint_timer_event_stop (
            void  )
```

26.134.2.11 clint_timer_value()

```
u64 clint_timer_value (
            void  )
```

26.134.2.12 clint_warm_ipi_init()

```
int clint_warm_ipi_init (
            void  )
```

Here is the call graph for this function:

26.134.2.13 clint_warm_timer_init()

```
int clint_warm_timer_init (
            void  )
```

Here is the call graph for this function:

26.134.3 Variable Documentation

26.134.3.1 clint_ipi_hartid2data

```
struct clint_data* clint_ipi_hartid2data[SBI_HARTMASK_MAX_BITS]  [static]
```

26.134.3.2 clint_timer_hartid2data

```
struct clint_data* clint_timer_hartid2data[SBI_HARTMASK_MAX_BITS]  [static]
```

26.135 lib/utils/sys/htif.c File Reference

```
#include <sbi/riscv_locks.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 void __check_fromhost ()
- static void __set_tohost (uint64_t dev, uint64_t cmd, uint64_t data)
- void htif_putc (char ch)
- int htif_getc (void)
- int htif_system_reset (u32 type)

Variables

- static int htif_console_buf
- static spinlock_t htif_lock = SPIN_LOCK_INITIALIZER

26.135.1 Macro Definition Documentation

26.135.1.1 FROMHOST_CMD

```
#define FROMHOST_CMD(
            fromhost_value ) ((uint64_t)((fromhost_value) >> HTIF_CMD_SHIFT) & HTIF_CMD_MASK)
```

26.135.1.2 FROMHOST_DATA

```
#define FROMHOST_DATA(
            fromhost_value ) ((uint64_t)((fromhost_value) >> HTIF_DATA_SHIFT) & HTIF_DATA_MASK)
```

26.135.1.3 FROMHOST_DEV

```
#define FROMHOST_DEV(
            fromhost_value ) ((uint64_t)((fromhost_value) >> HTIF_DEV_SHIFT) & HTIF_DEV_MASK)
```

26.135.1.4 HTIF_CMD_BITS

```
#define HTIF_CMD_BITS 8
```

26.135.1.5 HTIF_CMD_MASK

```
#define HTIF_CMD_MASK ((1ULL << HTIF_CMD_BITS) - 1)
```

26.135.1.6 HTIF_CMD_SHIFT

```
#define HTIF_CMD_SHIFT 48
```

26.135.1.7 HTIF_CONSOLE_CMD_GETC

```
#define HTIF_CONSOLE_CMD_GETC 0
```

26.135.1.8 HTIF_CONSOLE_CMD_PUTC

```
#define HTIF_CONSOLE_CMD_PUTC 1
```

26.135.1.9 HTIF_DATA_BITS

```
#define HTIF_DATA_BITS 48
```

26.135.1.10 HTIF_DATA_MASK

```
#define HTIF_DATA_MASK ((1ULL << HTIF_DATA_BITS) - 1)
```

26.135.1.11 HTIF_DATA_SHIFT

```
#define HTIF_DATA_SHIFT 0
```

26.135.1.12 HTIF_DEV_BITS

```
#define HTIF_DEV_BITS 8
```

26.135.1.13 HTIF_DEV_CONSOLE

```
#define HTIF_DEV_CONSOLE 1
```

26.135.1.14 HTIF_DEV_MASK

```
#define HTIF_DEV_MASK ((1ULL << HTIF_DEV_BITS) - 1)
```

26.135.1.15 HTIF_DEV_SHIFT

```
#define HTIF_DEV_SHIFT 56
```

26.135.1.16 HTIF_DEV_SYSTEM

```
#define HTIF_DEV_SYSTEM 0
```

26.135.1.17 PK_SYS_write

```
#define PK_SYS_write 64
```

26.135.1.18 TOHOST_CMD

```
#define TOHOST_CMD(
              dev,
              cmd,
              payload )
```

Value:

```
({ \
if ((dev) || (cmd)) __builtin_trap(); \
(payload); })
```

26.135.2 Function Documentation

26.135.2.1 __attribute__()

```
volatile uint64_t fromhost __attribute__ (
            (section(".htif"))  )
```

26.135.2.2 __check_fromhost()

```
static void __check_fromhost ( )  [static]
```

Here is the caller graph for this function:

26.135.2.3 __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:

26.135.2.4 htif_getc()

```
int htif_getc (
            void  )
```

Here is the call graph for this function:

26.135.2.5 htif_putc()

```
void htif_putc (
            char ch )
```

Here is the call graph for this function:

26.135.2.6 htif_system_reset()

```
int htif_system_reset (
            u32 type )
```

26.135.3 Variable Documentation

26.135.3.1 htif_console_buf

```
int htif_console_buf [static]
```

26.135.3.2 htif_lock

```
spinlock_t htif_lock = SPIN_LOCK_INITIALIZER [static]
```

26.136 lib/utils/sys/sifive_test.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_platform.h>
#include <sbi_utils/sys/sifive_test.h>
```

Include dependency graph for sifive_test.c:

Macros

- #define FINISHER_FAIL 0x3333
- #define FINISHER_PASS 0x5555
- #define FINISHER_RESET 0x7777

Functions

- int sifive_test_system_reset (u32 type)
- int sifive_test_init (unsigned long base)

Variables

- static void * sifive_test_base

26.136.1 Macro Definition Documentation

26.136.1.1 FINISHER_FAIL

```
#define FINISHER_FAIL 0x3333
```

26.136.1.2 FINISHER_PASS

```
#define FINISHER_PASS 0x5555
```

26.136.1.3 FINISHER_RESET

```
#define FINISHER_RESET 0x7777
```

26.136.2 Function Documentation

26.136.2.1 sifive_test_init()

```
int sifive_test_init (
            unsigned long base )
```

Here is the caller graph for this function:

26.136.2.2 sifive_test_system_reset()

```
int sifive_test_system_reset (
            u32 type )
```

26.136.3 Variable Documentation

26.136.3.1 sifive_test_base

```
void* sifive_test_base [static]
```

26.137 lib/utils/timer/fdt_timer.c File Reference

```
#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

- static u64 dummy_value (void)
- static void dummy_event_stop (void)
- static void dummy_event_start (u64 next_event)
- u64 fdt_timer_value (void)
- void fdt_timer_event_stop (void)
- void fdt_timer_event_start (u64 next_event)
- void fdt_timer_exit (void)
- static int fdt_timer_warm_init (void)
- static int fdt_timer_cold_init (void)
- int fdt_timer_init (bool cold_boot)

Variables

- struct fdt_timer fdt_timer_clint
- static struct fdt_timer ∗ timer_drivers [ ]
- static struct fdt_timer dummy
- static struct fdt_timer ∗ current_driver = &dummy

26.137.1 Function Documentation

26.137.1.1 dummy_event_start()

```
static void dummy_event_start (
            u64 next_event ) [static]
```

26.137.1.2 dummy_event_stop()

```
static void dummy_event_stop (
            void  )  [static]
```

26.137.1.3 dummy_value()

```
static u64 dummy_value (
            void  )  [static]
```

26.137.1.4 fdt_timer_cold_init()

```
static int fdt_timer_cold_init (
            void  )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

26.137.1.5 fdt_timer_event_start()

```
void fdt_timer_event_start (
            u64 next_event )
```

26.137.1.6 fdt_timer_event_stop()

```
void fdt_timer_event_stop (
            void  )
```

26.137.1.7 fdt_timer_exit()

```
void fdt_timer_exit (
            void  )
```

26.137.1.8 fdt_timer_init()

```
int fdt_timer_init (
            bool cold_boot )
```

Here is the call graph for this function:

26.137.1.9 fdt_timer_value()

```
u64 fdt_timer_value (
            void  )
```

26.137.1.10 fdt_timer_warm_init()

```
static int fdt_timer_warm_init (
            void  ) [static]
```

Here is the caller graph for this function:

26.137.2 Variable Documentation

26.137.2.1 current_driver

```
struct fdt_timer* current_driver = &dummy  [static]
```

26.137.2.2 dummy

```
struct fdt_timer dummy  [static]
```

Initial value:

```
= {
    .match_table = NULL,
    .cold_init = NULL,
    .warm_init = NULL,
    .exit = NULL,
    .value = dummy_value,
    .event_stop = dummy_event_stop,
    .event_start = dummy_event_start
}
```

26.137.2.3 fdt_timer_clint

```
struct fdt_timer fdt_timer_clint
```

26.137.2.4 timer_drivers

```
struct fdt_timer* timer_drivers[] [static]
```

Initial value:

```
= {
    &fdt_timer_clint
}
```

26.138 lib/utils/timer/fdt_timer_clint.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/timer/fdt_timer.h>
#include <sbi_utils/sys/clint.h>
```

Include dependency graph for fdt_timer_clint.c:

Macros

- #define CLINT_TIMER_MAX_NR 16

Functions

- static int timer_clint_cold_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static unsigned long clint_timer_count = 0
- static struct clint_data clint_timer [CLINT_TIMER_MAX_NR]
- static const struct fdt_match timer_clint_match [ ]
- struct fdt_timer fdt_timer_clint

26.138.1 Macro Definition Documentation

26.138.1.1 CLINT_TIMER_MAX_NR

```
#define CLINT_TIMER_MAX_NR 16
```

26.138.2 Function Documentation

26.138.2.1 timer_clint_cold_init()

```
static int timer_clint_cold_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:

26.138.3 Variable Documentation

26.138.3.1 clint_timer

```
struct clint_data clint_timer[CLINT_TIMER_MAX_NR]  [static]
```

26.138.3.2 clint_timer_count

```
unsigned long clint_timer_count = 0  [static]
```

26.138.3.3 fdt_timer_clint

```
struct fdt_timer fdt_timer_clint
```

Initial value:

```
= {
    .match_table = timer_clint_match,
    .cold_init = timer_clint_cold_init,
    .warm_init = clint_warm_timer_init,
    .exit = NULL,
    .value = clint_timer_value,
    .event_stop = clint_timer_event_stop,
    .event_start = clint_timer_event_start,
}
```

26.138.3.4 timer_clint_match

```
const struct fdt_match timer_clint_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "riscv,clint0" },
    { },
}
```

26.139 README.md File Reference

Index

_AC
sbi_const.h, 176
_AT
sbi_const.h, 176
_BITUL
sbi_const.h, 176
_BITULL
sbi_const.h, 176
_UL
sbi_const.h, 176
_ULL
sbi_const.h, 176
__AC
sbi_const.h, 175
__ASM_STR
riscv_asm.h, 86
__NOP
riscv_atomic.c, 341
__NOT
riscv_atomic.c, 341
__RISCV_SPIN_UNLOCKED
riscv_locks.h, 155
__STR
sbi_const.h, 176
__atomic_op_bit
riscv_atomic.c, 340
__atomic_op_bit_ord
riscv_atomic.c, 340
__attribute__
htif.c, 681
sbi_hart.c, 385
sbi_hart.h, 201
__axchg
riscv_atomic.c, 340
__bitmap_and
sbi_bitmap.c, 349
sbi_bitmap.h, 160
__bitmap_or
sbi_bitmap.c, 349
sbi_bitmap.h, 160
__bitmap_xor
sbi_bitmap.c, 349
sbi_bitmap.h, 160
__change_bit
sbi_bitops.h, 167
__check_fromhost
htif.c, 681
__clear_bit
sbi_bitops.h, 167
__cmpxchg
riscv_atomic.c, 341
__detect_pmp
sbi_hart.c, 384
__ffs
sbi_bitops.h, 167
__fls
sbi_bitops.h, 168
__io_ar
riscv_io.h, 151
__io_aw
riscv_io.h, 151
__io_br
riscv_io.h, 151
__io_bw
riscv_io.h, 151
__io_rar
riscv_io.h, 151
__io_raw
riscv_io.h, 151
__io_rbr
riscv_io.h, 152
__io_rbw
riscv_io.h, 152
__noreturn
sbi_types.h, 294
__packed
fw_dynamic.h, 84
sbi_platform.h, 266
sbi_scratch.h, 272
sbi_trap.h, 292
sbi_types.h, 294
__printf
sbi_console.h, 172
__raw_readb
riscv_io.h, 154
__raw_readl
riscv_io.h, 154
__raw_readq
riscv_io.h, 154
__raw_readw
riscv_io.h, 154
__raw_writeb
riscv_io.h, 154
__raw_writel
riscv_io.h, 154
__raw_writeq
riscv_io.h, 154
__raw_writew

riscv_io.h, 154
__sbi_expected_trap
sbi_hart.c, 385
__sbi_expected_trap_hext
sbi_hart.c, 385
__sbi_fifo_enqueue
sbi_fifo.c, 377
__sbi_fifo_is_empty
sbi_fifo.c, 378
__sbi_fifo_is_full
sbi_fifo.c, 378
__sbi_fifo_reset
sbi_fifo.c, 378
__sbi_hfence_gvma_all
sbi_hfence.h, 214
__sbi_hfence_gvma_gpa
sbi_hfence.h, 215
__sbi_hfence_gvma_vmid
sbi_hfence.h, 215
__sbi_hfence_gvma_vmid_gpa
sbi_hfence.h, 215
__sbi_hfence_vvma_all
sbi_hfence.h, 215
__sbi_hfence_vvma_asid
sbi_hfence.h, 216
__sbi_hfence_vvma_asid_va
sbi_hfence.h, 216
__sbi_hfence_vvma_va
sbi_hfence.h, 216
__sbi_list_add
sbi_list.h, 237
__sbi_list_del
sbi_list.h, 237
__sbi_list_del_entry
sbi_list.h, 238
__sbi_tlb_range_check
sbi_tlb.c, 437
__set_bit
sbi_bitops.h, 168
__set_tohost
htif.c, 681
__smp_load_acquire
riscv_barrier.h, 96
__smp_store_release
riscv_barrier.h, 97
__test_and_clear_bit
sbi_bitops.h, 168
__test_and_set_bit
sbi_bitops.h, 169
__test_bit
sbi_bitops.h, 169
__xchg
riscv_atomic.c, 341

a0
sbi_trap_regs, 75
a1
sbi_trap_regs, 75
a2
sbi_trap_regs, 75
a3
sbi_trap_regs, 76
a4
sbi_trap_regs, 76
a5
sbi_trap_regs, 76
a6
sbi_trap_regs, 76
a7
sbi_trap_regs, 76
addr
clint_data, 53
platform_uart_data, 63
plic_data, 63
address
fdt_reserve_entry, 59
arg1
sbi_hart.h, 207
array_size
sbi_types.h, 294
asid
sbi_tlb_info, 73
atomic_add_return
riscv_atomic.c, 342
riscv_atomic.h, 93
atomic_clear_bit
riscv_atomic.c, 342
riscv_atomic.h, 93
atomic_cmpxchg
riscv_atomic.c, 342
riscv_atomic.h, 93
ATOMIC_INIT
riscv_atomic.h, 92
ATOMIC_INITIALIZER
riscv_atomic.h, 92
atomic_raw_clear_bit
riscv_atomic.c, 342
riscv_atomic.h, 93
atomic_raw_set_bit
riscv_atomic.c, 343
riscv_atomic.h, 94
atomic_raw_xchg_uint
riscv_atomic.c, 343
riscv_atomic.h, 94
atomic_raw_xchg_ulong
riscv_atomic.c, 343
riscv_atomic.h, 94
atomic_read
riscv_atomic.c, 344
riscv_atomic.h, 95
atomic_set_bit
riscv_atomic.c, 344
riscv_atomic.h, 95
atomic_sub_return
riscv_atomic.c, 344
riscv_atomic.h, 95
atomic_t, 53

counter, 53
atomic_write
riscv_atomic.c, 345
riscv_atomic.h, 96
atomic_xchg
riscv_atomic.c, 345
riscv_atomic.h, 96
avail
sbi_fifo, 65
axchg
riscv_atomic.c, 341

BANNER
sbi_init.c, 403
baud
platform_uart_data, 63
BIT
sbi_bitops.h, 165
BIT_MASK
sbi_bitops.h, 165
BIT_WORD
sbi_bitops.h, 165
BIT_WORD_OFFSET
sbi_bitops.h, 166
bitmap_and
sbi_bitmap.h, 161
bitmap_clear
sbi_bitmap.h, 161
bitmap_copy
sbi_bitmap.h, 162
bitmap_estimate_size
sbi_bitmap.h, 162
bitmap_fill
sbi_bitmap.h, 162
BITMAP_FIRST_WORD_MASK
sbi_bitmap.h, 159
BITMAP_LAST_WORD_MASK
sbi_bitmap.h, 159
bitmap_or
sbi_bitmap.h, 162
bitmap_set
sbi_bitmap.h, 163
bitmap_xor
sbi_bitmap.h, 163
bitmap_zero
sbi_bitmap.h, 163
bitmap_zero_except
sbi_bitmap.h, 164
BITOP_WORD
sbi_bitops.c, 350
BITS_TO_LONGS
sbi_bitops.h, 166
bool
sbi_types.h, 296
boot_cpuid_phys
fdt_header, 55
boot_hart
fw_dynamic_info, 62

cause
sbi_trap_info, 74
CAUSE_BREAKPOINT
riscv_encoding.h, 107
CAUSE_FETCH_ACCESS
riscv_encoding.h, 107
CAUSE_FETCH_GUEST_PAGE_FAULT
riscv_encoding.h, 107
CAUSE_FETCH_PAGE_FAULT
riscv_encoding.h, 107
CAUSE_HYPERVISOR_ECALL
riscv_encoding.h, 107
CAUSE_ILLEGAL_INSTRUCTION
riscv_encoding.h, 107
CAUSE_LOAD_ACCESS
riscv_encoding.h, 108
CAUSE_LOAD_GUEST_PAGE_FAULT
riscv_encoding.h, 108
CAUSE_LOAD_PAGE_FAULT
riscv_encoding.h, 108
CAUSE_MACHINE_ECALL
riscv_encoding.h, 108
CAUSE_MISALIGNED_FETCH
riscv_encoding.h, 108
CAUSE_MISALIGNED_LOAD
riscv_encoding.h, 108
CAUSE_MISALIGNED_STORE
riscv_encoding.h, 108
CAUSE_STORE_ACCESS
riscv_encoding.h, 108
CAUSE_STORE_GUEST_PAGE_FAULT
riscv_encoding.h, 108
CAUSE_STORE_PAGE_FAULT
riscv_encoding.h, 108
CAUSE_SUPERVISOR_ECALL
riscv_encoding.h, 108
CAUSE_USER_ECALL
riscv_encoding.h, 108
CAUSE_VIRTUAL_INST_FAULT
riscv_encoding.h, 109
check_block_
fdt.c, 480
check_off_
fdt.c, 481
CLAMP
sbi_types.h, 294
clear
fdt_ipi, 56
clint.c
clint_cold_ipi_init, 674
clint_cold_timer_init, 674
clint_ipi_clear, 675
clint_ipi_hartid2data, 678
CLINT_IPI_OFF, 674
clint_ipi_send, 675
CLINT_TIME_CMP_OFF, 674
clint_time_rd32, 675
clint_time_rd64, 676

CLINT_TIME_VAL_OFF, 674
clint_time_wr32, 676
clint_time_wr64, 676
clint_timer_event_start, 677
clint_timer_event_stop, 677
clint_timer_hartid2data, 678
clint_timer_value, 677
clint_warm_ipi_init, 677
clint_warm_timer_init, 677
clint.h
clint_cold_ipi_init, 327
clint_cold_timer_init, 328
clint_ipi_clear, 328
clint_ipi_send, 329
clint_timer_event_start, 329
clint_timer_event_stop, 329
clint_timer_value, 329
clint_warm_ipi_init, 329
clint_warm_timer_init, 329
clint_cold_ipi_init
clint.c, 674
clint.h, 327
clint_cold_timer_init
clint.c, 674
clint.h, 328
clint_data, 53
addr, 53
first_hartid, 54
hart_count, 54
has_64bit_mmio, 54
ipi, 54
time_cmp, 54
time_delta, 54
time_delta_computed, 54
time_delta_reference, 54
time_rd, 54
time_val, 54
time_wr, 54
clint_ipi
fdt_ipi_clint.c, 470
clint_ipi_clear
clint.c, 675
clint.h, 328
clint_ipi_count
fdt_ipi_clint.c, 470
clint_ipi_hartid2data
clint.c, 678
CLINT_IPI_MAX_NR
fdt_ipi_clint.c, 469
CLINT_IPI_OFF
clint.c, 674
clint_ipi_send
clint.c, 675
clint.h, 329
CLINT_TIME_CMP_OFF
clint.c, 674
clint_time_rd32
clint.c, 675
clint_time_rd64
clint.c, 676
CLINT_TIME_VAL_OFF
clint.c, 674
clint_time_wr32
clint.c, 676
clint_time_wr64
clint.c, 676
clint_timer
fdt_timer_clint.c, 690
clint_timer_count
fdt_timer_clint.c, 690
clint_timer_event_start
clint.c, 677
clint.h, 329
clint_timer_event_stop
clint.c, 677
clint.h, 329
clint_timer_hartid2data
clint.c, 678
CLINT_TIMER_MAX_NR
fdt_timer_clint.c, 689
clint_timer_value
clint.c, 677
clint.h, 329
clint_warm_ipi_init
clint.c, 677
clint.h, 329
clint_warm_timer_init
clint.c, 677
clint.h, 329
cmpxchg
riscv_atomic.c, 341
cold_init
fdt_ipi, 56
fdt_irqchip, 57
fdt_timer, 61
coldboot_done
sbi_init.c, 411
coldboot_lock
sbi_init.c, 411
coldboot_lottery
sbi_init.c, 411
coldboot_wait_hmask
sbi_init.c, 411
compatible
fdt_match, 57
console_getc
sbi_platform_operations, 69
console_init
sbi_platform_operations, 69
console_out_lock
sbi_console.c, 361
console_plat
sbi_console.c, 361
console_putc
sbi_platform_operations, 69
container_of

sbi_types.h, 294
counter
atomic_t, 53
cpu_relax
riscv_barrier.h, 97
CPU_TO_FDT16
libfdt_env.h, 636
cpu_to_fdt16
libfdt_env.h, 638
CPU_TO_FDT32
libfdt_env.h, 636
cpu_to_fdt32
libfdt_env.h, 638
CPU_TO_FDT64
libfdt_env.h, 636
cpu_to_fdt64
libfdt_env.h, 639
csr_clear
riscv_asm.h, 86
CSR_CYCLE
riscv_encoding.h, 109
CSR_CYCLEH
riscv_encoding.h, 109
CSR_DCSR
riscv_encoding.h, 109
CSR_DPC
riscv_encoding.h, 109
CSR_DSCRATCH
riscv_encoding.h, 109
CSR_FCSR
riscv_encoding.h, 109
CSR_FFLAGS
riscv_encoding.h, 109
CSR_FRM
riscv_encoding.h, 109
CSR_HCOUNTERNEN
riscv_encoding.h, 109
CSR_HEDELEG
riscv_encoding.h, 109
CSR_HGATP
riscv_encoding.h, 109
CSR_HGEIE
riscv_encoding.h, 110
CSR_HGEIP
riscv_encoding.h, 110
CSR_HIDELEG
riscv_encoding.h, 110
CSR_HIE
riscv_encoding.h, 110
CSR_HIP
riscv_encoding.h, 110
CSR_HPMCOUNTER10
riscv_encoding.h, 110
CSR_HPMCOUNTER10H
riscv_encoding.h, 110
CSR_HPMCOUNTER11
riscv_encoding.h, 110
CSR_HPMCOUNTER11H
riscv_encoding.h, 110
CSR_HPMCOUNTER12
riscv_encoding.h, 110
CSR_HPMCOUNTER12H
riscv_encoding.h, 110
CSR_HPMCOUNTER13
riscv_encoding.h, 110
CSR_HPMCOUNTER13H
riscv_encoding.h, 111
CSR_HPMCOUNTER14
riscv_encoding.h, 111
CSR_HPMCOUNTER14H
riscv_encoding.h, 111
CSR_HPMCOUNTER15
riscv_encoding.h, 111
CSR_HPMCOUNTER15H
riscv_encoding.h, 111
CSR_HPMCOUNTER16
riscv_encoding.h, 111
CSR_HPMCOUNTER16H
riscv_encoding.h, 111
CSR_HPMCOUNTER17
riscv_encoding.h, 111
CSR_HPMCOUNTER17H
riscv_encoding.h, 111
CSR_HPMCOUNTER18
riscv_encoding.h, 111
CSR_HPMCOUNTER18H
riscv_encoding.h, 111
CSR_HPMCOUNTER19
riscv_encoding.h, 111
CSR_HPMCOUNTER19H
riscv_encoding.h, 112
CSR_HPMCOUNTER20
riscv_encoding.h, 112
CSR_HPMCOUNTER20H
riscv_encoding.h, 112
CSR_HPMCOUNTER21
riscv_encoding.h, 112
CSR_HPMCOUNTER21H
riscv_encoding.h, 112
CSR_HPMCOUNTER22
riscv_encoding.h, 112
CSR_HPMCOUNTER22H
riscv_encoding.h, 112
CSR_HPMCOUNTER23
riscv_encoding.h, 112
CSR_HPMCOUNTER23H
riscv_encoding.h, 112
CSR_HPMCOUNTER24
riscv_encoding.h, 112
CSR_HPMCOUNTER24H
riscv_encoding.h, 112
CSR_HPMCOUNTER25
riscv_encoding.h, 112
CSR_HPMCOUNTER25H
riscv_encoding.h, 113
CSR_HPMCOUNTER26

riscv_encoding.h, 113
CSR_HPMCOUNTER26H
riscv_encoding.h, 113
CSR_HPMCOUNTER27
riscv_encoding.h, 113
CSR_HPMCOUNTER27H
riscv_encoding.h, 113
CSR_HPMCOUNTER28
riscv_encoding.h, 113
CSR_HPMCOUNTER28H
riscv_encoding.h, 113
CSR_HPMCOUNTER29
riscv_encoding.h, 113
CSR_HPMCOUNTER29H
riscv_encoding.h, 113
CSR_HPMCOUNTER3
riscv_encoding.h, 113
CSR_HPMCOUNTER30
riscv_encoding.h, 113
CSR_HPMCOUNTER30H
riscv_encoding.h, 113
CSR_HPMCOUNTER31
riscv_encoding.h, 114
CSR_HPMCOUNTER31H
riscv_encoding.h, 114
CSR_HPMCOUNTER3H
riscv_encoding.h, 114
CSR_HPMCOUNTER4
riscv_encoding.h, 114
CSR_HPMCOUNTER4H
riscv_encoding.h, 114
CSR_HPMCOUNTER5
riscv_encoding.h, 114
CSR_HPMCOUNTER5H
riscv_encoding.h, 114
CSR_HPMCOUNTER6
riscv_encoding.h, 114
CSR_HPMCOUNTER6H
riscv_encoding.h, 114
CSR_HPMCOUNTER7
riscv_encoding.h, 114
CSR_HPMCOUNTER7H
riscv_encoding.h, 114
CSR_HPMCOUNTER8
riscv_encoding.h, 114
CSR_HPMCOUNTER8H
riscv_encoding.h, 115
CSR_HPMCOUNTER9
riscv_encoding.h, 115
CSR_HPMCOUNTER9H
riscv_encoding.h, 115
CSR_HSTATUS
riscv_encoding.h, 115
CSR_HTIMEDELTA
riscv_encoding.h, 115
CSR_HTIMEDELTAH
riscv_encoding.h, 115
CSR_HTINST
riscv_encoding.h, 115
CSR_HTVAL
riscv_encoding.h, 115
CSR_INSTRET
riscv_encoding.h, 115
CSR_INSTRETH
riscv_encoding.h, 115
CSR_MARCHID
riscv_encoding.h, 115
CSR_MCAUSE
riscv_encoding.h, 115
CSR_MCOUNTEREN
riscv_encoding.h, 116
CSR_MCYCLE
riscv_encoding.h, 116
CSR_MCYCLEH
riscv_encoding.h, 116
CSR_MEDELEG
riscv_encoding.h, 116
CSR_MEPC
riscv_encoding.h, 116
CSR_MHARTID
riscv_encoding.h, 116
CSR_MHPMCOUNTER10
riscv_encoding.h, 116
CSR_MHPMCOUNTER10H
riscv_encoding.h, 116
CSR_MHPMCOUNTER11
riscv_encoding.h, 116
CSR_MHPMCOUNTER11H
riscv_encoding.h, 116
CSR_MHPMCOUNTER12
riscv_encoding.h, 116
CSR_MHPMCOUNTER12H
riscv_encoding.h, 116
CSR_MHPMCOUNTER13
riscv_encoding.h, 117
CSR_MHPMCOUNTER13H
riscv_encoding.h, 117
CSR_MHPMCOUNTER14
riscv_encoding.h, 117
CSR_MHPMCOUNTER14H
riscv_encoding.h, 117
CSR_MHPMCOUNTER15
riscv_encoding.h, 117
CSR_MHPMCOUNTER15H
riscv_encoding.h, 117
CSR_MHPMCOUNTER16
riscv_encoding.h, 117
CSR_MHPMCOUNTER16H
riscv_encoding.h, 117
CSR_MHPMCOUNTER17
riscv_encoding.h, 117
CSR_MHPMCOUNTER17H
riscv_encoding.h, 117
CSR_MHPMCOUNTER18
riscv_encoding.h, 117
CSR_MHPMCOUNTER18H

riscv_encoding.h, 117
CSR_MHPMCOUNTER19
riscv_encoding.h, 118
CSR_MHPMCOUNTER19H
riscv_encoding.h, 118
CSR_MHPMCOUNTER20
riscv_encoding.h, 118
CSR_MHPMCOUNTER20H
riscv_encoding.h, 118
CSR_MHPMCOUNTER21
riscv_encoding.h, 118
CSR_MHPMCOUNTER21H
riscv_encoding.h, 118
CSR_MHPMCOUNTER22
riscv_encoding.h, 118
CSR_MHPMCOUNTER22H
riscv_encoding.h, 118
CSR_MHPMCOUNTER23
riscv_encoding.h, 118
CSR_MHPMCOUNTER23H
riscv_encoding.h, 118
CSR_MHPMCOUNTER24
riscv_encoding.h, 118
CSR_MHPMCOUNTER24H
riscv_encoding.h, 118
CSR_MHPMCOUNTER25
riscv_encoding.h, 119
CSR_MHPMCOUNTER25H
riscv_encoding.h, 119
CSR_MHPMCOUNTER26
riscv_encoding.h, 119
CSR_MHPMCOUNTER26H
riscv_encoding.h, 119
CSR_MHPMCOUNTER27
riscv_encoding.h, 119
CSR_MHPMCOUNTER27H
riscv_encoding.h, 119
CSR_MHPMCOUNTER28
riscv_encoding.h, 119
CSR_MHPMCOUNTER28H
riscv_encoding.h, 119
CSR_MHPMCOUNTER29
riscv_encoding.h, 119
CSR_MHPMCOUNTER29H
riscv_encoding.h, 119
CSR_MHPMCOUNTER3
riscv_encoding.h, 119
CSR_MHPMCOUNTER30
riscv_encoding.h, 119
CSR_MHPMCOUNTER30H
riscv_encoding.h, 120
CSR_MHPMCOUNTER31
riscv_encoding.h, 120
CSR_MHPMCOUNTER31H
riscv_encoding.h, 120
CSR_MHPMCOUNTER3H
riscv_encoding.h, 120
CSR_MHPMCOUNTER4
riscv_encoding.h, 120
CSR_MHPMCOUNTER4H
riscv_encoding.h, 120
CSR_MHPMCOUNTER5
riscv_encoding.h, 120
CSR_MHPMCOUNTER5H
riscv_encoding.h, 120
CSR_MHPMCOUNTER6
riscv_encoding.h, 120
CSR_MHPMCOUNTER6H
riscv_encoding.h, 120
CSR_MHPMCOUNTER7
riscv_encoding.h, 120
CSR_MHPMCOUNTER7H
riscv_encoding.h, 120
CSR_MHPMCOUNTER8
riscv_encoding.h, 121
CSR_MHPMCOUNTER8H
riscv_encoding.h, 121
CSR_MHPMCOUNTER9
riscv_encoding.h, 121
CSR_MHPMCOUNTER9H
riscv_encoding.h, 121
CSR_MHPMEVENT10
riscv_encoding.h, 121
CSR_MHPMEVENT11
riscv_encoding.h, 121
CSR_MHPMEVENT12
riscv_encoding.h, 121
CSR_MHPMEVENT13
riscv_encoding.h, 121
CSR_MHPMEVENT14
riscv_encoding.h, 121
CSR_MHPMEVENT15
riscv_encoding.h, 121
CSR_MHPMEVENT16
riscv_encoding.h, 121
CSR_MHPMEVENT17
riscv_encoding.h, 121
CSR_MHPMEVENT18
riscv_encoding.h, 122
CSR_MHPMEVENT19
riscv_encoding.h, 122
CSR_MHPMEVENT20
riscv_encoding.h, 122
CSR_MHPMEVENT21
riscv_encoding.h, 122
CSR_MHPMEVENT22
riscv_encoding.h, 122
CSR_MHPMEVENT23
riscv_encoding.h, 122
CSR_MHPMEVENT24
riscv_encoding.h, 122
CSR_MHPMEVENT25
riscv_encoding.h, 122
CSR_MHPMEVENT26
riscv_encoding.h, 122
CSR_MHPMEVENT27

riscv_encoding.h, 122
CSR_MHPMEVENT28
riscv_encoding.h, 122
CSR_MHPMEVENT29
riscv_encoding.h, 122
CSR_MHPMEVENT3
riscv_encoding.h, 123
CSR_MHPMEVENT30
riscv_encoding.h, 123
CSR_MHPMEVENT31
riscv_encoding.h, 123
CSR_MHPMEVENT4
riscv_encoding.h, 123
CSR_MHPMEVENT5
riscv_encoding.h, 123
CSR_MHPMEVENT6
riscv_encoding.h, 123
CSR_MHPMEVENT7
riscv_encoding.h, 123
CSR_MHPMEVENT8
riscv_encoding.h, 123
CSR_MHPMEVENT9
riscv_encoding.h, 123
CSR_MIDELEG
riscv_encoding.h, 123
CSR_MIE
riscv_encoding.h, 123
CSR_MIMPID
riscv_encoding.h, 123
CSR_MINSTRET
riscv_encoding.h, 124
CSR_MINSTRETH
riscv_encoding.h, 124
CSR_MIP
riscv_encoding.h, 124
CSR_MISA
riscv_encoding.h, 124
CSR_MSCRATCH
riscv_encoding.h, 124
CSR_MSTATUS
riscv_encoding.h, 124
CSR_MSTATUSH
riscv_encoding.h, 124
CSR_MTINST
riscv_encoding.h, 124
CSR_MTVAL
riscv_encoding.h, 124
CSR_MTVAL2
riscv_encoding.h, 124
CSR_MTVEC
riscv_encoding.h, 124
CSR_MVENDORID
riscv_encoding.h, 124
CSR_PMPADDR0
riscv_encoding.h, 125
CSR_PMPADDR1
riscv_encoding.h, 125
CSR_PMPADDR10
riscv_encoding.h, 125
CSR_PMPADDR11
riscv_encoding.h, 125
CSR_PMPADDR12
riscv_encoding.h, 125
CSR_PMPADDR13
riscv_encoding.h, 125
CSR_PMPADDR14
riscv_encoding.h, 125
CSR_PMPADDR15
riscv_encoding.h, 125
CSR_PMPADDR2
riscv_encoding.h, 125
CSR_PMPADDR3
riscv_encoding.h, 125
CSR_PMPADDR4
riscv_encoding.h, 125
CSR_PMPADDR5
riscv_encoding.h, 125
CSR_PMPADDR6
riscv_encoding.h, 126
CSR_PMPADDR7
riscv_encoding.h, 126
CSR_PMPADDR8
riscv_encoding.h, 126
CSR_PMPADDR9
riscv_encoding.h, 126
CSR_PMPCFG0
riscv_encoding.h, 126
CSR_PMPCFG1
riscv_encoding.h, 126
CSR_PMPCFG2
riscv_encoding.h, 126
CSR_PMPCFG3
riscv_encoding.h, 126
csr_read
riscv_asm.h, 86
csr_read_allowed
sbi_csr_detect.h, 178
csr_read_clear
riscv_asm.h, 86
csr_read_num
riscv_asm.c, 335
riscv_asm.h, 88
csr_read_set
riscv_asm.h, 86
CSR_SATP
riscv_encoding.h, 126
CSR_SCAUSE
riscv_encoding.h, 126
CSR_SCOUNTEREN
riscv_encoding.h, 126
CSR_SEPC
riscv_encoding.h, 126
csr_set
riscv_asm.h, 87
CSR_SIE
riscv_encoding.h, 127

CSR_SIP
riscv_encoding.h, 127
CSR_SSCRATCH
riscv_encoding.h, 127
CSR_SSTATUS
riscv_encoding.h, 127
CSR_STVAL
riscv_encoding.h, 127
CSR_STVEC
riscv_encoding.h, 127
csr_swap
riscv_asm.h, 87
CSR_TDATA1
riscv_encoding.h, 127
CSR_TDATA2
riscv_encoding.h, 127
CSR_TDATA3
riscv_encoding.h, 127
CSR_TIME
riscv_encoding.h, 127
CSR_TIMEH
riscv_encoding.h, 127
CSR_TSELECT
riscv_encoding.h, 127
CSR_UCAUSE
riscv_encoding.h, 128
CSR_UEPC
riscv_encoding.h, 128
CSR_UIE
riscv_encoding.h, 128
CSR_UIP
riscv_encoding.h, 128
CSR_USCRATCH
riscv_encoding.h, 128
CSR_USTATUS
riscv_encoding.h, 128
CSR_UTVAL
riscv_encoding.h, 128
CSR_UTVEC
riscv_encoding.h, 128
CSR_VSATP
riscv_encoding.h, 128
CSR_VSCAUSE
riscv_encoding.h, 128
CSR_VSEPC
riscv_encoding.h, 128
CSR_VSIE
riscv_encoding.h, 128
CSR_VSIP
riscv_encoding.h, 129
CSR_VSSCRATCH
riscv_encoding.h, 129
CSR_VSSTATUS
riscv_encoding.h, 129
CSR_VSTVAL
riscv_encoding.h, 129
CSR_VSTVEC
riscv_encoding.h, 129
csr_write
riscv_asm.h, 87
csr_write_allowed
sbi_csr_detect.h, 178
csr_write_num
riscv_asm.c, 335
riscv_asm.h, 89
ctz
riscv_asm.c, 336
current_driver
fdt_ipi.c, 468
fdt_irqchip.c, 472
fdt_reset.c, 649
fdt_serial.c, 653
fdt_timer.c, 688
current_hartid
riscv_asm.h, 87

data
fdt_match, 57
fdt_property, 58
DECLARE_BITMAP
sbi_bitmap.h, 159
sbi_hartmask, 66
DECLARE_UNPRIVILEGED_LOAD_FUNCTION
sbi_unpriv.h, 298
DECLARE_UNPRIVILEGED_STORE_FUNCTION
sbi_unpriv.h, 298
DEFAULT_SHAKTI_UART_BAUD
fdt_helper.c, 457
DEFAULT_SHAKTI_UART_FREQ
fdt_helper.c, 457
DEFAULT_SIFIVE_UART_BAUD
fdt_helper.c, 457
DEFAULT_SIFIVE_UART_FREQ
fdt_helper.c, 458
DEFAULT_SIFIVE_UART_REG_IO_WIDTH
fdt_helper.c, 458
DEFAULT_SIFIVE_UART_REG_SHIFT
fdt_helper.c, 458
DEFAULT_UART_BAUD
fdt_helper.c, 458
DEFAULT_UART_FREQ
fdt_helper.c, 458
DEFAULT_UART_REG_IO_WIDTH
fdt_helper.c, 458
DEFAULT_UART_REG_SHIFT
fdt_helper.c, 458
DEFINE_BITMAP
sbi_bitmap.h, 160
DEFINE_UNPRIVILEGED_LOAD_FUNCTION
sbi_unpriv.c, 450
DEFINE_UNPRIVILEGED_STORE_FUNCTION
sbi_unpriv.c, 451
delegate_traps
sbi_hart.c, 385
docs/contributing.md, 82
docs/firmware/fw.md, 82
docs/firmware/fw_dynamic.md, 82

docs/firmware/fw_jump.md, 82
docs/firmware/fw_payload.md, 82
docs/firmware/payload_linux.md, 82
docs/firmware/payload_uboot.md, 82
docs/library_usage.md, 82
docs/platform/andes-ae350.md, 82
docs/platform/fpga-ariane.md, 82
docs/platform/fpga-openpiton.md, 82
docs/platform/generic.md, 82
docs/platform/nuclei_ux600.md, 82
docs/platform/platform.md, 82
docs/platform/qemu_virt.md, 82
docs/platform/shakti_cclass.md, 82
docs/platform/sifive_fu540.md, 82
docs/platform/spike.md, 82
docs/platform/thead-c910.md, 82
docs/platform_guide.md, 82
docs/platform_requirements.md, 82
dummy
    fdt_ipi.c, 468
    fdt_serial.c, 653
    fdt_timer.c, 688
dummy_clear
    fdt_ipi.c, 466
dummy_event_start
    fdt_timer.c, 686
dummy_event_stop
    fdt_timer.c, 686
dummy_getc
    fdt_serial.c, 653
dummy_putc
    fdt_serial.c, 653
dummy_send
    fdt_ipi.c, 466
dummy_value
    fdt_timer.c, 687

early_exit
    sbi_platform_operations, 69
early_init
    sbi_platform_operations, 69
ecall_base
    sbi_ecall.h, 183
    sbi_ecall_base.c, 367
ecall_hsm
    sbi_ecall.h, 183
    sbi_ecall_hsm.c, 368
ecall_impid
    sbi_ecall.c, 365
ecall_ipi
    sbi_ecall.h, 183
    sbi_ecall_replace.c, 372
ecall_legacy
    sbi_ecall.h, 183
    sbi_ecall_legacy.c, 370
ecall_rfence
    sbi_ecall.h, 183
    sbi_ecall_replace.c, 373
ecall_time
    sbi_ecall.h, 183
    sbi_ecall_replace.c, 373
ecall_vendor
    sbi_ecall.h, 183
    sbi_ecall_vendor.c, 374
entry_size
    sbi_fifo, 65
epc
    sbi_trap_info, 74
event_start
    fdt_timer, 61
event_stop
    fdt_timer, 61
exit
    fdt_ipi, 56
    fdt_irqchip, 57
    fdt_timer, 61
extid_end
    sbi_ecall_extension, 64
extid_start
    sbi_ecall_extension, 64
extra_lock
    sbi_scratch.c, 425
extra_offset
    sbi_scratch.c, 425
EXTRACT_BYTE
    libfdt_env.h, 636
EXTRACT_FIELD
    sbi_bitops.h, 166

FALSE
    sbi_types.h, 294
false
    sbi_types.h, 294
fdt.c
    check_block_, 480
    check_off_, 481
    fdt_check_header, 481
    fdt_check_node_offset_, 482
    fdt_check_prop_offset_, 483
    fdt_find_string_, 484
    fdt_first_subnode, 484
    fdt_header_size_, 485
    fdt_move, 485
    fdt_next_node, 485
    fdt_next_subnode, 486
    fdt_next_tag, 487
    fdt_offset_ptr, 488
    fdt_ro_probe_, 488
fdt.h
    FDT_BEGIN_NODE, 490
    FDT_END, 490
    FDT_END_NODE, 490
    FDT_MAGIC, 490
    FDT_NOP, 490
    FDT_PROP, 490
    FDT_TAGSIZE, 490
    FDT_V16_SIZE, 490
    FDT_V17_SIZE, 490

FDT_V1_SIZE, 490
FDT_V2_SIZE, 490
FDT_V3_SIZE, 490
fdt16_t
libfdt_env.h, 638
fdt16_to_cpu
libfdt_env.h, 639
fdt32_ld
libfdt.h, 579
fdt32_st
libfdt.h, 579
fdt32_t
libfdt_env.h, 638
fdt32_to_cpu
libfdt_env.h, 639
fdt64_ld
libfdt.h, 580
fdt64_st
libfdt.h, 580
fdt64_t
libfdt_env.h, 638
fdt64_to_cpu
libfdt_env.h, 640
fdt_add_mem_rsv
fdt_rw.c, 540
libfdt.h, 580
fdt_add_property_
fdt_rw.c, 540
fdt_add_reservemap_entry
fdt_sw.c, 556
libfdt.h, 581
fdt_add_string_
fdt_sw.c, 557
fdt_add_subnode
fdt_rw.c, 541
libfdt.h, 581
fdt_add_subnode_namelen
fdt_rw.c, 542
libfdt.h, 582
fdt_address_cells
fdt_addresses.c, 491
libfdt.h, 583
fdt_addresses.c
fdt_address_cells, 491
fdt_appendprop_addrrange, 492
fdt_cells, 493
fdt_size_cells, 493
FDT_ALIGN
libfdt_internal.h, 641
fdt_appendprop
fdt_rw.c, 542
libfdt.h, 583
fdt_appendprop_addrrange
fdt_addresses.c, 492
libfdt.h, 584
fdt_appendprop_cell
libfdt.h, 584
fdt_appendprop_string
libfdt.h, 574
fdt_appendprop_u32
libfdt.h, 585
fdt_appendprop_u64
libfdt.h, 585
FDT_BEGIN_NODE
fdt.h, 490
fdt_begin_node
fdt_sw.c, 557
libfdt.h, 586
FDT_BITWISE
libfdt_env.h, 636
fdt_blocks_misordered_
fdt_rw.c, 543
fdt_boot_cpuid_phys
libfdt.h, 574
fdt_cells
fdt_addresses.c, 493
fdt_check_full
fdt_ro.c, 508
libfdt.h, 586
fdt_check_header
fdt.c, 481
libfdt.h, 587
fdt_check_node_offset_
fdt.c, 482
libfdt_internal.h, 642
fdt_check_prop_offset_
fdt.c, 483
libfdt_internal.h, 643
fdt_cpu_fixup
fdt_fixup.c, 453
fdt_fixup.h, 300
fdt_create
fdt_sw.c, 557
libfdt.h, 587
fdt_create_empty_tree
fdt_empty_tree.c, 495
libfdt.h, 588
FDT_CREATE_FLAG_NO_NAME_DEDUP
libfdt.h, 574
FDT_CREATE_FLAGS_ALL
libfdt.h, 574
fdt_create_with_flags
fdt_sw.c, 558
libfdt.h, 589
fdt_data_size_
fdt_rw.c, 543
fdt_del_last_string_
fdt_rw.c, 544
fdt_sw.c, 558
fdt_del_mem_rsv
fdt_rw.c, 544
libfdt.h, 589
fdt_del_node
fdt_rw.c, 545
libfdt.h, 590
fdt_delprop

fdt_rw.c, 545
libfdt.h, 590
fdt_empty_tree.c
fdt_create_empty_tree, 495
FDT_END
fdt.h, 490
FDT_END_NODE
fdt.h, 490
fdt_end_node
fdt_sw.c, 559
libfdt.h, 590
FDT_ERR_BADFLAGS
libfdt.h, 574
FDT_ERR_BADLAYOUT
libfdt.h, 575
FDT_ERR_BADMAGIC
libfdt.h, 575
FDT_ERR_BADNCELLS
libfdt.h, 575
FDT_ERR_BADOFFSET
libfdt.h, 575
FDT_ERR_BADOVERLAY
libfdt.h, 575
FDT_ERR_BADPATH
libfdt.h, 575
FDT_ERR_BADPHANDLE
libfdt.h, 575
FDT_ERR_BADSTATE
libfdt.h, 575
FDT_ERR_BADSTRUCTURE
libfdt.h, 575
FDT_ERR_BADVALUE
libfdt.h, 575
FDT_ERR_BADVERSION
libfdt.h, 575
FDT_ERR_EXISTS
libfdt.h, 575
FDT_ERR_INTERNAL
libfdt.h, 576
FDT_ERR_MAX
libfdt.h, 576
FDT_ERR_NOPHANDLES
libfdt.h, 576
FDT_ERR_NOSPACE
libfdt.h, 576
FDT_ERR_NOTFOUND
libfdt.h, 576
FDT_ERR_TRUNCATED
libfdt.h, 576
FDT_ERRTABENT
fdt_strerror.c, 553
fdt_errtable
fdt_strerror.c, 554
FDT_ERRTABSIZE
fdt_strerror.c, 553
fdt_find_add_string_
fdt_rw.c, 545
fdt_sw.c, 559
fdt_find_match
fdt_helper.c, 458
fdt_helper.h, 305
fdt_find_max_phandle
fdt_ro.c, 509
libfdt.h, 591
fdt_find_string_
fdt.c, 484
libfdt_internal.h, 644
fdt_finish
fdt_sw.c, 560
libfdt.h, 592
fdt_finish_reservemap
fdt_sw.c, 561
libfdt.h, 592
fdt_first_property_offset
fdt_ro.c, 509
libfdt.h, 593
fdt_first_subnode
fdt.c, 484
libfdt.h, 594
FDT_FIRST_SUPPORTED_VERSION
libfdt.h, 576
fdt_fixup.c
fdt_cpu_fixup, 453
fdt_fixups, 453
fdt_plic_fixup, 454
fdt_reserved_memory_fixup, 454
fdt_reserved_memory_nomap_fixup, 455
fdt_resv_memory_update_node, 456
fdt_fixup.h
fdt_cpu_fixup, 300
fdt_fixups, 301
fdt_plic_fixup, 301
fdt_reserved_memory_fixup, 302
fdt_reserved_memory_nomap_fixup, 303
fdt_fixups
fdt_fixup.c, 453
fdt_fixup.h, 301
fdt_for_each_property_offset
libfdt.h, 576
fdt_for_each_subnode
libfdt.h, 576
FDT_FORCE
libfdt_env.h, 636
fdt_generate_phandle
fdt_ro.c, 510
libfdt.h, 594
fdt_get_alias
fdt_ro.c, 511
libfdt.h, 595
fdt_get_alias_namelen
fdt_ro.c, 511
libfdt.h, 595
fdt_get_header
libfdt.h, 577
fdt_get_max_phandle
libfdt.h, 596

fdt_get_mem_rsv
fdt_ro.c, 512
libfdt.h, 596
fdt_get_name
fdt_ro.c, 512
libfdt.h, 597
fdt_get_node_addr_size
fdt_helper.c, 459
fdt_helper.h, 305
fdt_get_path
fdt_ro.c, 513
libfdt.h, 598
fdt_get_phandle
fdt_ro.c, 514
libfdt.h, 599
fdt_get_property
fdt_ro.c, 515
libfdt.h, 599
fdt_get_property_by_offset
fdt_ro.c, 515
libfdt.h, 600
fdt_get_property_by_offset_
fdt_ro.c, 516
fdt_get_property_namelen
fdt_ro.c, 517
libfdt.h, 600
fdt_get_property_namelen_
fdt_ro.c, 518
fdt_get_property_w
libfdt.h, 601
fdt_get_string
fdt_ro.c, 518
libfdt.h, 601
fdt_getprop
fdt_ro.c, 519
libfdt.h, 602
fdt_getprop_by_offset
fdt_ro.c, 520
libfdt.h, 603
fdt_getprop_namelen
fdt_ro.c, 521
libfdt.h, 604
fdt_getprop_namelen_w
libfdt.h, 605
fdt_getprop_w
libfdt.h, 605
fdt_grab_space_
fdt_sw.c, 561
fdt_header, 55
boot_cpuid_phys, 55
last_comp_version, 55
magic, 55
off_dt_strings, 55
off_dt_struct, 55
off_mem_rsvmap, 55
size_dt_strings, 55
size_dt_struct, 55
totalsize, 55
version, 56
fdt_header_size
libfdt.h, 606
fdt_header_size_
fdt.c, 485
libfdt.h, 606
fdt_helper.c
DEFAULT_SHAKTI_UART_BAUD, 457
DEFAULT_SHAKTI_UART_FREQ, 457
DEFAULT_SIFIVE_UART_BAUD, 457
DEFAULT_SIFIVE_UART_FREQ, 458
DEFAULT_SIFIVE_UART_REG_IO_WIDTH, 458
DEFAULT_SIFIVE_UART_REG_SHIFT, 458
DEFAULT_UART_BAUD, 458
DEFAULT_UART_FREQ, 458
DEFAULT_UART_REG_IO_WIDTH, 458
DEFAULT_UART_REG_SHIFT, 458
fdt_find_match, 458
fdt_get_node_addr_size, 459
fdt_match_node, 460
fdt_parse_clint_node, 460
fdt_parse_compat_addr, 461
fdt_parse_hart_id, 461
fdt_parse_max_hart_id, 462
fdt_parse_plic, 462
fdt_parse_plic_node, 463
fdt_parse_shakti_uart_node, 463
fdt_parse_sifive_uart_node, 464
fdt_parse_uart8250, 464
fdt_parse_uart8250_node, 465
fdt_helper.h
fdt_find_match, 305
fdt_get_node_addr_size, 305
fdt_match_node, 306
fdt_parse_clint_node, 307
fdt_parse_compat_addr, 307
fdt_parse_hart_id, 308
fdt_parse_max_hart_id, 308
fdt_parse_plic, 309
fdt_parse_plic_node, 309
fdt_parse_shakti_uart_node, 310
fdt_parse_sifive_uart_node, 310
fdt_parse_uart8250, 311
fdt_parse_uart8250_node, 311
fdt_ipi, 56
clear, 56
cold_init, 56
exit, 56
match_table, 56
send, 56
warm_init, 56
fdt_ipi.c
current_driver, 468
dummy, 468
dummy_clear, 466
dummy_send, 466
fdt_ipi_clear, 466
fdt_ipi_clint, 468

fdt_ipi_cold_init, 467
fdt_ipi_exit, 467
fdt_ipi_init, 467
fdt_ipi_send, 467
fdt_ipi_warm_init, 467
ipi_drivers, 468
fdt_ipi.h
fdt_ipi_clear, 313
fdt_ipi_exit, 313
fdt_ipi_init, 313
fdt_ipi_send, 314
fdt_ipi_clear
fdt_ipi.c, 466
fdt_ipi.h, 313
fdt_ipi_clint
fdt_ipi.c, 468
fdt_ipi_clint.c, 470
fdt_ipi_clint.c
clint_ipi, 470
clint_ipi_count, 470
CLINT_IPI_MAX_NR, 469
fdt_ipi_clint, 470
ipi_clint_cold_init, 469
ipi_clint_match, 470
fdt_ipi_cold_init
fdt_ipi.c, 467
fdt_ipi_exit
fdt_ipi.c, 467
fdt_ipi.h, 313
fdt_ipi_init
fdt_ipi.c, 467
fdt_ipi.h, 313
fdt_ipi_send
fdt_ipi.c, 467
fdt_ipi.h, 314
fdt_ipi_warm_init
fdt_ipi.c, 467
fdt_irqchip, 57
cold_init, 57
exit, 57
match_table, 57
warm_init, 57
fdt_irqchip.c
current_driver, 472
fdt_irqchip_cold_init, 471
fdt_irqchip_exit, 472
fdt_irqchip_init, 472
fdt_irqchip_plic, 472
fdt_irqchip_warm_init, 472
irqchip_drivers, 473
fdt_irqchip.h
fdt_irqchip_exit, 315
fdt_irqchip_init, 315
fdt_irqchip_cold_init
fdt_irqchip.c, 471
fdt_irqchip_exit
fdt_irqchip.c, 472
fdt_irqchip.h, 315
fdt_irqchip_init
fdt_irqchip.c, 472
fdt_irqchip.h, 315
fdt_irqchip_plic
fdt_irqchip.c, 472
fdt_irqchip_plic.c, 475
fdt_irqchip_plic.c
fdt_irqchip_plic, 475
irqchip_plic_cold_init, 474
irqchip_plic_match, 475
irqchip_plic_update_hartid_table, 474
irqchip_plic_warm_init, 475
plic, 475
plic_count, 475
plic_hartid2context, 476
plic_hartid2data, 476
PLIC_MAX_NR, 474
fdt_irqchip_warm_init
fdt_irqchip.c, 472
fdt_last_comp_version
libfdt.h, 577
FDT_LAST_SUPPORTED_VERSION
libfdt.h, 577
FDT_MAGIC
fdt.h, 490
fdt_magic
libfdt.h, 577
fdt_match, 57
compatible, 57
data, 57
fdt_match_node
fdt_helper.c, 460
fdt_helper.h, 306
FDT_MAX_NCELLS
libfdt.h, 577
FDT_MAX_PHANDLE
libfdt.h, 577
fdt_mem_rsv
fdt_ro.c, 522
fdt_mem_rsv_
libfdt_internal.h, 644
fdt_mem_rsv_w_
libfdt_internal.h, 645
fdt_move
fdt.c, 485
libfdt.h, 607
fdt_next_node
fdt.c, 485
libfdt.h, 607
fdt_next_property_offset
fdt_ro.c, 523
libfdt.h, 608
fdt_next_subnode
fdt.c, 486
libfdt.h, 609
fdt_next_tag
fdt.c, 487
libfdt.h, 609

fdt_node_check_compatible
fdt_ro.c, 524
libfdt.h, 610
fdt_node_depth
fdt_ro.c, 525
libfdt.h, 611
fdt_node_end_offset_
fdt_wip.c, 565
libfdt_internal.h, 645
fdt_node_header, 58
name, 58
tag, 58
fdt_node_offset_by_compatible
fdt_ro.c, 525
libfdt.h, 612
fdt_node_offset_by_phandle
fdt_ro.c, 526
libfdt.h, 613
fdt_node_offset_by_prop_value
fdt_ro.c, 527
libfdt.h, 614
fdt_nodename_eq_
fdt_ro.c, 528
FDT_NOP
fdt.h, 490
fdt_nop_node
fdt_wip.c, 566
libfdt.h, 614
fdt_nop_property
fdt_wip.c, 566
libfdt.h, 615
fdt_nop_region_
fdt_wip.c, 566
fdt_num_mem_rsv
fdt_ro.c, 528
libfdt.h, 615
fdt_off_dt_strings
libfdt.h, 577
fdt_off_dt_struct
libfdt.h, 577
fdt_off_mem_rsvmap
libfdt.h, 578
fdt_offset_ptr
fdt.c, 488
libfdt.h, 615
fdt_offset_ptr_
libfdt_internal.h, 646
fdt_offset_ptr_w
libfdt.h, 616
fdt_offset_ptr_w_
libfdt_internal.h, 646
fdt_open_into
fdt_rw.c, 546
libfdt.h, 617
fdt_overlay.c
fdt_overlay_apply, 497
get_path_len, 498
overlay_adjust_local_phandles, 498
overlay_adjust_node_phandles, 499
overlay_apply_node, 499
overlay_fixup_one_phandle, 500
overlay_fixup_phandle, 501
overlay_fixup_phandles, 501
overlay_get_target, 502
overlay_get_target_phandle, 503
overlay_merge, 503
overlay_phandle_add_offset, 504
overlay_symbol_update, 504
overlay_update_local_node_references, 505
overlay_update_local_references, 506
fdt_overlay_apply
fdt_overlay.c, 497
libfdt.h, 617
fdt_pack
fdt_rw.c, 546
libfdt.h, 618
fdt_packblocks_
fdt_rw.c, 547
fdt_parent_offset
fdt_ro.c, 529
libfdt.h, 618
fdt_parse_clint_node
fdt_helper.c, 460
fdt_helper.h, 307
fdt_parse_compat_addr
fdt_helper.c, 461
fdt_helper.h, 307
fdt_parse_hart_id
fdt_helper.c, 461
fdt_helper.h, 308
fdt_parse_max_hart_id
fdt_helper.c, 462
fdt_helper.h, 308
fdt_parse_plic
fdt_helper.c, 462
fdt_helper.h, 309
fdt_parse_plic_node
fdt_helper.c, 463
fdt_helper.h, 309
fdt_parse_shakti_uart_node
fdt_helper.c, 463
fdt_helper.h, 310
fdt_parse_sifive_uart_node
fdt_helper.c, 464
fdt_helper.h, 310
fdt_parse_uart8250
fdt_helper.c, 464
fdt_helper.h, 311
fdt_parse_uart8250_node
fdt_helper.c, 465
fdt_helper.h, 311
fdt_path_offset
fdt_ro.c, 530
libfdt.h, 619
fdt_path_offset_namelen
fdt_ro.c, 531

libfdt.h, 620
fdt_plic_fixup
fdt_fixup.c, 454
fdt_fixup.h, 301
FDT_PROP
fdt.h, 490
fdt_property, 58
data, 58
fdt_sw.c, 562
len, 58
libfdt.h, 621
nameoff, 58
tag, 59
fdt_property_cell
libfdt.h, 621
fdt_property_placeholder
fdt_sw.c, 562
libfdt.h, 622
fdt_property_string
libfdt.h, 578
fdt_property_u32
libfdt.h, 622
fdt_property_u64
libfdt.h, 623
fdt_reserve_entry, 59
address, 59
size, 59
fdt_reserved_memory_fixup
fdt_fixup.c, 454
fdt_fixup.h, 302
fdt_reserved_memory_nomap_fixup
fdt_fixup.c, 455
fdt_fixup.h, 303
fdt_reset, 59
init, 59
match_table, 59
system_reset, 60
fdt_reset.c
current_driver, 649
fdt_reset_htif, 649
fdt_reset_init, 648
fdt_reset_sifive, 649
fdt_system_reset, 649
reset_drivers, 649
fdt_reset.h
fdt_reset_init, 319
fdt_system_reset, 319
fdt_reset_htif
fdt_reset.c, 649
fdt_reset_htif.c, 650
fdt_reset_htif.c
fdt_reset_htif, 650
htif_reset_match, 650
fdt_reset_init
fdt_reset.c, 648
fdt_reset.h, 319
fdt_reset_sifive
fdt_reset.c, 649
fdt_reset_sifive.c, 651
fdt_reset_sifive.c
fdt_reset_sifive, 651
sifive_test_reset_init, 651
sifive_test_reset_match, 652
fdt_resize
fdt_sw.c, 563
libfdt.h, 623
fdt_resize_property_
fdt_rw.c, 547
fdt_resv_memory_update_node
fdt_fixup.c, 456
fdt_ro.c
fdt_check_full, 508
fdt_find_max_phandle, 509
fdt_first_property_offset, 509
fdt_generate_phandle, 510
fdt_get_alias, 511
fdt_get_alias_namelen, 511
fdt_get_mem_rsv, 512
fdt_get_name, 512
fdt_get_path, 513
fdt_get_phandle, 514
fdt_get_property, 515
fdt_get_property_by_offset, 515
fdt_get_property_by_offset_, 516
fdt_get_property_namelen, 517
fdt_get_property_namelen_, 518
fdt_get_string, 518
fdt_getprop, 519
fdt_getprop_by_offset, 520
fdt_getprop_namelen, 521
fdt_mem_rsv, 522
fdt_next_property_offset, 523
fdt_node_check_compatible, 524
fdt_node_depth, 525
fdt_node_offset_by_compatible, 525
fdt_node_offset_by_phandle, 526
fdt_node_offset_by_prop_value, 527
fdt_nodename_eq_, 528
fdt_num_mem_rsv, 528
fdt_parent_offset, 529
fdt_path_offset, 530
fdt_path_offset_namelen, 531
fdt_string, 531
fdt_string_eq_, 532
fdt_stringlist_contains, 533
fdt_stringlist_count, 533
fdt_stringlist_get, 534
fdt_stringlist_search, 534
fdt_subnode_offset, 534
fdt_subnode_offset_namelen, 535
fdt_supernode_atdepth_offset, 536
nextprop_, 537
FDT_RO_PROBE
libfdt_internal.h, 641
fdt_ro_probe_
fdt.c, 488

libfdt_internal.h, 647
fdt_rw.c
fdt_add_mem_rsv, 540
fdt_add_property_, 540
fdt_add_subnode, 541
fdt_add_subnode_namelen, 542
fdt_appendprop, 542
fdt_blocks_misordered_, 543
fdt_data_size_, 543
fdt_del_last_string_, 544
fdt_del_mem_rsv, 544
fdt_del_node, 545
fdt_delprop, 545
fdt_find_add_string_, 545
fdt_open_into, 546
fdt_pack, 546
fdt_packblocks_, 547
fdt_resize_property_, 547
FDT_RW_PROBE, 540
fdt_rw_probe_, 548
fdt_set_name, 548
fdt_setprop, 548
fdt_setprop_placeholder, 549
fdt_splice_, 549
fdt_splice_mem_rsv_, 550
fdt_splice_string_, 551
fdt_splice_struct_, 551
FDT_RW_PROBE
fdt_rw.c, 540
fdt_rw_probe_
fdt_rw.c, 548
fdt_serial, 60
getc, 60
init, 60
match_table, 60
putc, 60
fdt_serial.c
current_driver, 653
dummy, 653
dummy_getc, 653
dummy_putc, 653
fdt_serial_getc, 653
fdt_serial_htif, 654
fdt_serial_init, 653
fdt_serial_putc, 653
fdt_serial_shakti, 654
fdt_serial_sifive, 654
fdt_serial_uart8250, 654
serial_drivers, 654
fdt_serial.h
fdt_serial_getc, 320
fdt_serial_init, 320
fdt_serial_putc, 320
fdt_serial_getc
fdt_serial.c, 653
fdt_serial.h, 320
fdt_serial_htif
fdt_serial.c, 654
fdt_serial_htif.c, 655
fdt_serial_htif.c
fdt_serial_htif, 655
serial_htif_match, 655
fdt_serial_init
fdt_serial.c, 653
fdt_serial.h, 320
fdt_serial_putc
fdt_serial.c, 653
fdt_serial.h, 320
fdt_serial_shakti
fdt_serial.c, 654
fdt_serial_shakti.c, 656
fdt_serial_shakti.c
fdt_serial_shakti, 656
serial_shakti_init, 656
serial_shakti_match, 657
fdt_serial_sifive
fdt_serial.c, 654
fdt_serial_sifive.c, 658
fdt_serial_sifive.c
fdt_serial_sifive, 658
serial_sifive_init, 657
serial_sifive_match, 658
fdt_serial_uart8250
fdt_serial.c, 654
fdt_serial_uart8250.c, 659
fdt_serial_uart8250.c
fdt_serial_uart8250, 659
serial_uart8250_init, 659
serial_uart8250_match, 659
fdt_set_hdr_
libfdt.h, 578, 624
fdt_set_name
fdt_rw.c, 548
libfdt.h, 625
fdt_setprop
fdt_rw.c, 548
libfdt.h, 625
fdt_setprop_cell
libfdt.h, 625
fdt_setprop_empty
libfdt.h, 578
fdt_setprop_inplace
fdt_wip.c, 567
libfdt.h, 626
fdt_setprop_inplace_cell
libfdt.h, 626
fdt_setprop_inplace_namelen_partial
fdt_wip.c, 568
libfdt.h, 627
fdt_setprop_inplace_u32
libfdt.h, 627
fdt_setprop_inplace_u64
libfdt.h, 628
fdt_setprop_placeholder
fdt_rw.c, 549
libfdt.h, 628

fdt_setprop_string
libfdt.h, 578
fdt_setprop_u32
libfdt.h, 629
fdt_setprop_u64
libfdt.h, 629
fdt_size_cells
fdt_addresses.c, 493
libfdt.h, 629
fdt_size_dt_strings
libfdt.h, 578
fdt_size_dt_struct
libfdt.h, 578
fdt_splice_
fdt_rw.c, 549
fdt_splice_mem_rsv_
fdt_rw.c, 550
fdt_splice_string_
fdt_rw.c, 551
fdt_splice_struct_
fdt_rw.c, 551
fdt_strerror
fdt_strerror.c, 554
libfdt.h, 630
fdt_strerror.c
FDT_ERRTABENT, 553
fdt_errtable, 554
FDT_ERRTABSIZE, 553
fdt_strerror, 554
fdt_string
fdt_ro.c, 531
libfdt.h, 630
fdt_string_eq_
fdt_ro.c, 532
fdt_stringlist_contains
fdt_ro.c, 533
libfdt.h, 631
fdt_stringlist_count
fdt_ro.c, 533
libfdt.h, 631
fdt_stringlist_get
fdt_ro.c, 534
libfdt.h, 631
fdt_stringlist_search
fdt_ro.c, 534
libfdt.h, 632
fdt_subnode_offset
fdt_ro.c, 534
libfdt.h, 632
fdt_subnode_offset_namelen
fdt_ro.c, 535
libfdt.h, 633
fdt_supernode_atdepth_offset
fdt_ro.c, 536
libfdt.h, 633
fdt_sw.c
fdt_add_reservemap_entry, 556
fdt_add_string_, 557
fdt_begin_node, 557
fdt_create, 557
fdt_create_with_flags, 558
fdt_del_last_string_, 558
fdt_end_node, 559
fdt_find_add_string_, 559
fdt_finish, 560
fdt_finish_reservemap, 561
fdt_grab_space_, 561
fdt_property, 562
fdt_property_placeholder, 562
fdt_resize, 563
FDT_SW_PROBE, 556
fdt_sw_probe_, 563
FDT_SW_PROBE_MEMRSV, 556
fdt_sw_probe_memrsv_, 563
FDT_SW_PROBE_STRUCT, 556
fdt_sw_probe_struct_, 564
sw_flags, 564
FDT_SW_MAGIC
libfdt_internal.h, 642
FDT_SW_PROBE
fdt_sw.c, 556
fdt_sw_probe_
fdt_sw.c, 563
FDT_SW_PROBE_MEMRSV
fdt_sw.c, 556
fdt_sw_probe_memrsv_
fdt_sw.c, 563
FDT_SW_PROBE_STRUCT
fdt_sw.c, 556
fdt_sw_probe_struct_
fdt_sw.c, 564
fdt_system_reset
fdt_reset.c, 649
fdt_reset.h, 319
FDT_TAGALIGN
libfdt_internal.h, 642
FDT_TAGSIZE
fdt.h, 490
fdt_timer, 60
cold_init, 61
event_start, 61
event_stop, 61
exit, 61
match_table, 61
value, 61
warm_init, 61
fdt_timer.c
current_driver, 688
dummy, 688
dummy_event_start, 686
dummy_event_stop, 686
dummy_value, 687
fdt_timer_clint, 688
fdt_timer_cold_init, 687
fdt_timer_event_start, 687
fdt_timer_event_stop, 687

fdt_timer_exit, 687
fdt_timer_init, 687
fdt_timer_value, 688
fdt_timer_warm_init, 688
timer_drivers, 689
fdt_timer.h
fdt_timer_event_start, 334
fdt_timer_event_stop, 334
fdt_timer_exit, 334
fdt_timer_init, 334
fdt_timer_value, 334
fdt_timer_clint
fdt_timer.c, 688
fdt_timer_clint.c, 690
fdt_timer_clint.c
clint_timer, 690
clint_timer_count, 690
CLINT_TIMER_MAX_NR, 689
fdt_timer_clint, 690
timer_clint_cold_init, 690
timer_clint_match, 690
fdt_timer_cold_init
fdt_timer.c, 687
fdt_timer_event_start
fdt_timer.c, 687
fdt_timer.h, 334
fdt_timer_event_stop
fdt_timer.c, 687
fdt_timer.h, 334
fdt_timer_exit
fdt_timer.c, 687
fdt_timer.h, 334
fdt_timer_init
fdt_timer.c, 687
fdt_timer.h, 334
fdt_timer_value
fdt_timer.c, 688
fdt_timer.h, 334
fdt_timer_warm_init
fdt_timer.c, 688
fdt_totalsize
libfdt.h, 578
FDT_V16_SIZE
fdt.h, 490
FDT_V17_SIZE
fdt.h, 490
FDT_V1_SIZE
fdt.h, 490
FDT_V2_SIZE
fdt.h, 490
FDT_V3_SIZE
fdt.h, 490
fdt_version
libfdt.h, 579
fdt_wip.c
fdt_node_end_offset_, 565
fdt_nop_node, 566
fdt_nop_property, 566
fdt_nop_region_, 566
fdt_setprop_inplace, 567
fdt_setprop_inplace_namelen_partial, 568
features
sbi_platform, 67
ffs
sbi_bitops.h, 169
ffz
sbi_bitops.h, 166
final_exit
sbi_platform_operations, 69
final_init
sbi_platform_operations, 70
find_first_bit
sbi_bitops.c, 351
sbi_bitops.h, 169
find_first_zero_bit
sbi_bitops.c, 351
sbi_bitops.h, 170
find_last_bit
sbi_bitops.c, 351
sbi_bitops.h, 170
find_next_bit
sbi_bitops.c, 351
sbi_bitops.h, 170
find_next_zero_bit
sbi_bitops.c, 352
sbi_bitops.h, 170
FINISHER_FAIL
sifive_test.c, 684
FINISHER_PASS
sifive_test.c, 684
FINISHER_RESET
sifive_test.c, 684
firmware_context
sbi_platform, 67
first_hartid
clint_data, 54
fls
sbi_bitops.h, 171
for_each_clear_bit
sbi_bitops.h, 166
for_each_clear_bit_from
sbi_bitops.h, 166
for_each_set_bit
sbi_bitops.h, 166
for_each_set_bit_from
sbi_bitops.h, 167
format
sbi_console.h, 175
fp_init
sbi_hart.c, 385
freq
platform_uart_data, 63
FROMHOST_CMD
htif.c, 679
FROMHOST_DATA
htif.c, 679

FROMHOST_DEV
htif.c, 679
fw_dynamic.h
__packed, 84
FW_DYNAMIC_INFO_BOOT_HART_OFFSET, 83
FW_DYNAMIC_INFO_MAGIC_OFFSET, 83
FW_DYNAMIC_INFO_MAGIC_VALUE, 84
FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET, 84
FW_DYNAMIC_INFO_NEXT_MODE_M, 84
FW_DYNAMIC_INFO_NEXT_MODE_OFFSET, 84
FW_DYNAMIC_INFO_NEXT_MODE_S, 84
FW_DYNAMIC_INFO_NEXT_MODE_U, 84
FW_DYNAMIC_INFO_OPTIONS_OFFSET, 84
FW_DYNAMIC_INFO_VERSION_MAX, 84
FW_DYNAMIC_INFO_VERSION_OFFSET, 84
fw_dynamic_info, 61
boot_hart, 62
magic, 62
next_addr, 62
next_mode, 62
options, 62
version, 62
FW_DYNAMIC_INFO_BOOT_HART_OFFSET
fw_dynamic.h, 83
FW_DYNAMIC_INFO_MAGIC_OFFSET
fw_dynamic.h, 83
FW_DYNAMIC_INFO_MAGIC_VALUE
fw_dynamic.h, 84
FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET
fw_dynamic.h, 84
FW_DYNAMIC_INFO_NEXT_MODE_M
fw_dynamic.h, 84
FW_DYNAMIC_INFO_NEXT_MODE_OFFSET
fw_dynamic.h, 84
FW_DYNAMIC_INFO_NEXT_MODE_S
fw_dynamic.h, 84
FW_DYNAMIC_INFO_NEXT_MODE_U
fw_dynamic.h, 84
FW_DYNAMIC_INFO_OPTIONS_OFFSET
fw_dynamic.h, 84
FW_DYNAMIC_INFO_VERSION_MAX
fw_dynamic.h, 84
FW_DYNAMIC_INFO_VERSION_OFFSET
fw_dynamic.h, 84
fw_size
sbi_scratch, 72
fw_start
sbi_scratch, 72

GENMASK
sbi_bitops.h, 167
get_path_len
fdt_overlay.c, 498
GET_PRECISION
riscv_fp.h, 149
get_reg
sifive-uart.c, 664
uart8250.c, 670
GET_RM
riscv_encoding.h, 129
riscv_fp.h, 150
GET_RS1
riscv_encoding.h, 129
GET_RS1S
riscv_encoding.h, 129
GET_RS2
riscv_encoding.h, 129
GET_RS2C
riscv_encoding.h, 129
GET_RS2S
riscv_encoding.h, 130
GET_SP
riscv_encoding.h, 130
get_ticks
sbi_timer.c, 432
get_time_val
sbi_timer.c, 436
get_tlbr_flush_limit
sbi_platform_operations, 70
getc
fdt_serial, 60
gp
sbi_trap_regs, 76

handle
sbi_ecall_extension, 64
hart_count
clint_data, 54
sbi_platform, 67
hart_data_offset
sbi_hsm.c, 400
hart_detect_features
sbi_hart.c, 386
hart_features_offset
sbi_hart.c, 393
hart_get_features
sbi_hart.c, 386
hart_index2id
sbi_platform, 68
hart_stack_size
sbi_platform, 68
hart_start
sbi_platform_operations, 70
hart_stop
sbi_platform_operations, 70
hartid2scratch
sbi_scratch.c, 424
hartid_to_scratch
sbi_scratch, 72
hartid_to_scratch_table
sbi_scratch.c, 425
sbi_scratch.h, 272
has_64bit_mmio
clint_data, 54
head
sbi_ecall_extension, 65
HGATP32_MODE_SHIFT
riscv_encoding.h, 130

HGATP32_PPN
riscv_encoding.h, 130
HGATP32_VMID_MASK
riscv_encoding.h, 130
HGATP32_VMID_SHIFT
riscv_encoding.h, 130
HGATP64_MODE_SHIFT
riscv_encoding.h, 130
HGATP64_PPN
riscv_encoding.h, 130
HGATP64_VMID_MASK
riscv_encoding.h, 130
HGATP64_VMID_SHIFT
riscv_encoding.h, 130
HGATP_MODE_OFF
riscv_encoding.h, 130
HGATP_MODE_SHIFT
riscv_encoding.h, 131
HGATP_MODE_SV32X4
riscv_encoding.h, 131
HGATP_MODE_SV39X4
riscv_encoding.h, 131
HGATP_MODE_SV48X4
riscv_encoding.h, 131
HGATP_PPN
riscv_encoding.h, 131
HGATP_VMID_MASK
riscv_encoding.h, 131
HGATP_VMID_SHIFT
riscv_encoding.h, 131
HSTATUS_GVA
riscv_encoding.h, 131
HSTATUS_HU
riscv_encoding.h, 131
HSTATUS_SPV
riscv_encoding.h, 131
HSTATUS_SPVP
riscv_encoding.h, 131
HSTATUS_VGEIN
riscv_encoding.h, 131
HSTATUS_VGEIN_SHIFT
riscv_encoding.h, 132
HSTATUS_VSBE
riscv_encoding.h, 132
HSTATUS_VTSR
riscv_encoding.h, 132
HSTATUS_VTVM
riscv_encoding.h, 132
HSTATUS_VTW
riscv_encoding.h, 132
htif.c
__attribute__, 681
__check_fromhost, 681
__set_tohost, 681
FROMHOST_CMD, 679
FROMHOST_DATA, 679
FROMHOST_DEV, 679
HTIF_CMD_BITS, 679
HTIF_CMD_MASK, 679
HTIF_CMD_SHIFT, 680
htif_console_buf, 683
HTIF_CONSOLE_CMD_GETC, 680
HTIF_CONSOLE_CMD_PUTC, 680
HTIF_DATA_BITS, 680
HTIF_DATA_MASK, 680
HTIF_DATA_SHIFT, 680
HTIF_DEV_BITS, 680
HTIF_DEV_CONSOLE, 680
HTIF_DEV_MASK, 680
HTIF_DEV_SHIFT, 680
HTIF_DEV_SYSTEM, 680
htif_getc, 682
htif_lock, 683
htif_putc, 682
htif_system_reset, 683
PK_SYS_write, 680
TOHOST_CMD, 681
htif.h
htif_getc, 331
htif_putc, 331
htif_system_reset, 331
HTIF_CMD_BITS
htif.c, 679
HTIF_CMD_MASK
htif.c, 679
HTIF_CMD_SHIFT
htif.c, 680
htif_console_buf
htif.c, 683
HTIF_CONSOLE_CMD_GETC
htif.c, 680
HTIF_CONSOLE_CMD_PUTC
htif.c, 680
HTIF_DATA_BITS
htif.c, 680
HTIF_DATA_MASK
htif.c, 680
HTIF_DATA_SHIFT
htif.c, 680
HTIF_DEV_BITS
htif.c, 680
HTIF_DEV_CONSOLE
htif.c, 680
HTIF_DEV_MASK
htif.c, 680
HTIF_DEV_SHIFT
htif.c, 680
HTIF_DEV_SYSTEM
htif.c, 680
htif_getc
htif.c, 682
htif.h, 331
htif_lock
htif.c, 683
htif_putc
htif.c, 682

htif.h, 331
htif_reset_match
fdt_reset_htif.c, 650
htif_system_reset
htif.c, 683
htif.h, 331

illegal_insn_func
sbi_illegal_insn.c, 401
illegal_insn_table
sbi_illegal_insn.c, 402
IMM_I
riscv_encoding.h, 132
IMM_S
riscv_encoding.h, 132
include/sbi/fw_dynamic.h, 82
include/sbi/riscv_asm.h, 85
include/sbi/riscv_atomic.h, 92
include/sbi/riscv_barrier.h, 96
include/sbi/riscv_encoding.h, 98
include/sbi/riscv_fp.h, 148
include/sbi/riscv_io.h, 150
include/sbi/riscv_locks.h, 155
include/sbi/sbi_bitmap.h, 158
include/sbi/sbi_bitops.h, 164
include/sbi/sbi_console.h, 171
include/sbi/sbi_const.h, 175
include/sbi/sbi_csr_detect.h, 177
include/sbi/sbi_ecall.h, 178
include/sbi/sbi_ecall_interface.h, 184
include/sbi/sbi_emulate_csr.h, 189
include/sbi/sbi_error.h, 191
include/sbi/sbi_fifo.h, 193
include/sbi/sbi_hart.h, 199
include/sbi/sbi_hartmask.h, 207
include/sbi/sbi_hfence.h, 214
include/sbi/sbi_hsm.h, 217
include/sbi/sbi_illegal_insn.h, 223
include/sbi/sbi_init.h, 225
include/sbi/sbi_ipi.h, 229
include/sbi/sbi_list.h, 234
include/sbi/sbi_math.h, 241
include/sbi/sbi_misaligned_ldst.h, 241
include/sbi/sbi_platform.h, 244
include/sbi/sbi_scratch.h, 267
include/sbi/sbi_string.h, 272
include/sbi/sbi_system.h, 276
include/sbi/sbi_timer.h, 277
include/sbi/sbi_tlb.h, 282
include/sbi/sbi_trap.h, 285
include/sbi/sbi_types.h, 293
include/sbi/sbi_unpriv.h, 297
include/sbi/sbi_version.h, 299
include/sbi_utils/fdt/fdt_fixup.h, 300
include/sbi_utils/fdt/fdt_helper.h, 304
include/sbi_utils/ipi/fdt_ipi.h, 312
include/sbi_utils/irqchip/fdt_irqchip.h, 314
include/sbi_utils/irqchip/plic.h, 315
include/sbi_utils/reset/fdt_reset.h, 318
include/sbi_utils/serial/fdt_serial.h, 319
include/sbi_utils/serial/shakti-uart.h, 320
include/sbi_utils/serial/sifive-uart.h, 322
include/sbi_utils/serial/uart8250.h, 324
include/sbi_utils/sys/clint.h, 326
include/sbi_utils/sys/htif.h, 330
include/sbi_utils/sys/sifive_test.h, 332
include/sbi_utils/timer/fdt_timer.h, 333
init
fdt_reset, 59
fdt_serial, 60
init_coldboot
sbi_init.c, 404
init_count_offset
sbi_init.c, 411
init_warmboot
sbi_init.c, 405
INSERT_FIELD
sbi_bitops.h, 167
INSN_16BIT_MASK
riscv_encoding.h, 132
INSN_32BIT_MASK
riscv_encoding.h, 132
INSN_IS_16BIT
riscv_encoding.h, 132
INSN_IS_32BIT
riscv_encoding.h, 132
INSN_LEN
riscv_encoding.h, 133
INSN_MASK_C_FLD
riscv_encoding.h, 133
INSN_MASK_C_FLDSP
riscv_encoding.h, 133
INSN_MASK_C_FLW
riscv_encoding.h, 133
INSN_MASK_C_FLWSP
riscv_encoding.h, 133
INSN_MASK_C_FSD
riscv_encoding.h, 133
INSN_MASK_C_FSDSP
riscv_encoding.h, 133
INSN_MASK_C_FSW
riscv_encoding.h, 133
INSN_MASK_C_FSWSP
riscv_encoding.h, 133
INSN_MASK_C_LD
riscv_encoding.h, 133
INSN_MASK_C_LDSP
riscv_encoding.h, 133
INSN_MASK_C_LW
riscv_encoding.h, 134
INSN_MASK_C_LWSP
riscv_encoding.h, 134
INSN_MASK_C_SD
riscv_encoding.h, 134
INSN_MASK_C_SDSP
riscv_encoding.h, 134
INSN_MASK_C_SW

riscv_encoding.h, 134
INSN_MASK_C_SWSP
riscv_encoding.h, 134
INSN_MASK_FLD
riscv_encoding.h, 134
INSN_MASK_FLQ
riscv_encoding.h, 134
INSN_MASK_FLW
riscv_encoding.h, 134
INSN_MASK_FSD
riscv_encoding.h, 134
INSN_MASK_FSQ
riscv_encoding.h, 134
INSN_MASK_FSW
riscv_encoding.h, 134
INSN_MASK_LB
riscv_encoding.h, 135
INSN_MASK_LBU
riscv_encoding.h, 135
INSN_MASK_LD
riscv_encoding.h, 135
INSN_MASK_LH
riscv_encoding.h, 135
INSN_MASK_LHU
riscv_encoding.h, 135
INSN_MASK_LW
riscv_encoding.h, 135
INSN_MASK_LWU
riscv_encoding.h, 135
INSN_MASK_SB
riscv_encoding.h, 135
INSN_MASK_SD
riscv_encoding.h, 135
INSN_MASK_SH
riscv_encoding.h, 135
INSN_MASK_SW
riscv_encoding.h, 135
INSN_MASK_WFI
riscv_encoding.h, 135
INSN_MATCH_C_FLD
riscv_encoding.h, 136
INSN_MATCH_C_FLDSP
riscv_encoding.h, 136
INSN_MATCH_C_FLW
riscv_encoding.h, 136
INSN_MATCH_C_FLWSP
riscv_encoding.h, 136
INSN_MATCH_C_FSD
riscv_encoding.h, 136
INSN_MATCH_C_FSDSP
riscv_encoding.h, 136
INSN_MATCH_C_FSW
riscv_encoding.h, 136
INSN_MATCH_C_FSWSP
riscv_encoding.h, 136
INSN_MATCH_C_LD
riscv_encoding.h, 136
INSN_MATCH_C_LDSP
riscv_encoding.h, 136
INSN_MATCH_C_LW
riscv_encoding.h, 136
INSN_MATCH_C_LWSP
riscv_encoding.h, 136
INSN_MATCH_C_SD
riscv_encoding.h, 137
INSN_MATCH_C_SDSP
riscv_encoding.h, 137
INSN_MATCH_C_SW
riscv_encoding.h, 137
INSN_MATCH_C_SWSP
riscv_encoding.h, 137
INSN_MATCH_FLD
riscv_encoding.h, 137
INSN_MATCH_FLQ
riscv_encoding.h, 137
INSN_MATCH_FLW
riscv_encoding.h, 137
INSN_MATCH_FSD
riscv_encoding.h, 137
INSN_MATCH_FSQ
riscv_encoding.h, 137
INSN_MATCH_FSW
riscv_encoding.h, 137
INSN_MATCH_LB
riscv_encoding.h, 137
INSN_MATCH_LBU
riscv_encoding.h, 137
INSN_MATCH_LD
riscv_encoding.h, 138
INSN_MATCH_LH
riscv_encoding.h, 138
INSN_MATCH_LHU
riscv_encoding.h, 138
INSN_MATCH_LW
riscv_encoding.h, 138
INSN_MATCH_LWU
riscv_encoding.h, 138
INSN_MATCH_SB
riscv_encoding.h, 138
INSN_MATCH_SD
riscv_encoding.h, 138
INSN_MATCH_SH
riscv_encoding.h, 138
INSN_MATCH_SW
riscv_encoding.h, 138
INSN_MATCH_WFI
riscv_encoding.h, 138
int16_t
sbi_types.h, 296
INT32_MAX
libfdt_env.h, 636
int32_t
sbi_types.h, 296
INT_MAX
libfdt_env.h, 637
ipi

clint_data, 54
ipi_clear
sbi_platform_operations, 70
ipi_clint_cold_init
fdt_ipi_clint.c, 469
ipi_clint_match
fdt_ipi_clint.c, 470
ipi_data_off
sbi_ipi.c, 418
ipi_drivers
fdt_ipi.c, 468
ipi_exit
sbi_platform_operations, 70
ipi_halt_event
sbi_ipi.c, 418
ipi_halt_ops
sbi_ipi.c, 418
ipi_init
sbi_platform_operations, 70
ipi_ops_array
sbi_ipi.c, 418
ipi_send
sbi_platform_operations, 70
ipi_smode_event
sbi_ipi.c, 418
ipi_smode_ops
sbi_ipi.c, 418
IRQ_M_EXT
riscv_encoding.h, 138
IRQ_M_SOFT
riscv_encoding.h, 138
IRQ_M_TIMER
riscv_encoding.h, 139
IRQ_S_EXT
riscv_encoding.h, 139
IRQ_S_GEXT
riscv_encoding.h, 139
IRQ_S_SOFT
riscv_encoding.h, 139
IRQ_S_TIMER
riscv_encoding.h, 139
IRQ_VS_EXT
riscv_encoding.h, 139
IRQ_VS_SOFT
riscv_encoding.h, 139
IRQ_VS_TIMER
riscv_encoding.h, 139
irqchip_drivers
fdt_irqchip.c, 473
irqchip_exit
sbi_platform_operations, 70
irqchip_init
sbi_platform_operations, 70
irqchip_plic_cold_init
fdt_irqchip_plic.c, 474
irqchip_plic_match
fdt_irqchip_plic.c, 475
irqchip_plic_update_hartid_table
fdt_irqchip_plic.c, 474
irqchip_plic_warm_init
fdt_irqchip_plic.c, 475

last_comp_version
fdt_header, 55
last_hartid_having_scratch
sbi_scratch.c, 426
sbi_scratch.h, 272
len
fdt_property, 58
LGREG
riscv_asm.h, 87
lib/sbi/riscv_asm.c, 335
lib/sbi/riscv_atomic.c, 339
lib/sbi/riscv_locks.c, 345
lib/sbi/sbi_bitmap.c, 348
lib/sbi/sbi_bitops.c, 350
lib/sbi/sbi_console.c, 352
lib/sbi/sbi_ecall.c, 361
lib/sbi/sbi_ecall_base.c, 366
lib/sbi/sbi_ecall_hsm.c, 367
lib/sbi/sbi_ecall_legacy.c, 369
lib/sbi/sbi_ecall_replace.c, 371
lib/sbi/sbi_ecall_vendor.c, 373
lib/sbi/sbi_emulate_csr.c, 375
lib/sbi/sbi_fifo.c, 376
lib/sbi/sbi_hart.c, 383
lib/sbi/sbi_hsm.c, 393
lib/sbi/sbi_illegal_insn.c, 400
lib/sbi/sbi_init.c, 402
lib/sbi/sbi_ipi.c, 412
lib/sbi/sbi_math.c, 419
lib/sbi/sbi_misaligned_ldst.c, 419
lib/sbi/sbi_platform.c, 421
lib/sbi/sbi_scratch.c, 423
lib/sbi/sbi_string.c, 426
lib/sbi/sbi_system.c, 430
lib/sbi/sbi_timer.c, 431
lib/sbi/sbi_tlb.c, 436
lib/sbi/sbi_trap.c, 447
lib/sbi/sbi_unpriv.c, 450
lib/utils/fdt/fdt_fixup.c, 452
lib/utils/fdt/fdt_helper.c, 456
lib/utils/ipi/fdt_ipi.c, 465
lib/utils/ipi/fdt_ipi_clint.c, 468
lib/utils/irqchip/fdt_irqchip.c, 470
lib/utils/irqchip/fdt_irqchip_plic.c, 473
lib/utils/irqchip/plic.c, 476
lib/utils/libfdt/fdt.c, 479
lib/utils/libfdt/fdt.h, 489
lib/utils/libfdt/fdt_addresses.c, 491
lib/utils/libfdt/fdt_empty_tree.c, 494
lib/utils/libfdt/fdt_overlay.c, 496
lib/utils/libfdt/fdt_ro.c, 507
lib/utils/libfdt/fdt_rw.c, 538
lib/utils/libfdt/fdt_strerror.c, 552
lib/utils/libfdt/fdt_sw.c, 554
lib/utils/libfdt/fdt_wip.c, 564

lib/utils/libfdt/libfdt.h, 568
lib/utils/libfdt/libfdt_env.h, 634
lib/utils/libfdt/libfdt_internal.h, 640
lib/utils/reset/fdt_reset.c, 648
lib/utils/reset/fdt_reset_htif.c, 649
lib/utils/reset/fdt_reset_sifive.c, 650
lib/utils/serial/fdt_serial.c, 652
lib/utils/serial/fdt_serial_htif.c, 654
lib/utils/serial/fdt_serial_shakti.c, 655
lib/utils/serial/fdt_serial_sifive.c, 657
lib/utils/serial/fdt_serial_uart8250.c, 658
lib/utils/serial/shakti-uart.c, 659
lib/utils/serial/sifive-uart.c, 662
lib/utils/serial/uart8250.c, 667
lib/utils/sys/clint.c, 673
lib/utils/sys/htif.c, 678
lib/utils/sys/sifive_test.c, 683
lib/utils/timer/fdt_timer.c, 685
lib/utils/timer/fdt_timer_clint.c, 689
libfdt.h
    fdt32_ld, 579
    fdt32_st, 579
    fdt64_ld, 580
    fdt64_st, 580
    fdt_add_mem_rsv, 580
    fdt_add_reservemap_entry, 581
    fdt_add_subnode, 581
    fdt_add_subnode_namelen, 582
    fdt_address_cells, 583
    fdt_appendprop, 583
    fdt_appendprop_addrrange, 584
    fdt_appendprop_cell, 584
    fdt_appendprop_string, 574
    fdt_appendprop_u32, 585
    fdt_appendprop_u64, 585
    fdt_begin_node, 586
    fdt_boot_cpuid_phys, 574
    fdt_check_full, 586
    fdt_check_header, 587
    fdt_create, 587
    fdt_create_empty_tree, 588
    FDT_CREATE_FLAG_NO_NAME_DEDUP, 574
    FDT_CREATE_FLAGS_ALL, 574
    fdt_create_with_flags, 589
    fdt_del_mem_rsv, 589
    fdt_del_node, 590
    fdt_delprop, 590
    fdt_end_node, 590
    FDT_ERR_BADFLAGS, 574
    FDT_ERR_BADLAYOUT, 575
    FDT_ERR_BADMAGIC, 575
    FDT_ERR_BADNCELLS, 575
    FDT_ERR_BADOFFSET, 575
    FDT_ERR_BADOVERLAY, 575
    FDT_ERR_BADPATH, 575
    FDT_ERR_BADPHANDLE, 575
    FDT_ERR_BADSTATE, 575
    FDT_ERR_BADSTRUCTURE, 575
    FDT_ERR_BADVALUE, 575
    FDT_ERR_BADVERSION, 575
    FDT_ERR_EXISTS, 575
    FDT_ERR_INTERNAL, 576
    FDT_ERR_MAX, 576
    FDT_ERR_NOPHANDLES, 576
    FDT_ERR_NOSPACE, 576
    FDT_ERR_NOTFOUND, 576
    FDT_ERR_TRUNCATED, 576
    fdt_find_max_phandle, 591
    fdt_finish, 592
    fdt_finish_reservemap, 592
    fdt_first_property_offset, 593
    fdt_first_subnode, 594
    FDT_FIRST_SUPPORTED_VERSION, 576
    fdt_for_each_property_offset, 576
    fdt_for_each_subnode, 576
    fdt_generate_phandle, 594
    fdt_get_alias, 595
    fdt_get_alias_namelen, 595
    fdt_get_header, 577
    fdt_get_max_phandle, 596
    fdt_get_mem_rsv, 596
    fdt_get_name, 597
    fdt_get_path, 598
    fdt_get_phandle, 599
    fdt_get_property, 599
    fdt_get_property_by_offset, 600
    fdt_get_property_namelen, 600
    fdt_get_property_w, 601
    fdt_get_string, 601
    fdt_getprop, 602
    fdt_getprop_by_offset, 603
    fdt_getprop_namelen, 604
    fdt_getprop_namelen_w, 605
    fdt_getprop_w, 605
    fdt_header_size, 606
    fdt_header_size_, 606
    fdt_last_comp_version, 577
    FDT_LAST_SUPPORTED_VERSION, 577
    fdt_magic, 577
    FDT_MAX_NCELLS, 577
    FDT_MAX_PHANDLE, 577
    fdt_move, 607
    fdt_next_node, 607
    fdt_next_property_offset, 608
    fdt_next_subnode, 609
    fdt_next_tag, 609
    fdt_node_check_compatible, 610
    fdt_node_depth, 611
    fdt_node_offset_by_compatible, 612
    fdt_node_offset_by_phandle, 613
    fdt_node_offset_by_prop_value, 614
    fdt_nop_node, 614
    fdt_nop_property, 615
    fdt_num_mem_rsv, 615
    fdt_off_dt_strings, 577
    fdt_off_dt_struct, 577

fdt_off_mem_rsvmap, 578
fdt_offset_ptr, 615
fdt_offset_ptr_w, 616
fdt_open_into, 617
fdt_overlay_apply, 617
fdt_pack, 618
fdt_parent_offset, 618
fdt_path_offset, 619
fdt_path_offset_namelen, 620
fdt_property, 621
fdt_property_cell, 621
fdt_property_placeholder, 622
fdt_property_string, 578
fdt_property_u32, 622
fdt_property_u64, 623
fdt_resize, 623
fdt_set_hdr_, 578, 624
fdt_set_name, 625
fdt_setprop, 625
fdt_setprop_cell, 625
fdt_setprop_empty, 578
fdt_setprop_inplace, 626
fdt_setprop_inplace_cell, 626
fdt_setprop_inplace_namelen_partial, 627
fdt_setprop_inplace_u32, 627
fdt_setprop_inplace_u64, 628
fdt_setprop_placeholder, 628
fdt_setprop_string, 578
fdt_setprop_u32, 629
fdt_setprop_u64, 629
fdt_size_cells, 629
fdt_size_dt_strings, 578
fdt_size_dt_struct, 578
fdt_strerror, 630
fdt_string, 630
fdt_stringlist_contains, 631
fdt_stringlist_count, 631
fdt_stringlist_get, 631
fdt_stringlist_search, 632
fdt_subnode_offset, 632
fdt_subnode_offset_namelen, 633
fdt_supernode_atdepth_offset, 633
fdt_totalsize, 578
fdt_version, 579
libfdt_env.h
CPU_TO_FDT16, 636
cpu_to_fdt16, 638
CPU_TO_FDT32, 636
cpu_to_fdt32, 638
CPU_TO_FDT64, 636
cpu_to_fdt64, 639
EXTRACT_BYTE, 636
fdt16_t, 638
fdt16_to_cpu, 639
fdt32_t, 638
fdt32_to_cpu, 639
fdt64_t, 638
fdt64_to_cpu, 640
FDT_BITWISE, 636
FDT_FORCE, 636
INT32_MAX, 636
INT_MAX, 637
memchr, 637
memcmp, 637
memcpy, 637
memmove, 637
memset, 637
strchr, 637
strcmp, 637
strcpy, 637
strlen, 637
strnlen, 637
strrchr, 637
UINT32_MAX, 638
UINT_MAX, 638
libfdt_internal.h
FDT_ALIGN, 641
fdt_check_node_offset_, 642
fdt_check_prop_offset_, 643
fdt_find_string_, 644
fdt_mem_rsv_, 644
fdt_mem_rsv_w_, 645
fdt_node_end_offset_, 645
fdt_offset_ptr_, 646
fdt_offset_ptr_w_, 646
FDT_RO_PROBE, 641
fdt_ro_probe_, 647
FDT_SW_MAGIC, 642
FDT_TAGALIGN, 642
likely
sbi_types.h, 294
lock
spinlock_t, 78
log2roundup
sbi_math.c, 419
sbi_math.h, 241
LOG_REGBYTES
riscv_encoding.h, 139

magic
fdt_header, 55
fw_dynamic_info, 62
MASK_FUNCT3
riscv_encoding.h, 139
match_table
fdt_ipi, 56
fdt_irqchip, 57
fdt_reset, 59
fdt_serial, 60
fdt_timer, 61
MAX
sbi_types.h, 294
mb
riscv_barrier.h, 97
memchr
libfdt_env.h, 637
memcmp

libfdt_env.h, 637
memcpy
libfdt_env.h, 637
memmove
libfdt_env.h, 637
memset
libfdt_env.h, 637
mepc
sbi_trap_regs, 76
MIN
sbi_types.h, 294
MIP_MEIP
riscv_encoding.h, 139
MIP_MSIP
riscv_encoding.h, 139
MIP_MTIP
riscv_encoding.h, 140
MIP_SEIP
riscv_encoding.h, 140
MIP_SGEIP
riscv_encoding.h, 140
MIP_SSIP
riscv_encoding.h, 140
MIP_STIP
riscv_encoding.h, 140
MIP_VSEIP
riscv_encoding.h, 140
MIP_VSSIP
riscv_encoding.h, 140
MIP_VSTIP
riscv_encoding.h, 140
misa_check_extension
sbi_platform_operations, 70
misa_extension
riscv_asm.h, 87
misa_extension_imp
riscv_asm.c, 336
riscv_asm.h, 89
misa_get_xlen
sbi_platform_operations, 70
misa_string
riscv_asm.c, 336
riscv_asm.h, 89
misa_xlen
riscv_asm.c, 337
riscv_asm.h, 90
mstatus
sbi_trap_regs, 76
MSTATUS32_SD
riscv_encoding.h, 140
MSTATUS64_SD
riscv_encoding.h, 140
MSTATUS_FS
riscv_encoding.h, 140
mstatus_init
sbi_hart.c, 386
MSTATUS_MIE
riscv_encoding.h, 141
MSTATUS_MPIE
riscv_encoding.h, 141
MSTATUS_MPP
riscv_encoding.h, 141
MSTATUS_MPP_SHIFT
riscv_encoding.h, 141
MSTATUS_MPRV
riscv_encoding.h, 141
MSTATUS_MXR
riscv_encoding.h, 141
MSTATUS_SD
riscv_encoding.h, 141
MSTATUS_SIE
riscv_encoding.h, 141
MSTATUS_SPIE
riscv_encoding.h, 141
MSTATUS_SPIE_SHIFT
riscv_encoding.h, 141
MSTATUS_SPP
riscv_encoding.h, 141
MSTATUS_SPP_SHIFT
riscv_encoding.h, 141
MSTATUS_SUM
riscv_encoding.h, 142
MSTATUS_TSR
riscv_encoding.h, 142
MSTATUS_TVM
riscv_encoding.h, 142
MSTATUS_TW
riscv_encoding.h, 142
MSTATUS_UBE
riscv_encoding.h, 142
MSTATUS_VS
riscv_encoding.h, 142
MSTATUS_XS
riscv_encoding.h, 142
mstatusH
sbi_trap_regs, 76
MSTATUSH_MBE
riscv_encoding.h, 142
MSTATUSH_MPV
riscv_encoding.h, 142
MSTATUSH_SBE
riscv_encoding.h, 142

name
fdt_node_header, 58
sbi_ipi_event_ops, 66
sbi_platform, 68
nameoff
fdt_property, 58
next
sbi_dlist, 64
next_addr
fw_dynamic_info, 62
sbi_hart.h, 207
sbi_scratch, 72
next_arg1
sbi_scratch, 72

next_mode
fw_dynamic_info, 62
sbi_hart.h, 207
sbi_scratch, 72
next_virt
sbi_hart.h, 207
nextprop_
fdt_ro.c, 537
NULL
sbi_types.h, 295
num_entries
sbi_fifo, 65
num_src
plic_data, 63

off_dt_strings
fdt_header, 55
off_dt_struct
fdt_header, 55
off_mem_rsvmap
fdt_header, 55
offsetof
sbi_types.h, 295
OPENSBI_VERSION
sbi_version.h, 299
opensbi_version
sbi_platform, 68
OPENSBI_VERSION_MAJOR
sbi_version.h, 300
OPENSBI_VERSION_MINOR
sbi_version.h, 300
options
fw_dynamic_info, 62
sbi_scratch, 72
out_sz
sbi_console.h, 175
overlay_adjust_local_phandles
fdt_overlay.c, 498
overlay_adjust_node_phandles
fdt_overlay.c, 499
overlay_apply_node
fdt_overlay.c, 499
overlay_fixup_one_phandle
fdt_overlay.c, 500
overlay_fixup_phandle
fdt_overlay.c, 501
overlay_fixup_phandles
fdt_overlay.c, 501
overlay_get_target
fdt_overlay.c, 502
overlay_get_target_phandle
fdt_overlay.c, 503
overlay_merge
fdt_overlay.c, 503
overlay_phandle_add_offset
fdt_overlay.c, 504
overlay_symbol_update
fdt_overlay.c, 504
overlay_update_local_node_references
fdt_overlay.c, 505
overlay_update_local_references
fdt_overlay.c, 506

PAD_ALTERNATE
sbi_console.c, 353
PAD_RIGHT
sbi_console.c, 353
PAD_ZERO
sbi_console.c, 353
PAGE_MASK
riscv_asm.h, 88
PAGE_SHIFT
riscv_asm.h, 88
PAGE_SIZE
riscv_asm.h, 88
physical_addr_t
sbi_types.h, 296
physical_size_t
sbi_types.h, 296
PK_SYS_write
htif.c, 680
platform_addr
sbi_scratch, 72
platform_ops_addr
sbi_platform, 68
platform_uart_data, 62
addr, 63
baud, 63
freq, 63
reg_io_width, 63
reg_shift, 63
platform_version
sbi_platform, 68
plic
fdt_irqchip_plic.c, 475
plic.c
plic_cold_irqchip_init, 477
PLIC_CONTEXT_BASE, 476
PLIC_CONTEXT_STRIDE, 476
PLIC_ENABLE_BASE, 477
PLIC_ENABLE_STRIDE, 477
PLIC_PENDING_BASE, 477
PLIC_PRIORITY_BASE, 477
plic_set_ie, 477
plic_set_priority, 478
plic_set_thresh, 478
plic_warm_irqchip_init, 478
plic.h
plic_cold_irqchip_init, 316
plic_set_ie, 316
plic_set_thresh, 317
plic_warm_irqchip_init, 317
plic_cold_irqchip_init
plic.c, 477
plic.h, 316
PLIC_CONTEXT_BASE
plic.c, 476
PLIC_CONTEXT_STRIDE

plic.c, 476
plic_count
fdt_irqchip_plic.c, 475
plic_data, 63
addr, 63
num_src, 63
PLIC_ENABLE_BASE
plic.c, 477
PLIC_ENABLE_STRIDE
plic.c, 477
plic_hartid2context
fdt_irqchip_plic.c, 476
plic_hartid2data
fdt_irqchip_plic.c, 476
PLIC_MAX_NR
fdt_irqchip_plic.c, 474
PLIC_PENDING_BASE
plic.c, 477
PLIC_PRIORITY_BASE
plic.c, 477
plic_set_ie
plic.c, 477
plic.h, 316
plic_set_priority
plic.c, 478
plic_set_thresh
plic.c, 478
plic.h, 317
plic_warm_irqchip_init
plic.c, 478
plic.h, 317
PMP_A
riscv_encoding.h, 142
PMP_A_NA4
riscv_encoding.h, 142
PMP_A_NAPOT
riscv_encoding.h, 143
PMP_A_TOR
riscv_encoding.h, 143
PMP_COUNT
riscv_encoding.h, 143
pmp_get
riscv_asm.c, 337
riscv_asm.h, 90
pmp_init
sbi_hart.c, 387
PMP_L
riscv_encoding.h, 143
PMP_R
riscv_encoding.h, 143
pmp_region_count
sbi_platform_operations, 71
pmp_region_info
sbi_platform_operations, 71
pmp_set
riscv_asm.c, 338
riscv_asm.h, 91
PMP_SHIFT
riscv_encoding.h, 143
PMP_W
riscv_encoding.h, 143
PMP_X
riscv_encoding.h, 143
PRECISION_D
riscv_fp.h, 150
PRECISION_S
riscv_fp.h, 150
prev
sbi_dlist, 64
print
sbi_console.c, 354
PRINT_BUF_LEN
sbi_console.c, 353
printc
sbi_console.c, 355
printi
sbi_console.c, 355
prints
sbi_console.c, 356
probe
sbi_ecall_extension, 65
process
sbi_ipi_event_ops, 66
PRV_M
riscv_encoding.h, 143
PRV_S
riscv_encoding.h, 143
PRV_U
riscv_encoding.h, 143
putc
fdt_serial, 60

qlock
sbi_fifo, 65
queue
sbi_fifo, 65

ra
sbi_trap_regs, 76
readb
riscv_io.h, 152
readb_relaxed
riscv_io.h, 152
readl
riscv_io.h, 152
readl_relaxed
riscv_io.h, 152
README.md, 691
readq
riscv_io.h, 152
readq_relaxed
riscv_io.h, 152
readw
riscv_io.h, 152
readw_relaxed
riscv_io.h, 152
REG_BAUD

shakti-uart.c, 660
REG_CONTROL
shakti-uart.c, 660
REG_DELAY
shakti-uart.c, 660
REG_INT_EN
shakti-uart.c, 661
reg_io_width
platform_uart_data, 63
REG_IQ_CYCLES
shakti-uart.c, 661
REG_L
riscv_asm.h, 88
REG_MASK
riscv_encoding.h, 143
REG_OFFSET
riscv_encoding.h, 144
REG_PTR
riscv_encoding.h, 144
REG_RX
shakti-uart.c, 661
REG_RX_THRES
shakti-uart.c, 661
REG_S
riscv_asm.h, 88
reg_shift
platform_uart_data, 63
REG_STATUS
shakti-uart.c, 661
REG_TX
shakti-uart.c, 661
REGBYTES
riscv_encoding.h, 144
reset_drivers
fdt_reset.c, 649
RISCV_ACQUIRE_BARRIER
riscv_barrier.h, 97
riscv_asm.c
csr_read_num, 335
csr_write_num, 335
ctz, 336
misa_extension_imp, 336
misa_string, 336
misa_xlen, 337
pmp_get, 337
pmp_set, 338
riscv_asm.h
__ASM_STR, 86
csr_clear, 86
csr_read, 86
csr_read_clear, 86
csr_read_num, 88
csr_read_set, 86
csr_set, 87
csr_swap, 87
csr_write, 87
csr_write_num, 89
current_hartid, 87
LGREG, 87
misa_extension, 87
misa_extension_imp, 89
misa_string, 89
misa_xlen, 90
PAGE_MASK, 88
PAGE_SHIFT, 88
PAGE_SIZE, 88
pmp_get, 90
pmp_set, 91
REG_L, 88
REG_S, 88
SZREG, 88
wfi, 88
riscv_atomic.c
__NOP, 341
__NOT, 341
__atomic_op_bit, 340
__atomic_op_bit_ord, 340
__axchg, 340
__cmpxchg, 341
__xchg, 341
atomic_add_return, 342
atomic_clear_bit, 342
atomic_cmpxchg, 342
atomic_raw_clear_bit, 342
atomic_raw_set_bit, 343
atomic_raw_xchg_uint, 343
atomic_raw_xchg_ulong, 343
atomic_read, 344
atomic_set_bit, 344
atomic_sub_return, 344
atomic_write, 345
atomic_xchg, 345
axchg, 341
cmpxchg, 341
xchg, 341
riscv_atomic.h
atomic_add_return, 93
atomic_clear_bit, 93
atomic_cmpxchg, 93
ATOMIC_INIT, 92
ATOMIC_INITIALIZER, 92
atomic_raw_clear_bit, 93
atomic_raw_set_bit, 94
atomic_raw_xchg_uint, 94
atomic_raw_xchg_ulong, 94
atomic_read, 95
atomic_set_bit, 95
atomic_sub_return, 95
atomic_write, 96
atomic_xchg, 96
riscv_barrier.h
__smp_load_acquire, 96
__smp_store_release, 97
cpu_relax, 97
mb, 97
RISCV_ACQUIRE_BARRIER, 97

RISCV_FENCE, 97
RISCV_RELEASE_BARRIER, 97
rmb, 97
smp_mb, 97
smp_rmb, 97
smp_wmb, 98
wmb, 98
riscv_encoding.h
CAUSE_BREAKPOINT, 107
CAUSE_FETCH_ACCESS, 107
CAUSE_FETCH_GUEST_PAGE_FAULT, 107
CAUSE_FETCH_PAGE_FAULT, 107
CAUSE_HYPERVISOR_ECALL, 107
CAUSE_ILLEGAL_INSTRUCTION, 107
CAUSE_LOAD_ACCESS, 108
CAUSE_LOAD_GUEST_PAGE_FAULT, 108
CAUSE_LOAD_PAGE_FAULT, 108
CAUSE_MACHINE_ECALL, 108
CAUSE_MISALIGNED_FETCH, 108
CAUSE_MISALIGNED_LOAD, 108
CAUSE_MISALIGNED_STORE, 108
CAUSE_STORE_ACCESS, 108
CAUSE_STORE_GUEST_PAGE_FAULT, 108
CAUSE_STORE_PAGE_FAULT, 108
CAUSE_SUPERVISOR_ECALL, 108
CAUSE_USER_ECALL, 108
CAUSE_VIRTUAL_INST_FAULT, 109
CSR_CYCLE, 109
CSR_CYCLEH, 109
CSR_DCSR, 109
CSR_DPC, 109
CSR_DSCRATCH, 109
CSR_FCSR, 109
CSR_FFLAGS, 109
CSR_FRM, 109
CSR_HCOUNTERNEN, 109
CSR_HEDELEG, 109
CSR_HGATP, 109
CSR_HGEIE, 110
CSR_HGEIP, 110
CSR_HIDELEG, 110
CSR_HIE, 110
CSR_HIP, 110
CSR_HPMCOUNTER10, 110
CSR_HPMCOUNTER10H, 110
CSR_HPMCOUNTER11, 110
CSR_HPMCOUNTER11H, 110
CSR_HPMCOUNTER12, 110
CSR_HPMCOUNTER12H, 110
CSR_HPMCOUNTER13, 110
CSR_HPMCOUNTER13H, 111
CSR_HPMCOUNTER14, 111
CSR_HPMCOUNTER14H, 111
CSR_HPMCOUNTER15, 111
CSR_HPMCOUNTER15H, 111
CSR_HPMCOUNTER16, 111
CSR_HPMCOUNTER16H, 111
CSR_HPMCOUNTER17, 111
CSR_HPMCOUNTER17H, 111
CSR_HPMCOUNTER18, 111
CSR_HPMCOUNTER18H, 111
CSR_HPMCOUNTER19, 111
CSR_HPMCOUNTER19H, 112
CSR_HPMCOUNTER20, 112
CSR_HPMCOUNTER20H, 112
CSR_HPMCOUNTER21, 112
CSR_HPMCOUNTER21H, 112
CSR_HPMCOUNTER22, 112
CSR_HPMCOUNTER22H, 112
CSR_HPMCOUNTER23, 112
CSR_HPMCOUNTER23H, 112
CSR_HPMCOUNTER24, 112
CSR_HPMCOUNTER24H, 112
CSR_HPMCOUNTER25, 112
CSR_HPMCOUNTER25H, 113
CSR_HPMCOUNTER26, 113
CSR_HPMCOUNTER26H, 113
CSR_HPMCOUNTER27, 113
CSR_HPMCOUNTER27H, 113
CSR_HPMCOUNTER28, 113
CSR_HPMCOUNTER28H, 113
CSR_HPMCOUNTER29, 113
CSR_HPMCOUNTER29H, 113
CSR_HPMCOUNTER3, 113
CSR_HPMCOUNTER30, 113
CSR_HPMCOUNTER30H, 113
CSR_HPMCOUNTER31, 114
CSR_HPMCOUNTER31H, 114
CSR_HPMCOUNTER3H, 114
CSR_HPMCOUNTER4, 114
CSR_HPMCOUNTER4H, 114
CSR_HPMCOUNTER5, 114
CSR_HPMCOUNTER5H, 114
CSR_HPMCOUNTER6, 114
CSR_HPMCOUNTER6H, 114
CSR_HPMCOUNTER7, 114
CSR_HPMCOUNTER7H, 114
CSR_HPMCOUNTER8, 114
CSR_HPMCOUNTER8H, 115
CSR_HPMCOUNTER9, 115
CSR_HPMCOUNTER9H, 115
CSR_HSTATUS, 115
CSR_HTIMEDELTA, 115
CSR_HTIMEDELTAH, 115
CSR_HTINST, 115
CSR_HTVAL, 115
CSR_INSTRET, 115
CSR_INSTRETH, 115
CSR_MARCHID, 115
CSR_MCAUSE, 115
CSR_MCOUNTEREN, 116
CSR_MCYCLE, 116
CSR_MCYCLEH, 116
CSR_MEDELEG, 116
CSR_MEPC, 116
CSR_MHARTID, 116

CSR_MHPMCOUNTER10, 116
CSR_MHPMCOUNTER10H, 116
CSR_MHPMCOUNTER11, 116
CSR_MHPMCOUNTER11H, 116
CSR_MHPMCOUNTER12, 116
CSR_MHPMCOUNTER12H, 116
CSR_MHPMCOUNTER13, 117
CSR_MHPMCOUNTER13H, 117
CSR_MHPMCOUNTER14, 117
CSR_MHPMCOUNTER14H, 117
CSR_MHPMCOUNTER15, 117
CSR_MHPMCOUNTER15H, 117
CSR_MHPMCOUNTER16, 117
CSR_MHPMCOUNTER16H, 117
CSR_MHPMCOUNTER17, 117
CSR_MHPMCOUNTER17H, 117
CSR_MHPMCOUNTER18, 117
CSR_MHPMCOUNTER18H, 117
CSR_MHPMCOUNTER19, 118
CSR_MHPMCOUNTER19H, 118
CSR_MHPMCOUNTER20, 118
CSR_MHPMCOUNTER20H, 118
CSR_MHPMCOUNTER21, 118
CSR_MHPMCOUNTER21H, 118
CSR_MHPMCOUNTER22, 118
CSR_MHPMCOUNTER22H, 118
CSR_MHPMCOUNTER23, 118
CSR_MHPMCOUNTER23H, 118
CSR_MHPMCOUNTER24, 118
CSR_MHPMCOUNTER24H, 118
CSR_MHPMCOUNTER25, 119
CSR_MHPMCOUNTER25H, 119
CSR_MHPMCOUNTER26, 119
CSR_MHPMCOUNTER26H, 119
CSR_MHPMCOUNTER27, 119
CSR_MHPMCOUNTER27H, 119
CSR_MHPMCOUNTER28, 119
CSR_MHPMCOUNTER28H, 119
CSR_MHPMCOUNTER29, 119
CSR_MHPMCOUNTER29H, 119
CSR_MHPMCOUNTER3, 119
CSR_MHPMCOUNTER30, 119
CSR_MHPMCOUNTER30H, 120
CSR_MHPMCOUNTER31, 120
CSR_MHPMCOUNTER31H, 120
CSR_MHPMCOUNTER3H, 120
CSR_MHPMCOUNTER4, 120
CSR_MHPMCOUNTER4H, 120
CSR_MHPMCOUNTER5, 120
CSR_MHPMCOUNTER5H, 120
CSR_MHPMCOUNTER6, 120
CSR_MHPMCOUNTER6H, 120
CSR_MHPMCOUNTER7, 120
CSR_MHPMCOUNTER7H, 120
CSR_MHPMCOUNTER8, 121
CSR_MHPMCOUNTER8H, 121
CSR_MHPMCOUNTER9, 121
CSR_MHPMCOUNTER9H, 121
CSR_MHPMEVENT10, 121
CSR_MHPMEVENT11, 121
CSR_MHPMEVENT12, 121
CSR_MHPMEVENT13, 121
CSR_MHPMEVENT14, 121
CSR_MHPMEVENT15, 121
CSR_MHPMEVENT16, 121
CSR_MHPMEVENT17, 121
CSR_MHPMEVENT18, 122
CSR_MHPMEVENT19, 122
CSR_MHPMEVENT20, 122
CSR_MHPMEVENT21, 122
CSR_MHPMEVENT22, 122
CSR_MHPMEVENT23, 122
CSR_MHPMEVENT24, 122
CSR_MHPMEVENT25, 122
CSR_MHPMEVENT26, 122
CSR_MHPMEVENT27, 122
CSR_MHPMEVENT28, 122
CSR_MHPMEVENT29, 122
CSR_MHPMEVENT3, 123
CSR_MHPMEVENT30, 123
CSR_MHPMEVENT31, 123
CSR_MHPMEVENT4, 123
CSR_MHPMEVENT5, 123
CSR_MHPMEVENT6, 123
CSR_MHPMEVENT7, 123
CSR_MHPMEVENT8, 123
CSR_MHPMEVENT9, 123
CSR_MIDELEG, 123
CSR_MIE, 123
CSR_MIMPID, 123
CSR_MINSTRET, 124
CSR_MINSTRETH, 124
CSR_MIP, 124
CSR_MISA, 124
CSR_MSCRATCH, 124
CSR_MSTATUS, 124
CSR_MSTATUSH, 124
CSR_MTINST, 124
CSR_MTVAL, 124
CSR_MTVAL2, 124
CSR_MTVEC, 124
CSR_MVENDORID, 124
CSR_PMPADDR0, 125
CSR_PMPADDR1, 125
CSR_PMPADDR10, 125
CSR_PMPADDR11, 125
CSR_PMPADDR12, 125
CSR_PMPADDR13, 125
CSR_PMPADDR14, 125
CSR_PMPADDR15, 125
CSR_PMPADDR2, 125
CSR_PMPADDR3, 125
CSR_PMPADDR4, 125
CSR_PMPADDR5, 125
CSR_PMPADDR6, 126
CSR_PMPADDR7, 126

CSR_PMPADDR8, 126
CSR_PMPADDR9, 126
CSR_PMPCFG0, 126
CSR_PMPCFG1, 126
CSR_PMPCFG2, 126
CSR_PMPCFG3, 126
CSR_SATP, 126
CSR_SCAUSE, 126
CSR_SCOUNTEREN, 126
CSR_SEPC, 126
CSR_SIE, 127
CSR_SIP, 127
CSR_SSCRATCH, 127
CSR_SSTATUS, 127
CSR_STVAL, 127
CSR_STVEC, 127
CSR_TDATA1, 127
CSR_TDATA2, 127
CSR_TDATA3, 127
CSR_TIME, 127
CSR_TIMEH, 127
CSR_TSELECT, 127
CSR_UCAUSE, 128
CSR_UEPC, 128
CSR_UIE, 128
CSR_UIP, 128
CSR_USCRATCH, 128
CSR_USTATUS, 128
CSR_UTVAL, 128
CSR_UTVEC, 128
CSR_VSATP, 128
CSR_VSCAUSE, 128
CSR_VSEPC, 128
CSR_VSIE, 128
CSR_VSIP, 129
CSR_VSSCRATCH, 129
CSR_VSSTATUS, 129
CSR_VSTVAL, 129
CSR_VSTVEC, 129
GET_RM, 129
GET_RS1, 129
GET_RS1S, 129
GET_RS2, 129
GET_RS2C, 129
GET_RS2S, 130
GET_SP, 130
HGATP32_MODE_SHIFT, 130
HGATP32_PPN, 130
HGATP32_VMID_MASK, 130
HGATP32_VMID_SHIFT, 130
HGATP64_MODE_SHIFT, 130
HGATP64_PPN, 130
HGATP64_VMID_MASK, 130
HGATP64_VMID_SHIFT, 130
HGATP_MODE_OFF, 130
HGATP_MODE_SHIFT, 131
HGATP_MODE_SV32X4, 131
HGATP_MODE_SV39X4, 131
HGATP_MODE_SV48X4, 131
HGATP_PPN, 131
HGATP_VMID_MASK, 131
HGATP_VMID_SHIFT, 131
HSTATUS_GVA, 131
HSTATUS_HU, 131
HSTATUS_SPV, 131
HSTATUS_SPVP, 131
HSTATUS_VGEIN, 131
HSTATUS_VGEIN_SHIFT, 132
HSTATUS_VSBE, 132
HSTATUS_VTSR, 132
HSTATUS_VTVM, 132
HSTATUS_VTW, 132
IMM_I, 132
IMM_S, 132
INSN_16BIT_MASK, 132
INSN_32BIT_MASK, 132
INSN_IS_16BIT, 132
INSN_IS_32BIT, 132
INSN_LEN, 133
INSN_MASK_C_FLD, 133
INSN_MASK_C_FLDSP, 133
INSN_MASK_C_FLW, 133
INSN_MASK_C_FLWSP, 133
INSN_MASK_C_FSD, 133
INSN_MASK_C_FSDSP, 133
INSN_MASK_C_FSW, 133
INSN_MASK_C_FSWSP, 133
INSN_MASK_C_LD, 133
INSN_MASK_C_LDSP, 133
INSN_MASK_C_LW, 134
INSN_MASK_C_LWSP, 134
INSN_MASK_C_SD, 134
INSN_MASK_C_SDSP, 134
INSN_MASK_C_SW, 134
INSN_MASK_C_SWSP, 134
INSN_MASK_FLD, 134
INSN_MASK_FLQ, 134
INSN_MASK_FLW, 134
INSN_MASK_FSD, 134
INSN_MASK_FSQ, 134
INSN_MASK_FSW, 134
INSN_MASK_LB, 135
INSN_MASK_LBU, 135
INSN_MASK_LD, 135
INSN_MASK_LH, 135
INSN_MASK_LHU, 135
INSN_MASK_LW, 135
INSN_MASK_LWU, 135
INSN_MASK_SB, 135
INSN_MASK_SD, 135
INSN_MASK_SH, 135
INSN_MASK_SW, 135
INSN_MASK_WFI, 135
INSN_MATCH_C_FLD, 136
INSN_MATCH_C_FLDSP, 136
INSN_MATCH_C_FLW, 136

INSN_MATCH_C_FLWSP, 136
INSN_MATCH_C_FSD, 136
INSN_MATCH_C_FSDSP, 136
INSN_MATCH_C_FSW, 136
INSN_MATCH_C_FSWSP, 136
INSN_MATCH_C_LD, 136
INSN_MATCH_C_LDSP, 136
INSN_MATCH_C_LW, 136
INSN_MATCH_C_LWSP, 136
INSN_MATCH_C_SD, 137
INSN_MATCH_C_SDSP, 137
INSN_MATCH_C_SW, 137
INSN_MATCH_C_SWSP, 137
INSN_MATCH_FLD, 137
INSN_MATCH_FLQ, 137
INSN_MATCH_FLW, 137
INSN_MATCH_FSD, 137
INSN_MATCH_FSQ, 137
INSN_MATCH_FSW, 137
INSN_MATCH_LB, 137
INSN_MATCH_LBU, 137
INSN_MATCH_LD, 138
INSN_MATCH_LH, 138
INSN_MATCH_LHU, 138
INSN_MATCH_LW, 138
INSN_MATCH_LWU, 138
INSN_MATCH_SB, 138
INSN_MATCH_SD, 138
INSN_MATCH_SH, 138
INSN_MATCH_SW, 138
INSN_MATCH_WFI, 138
IRQ_M_EXT, 138
IRQ_M_SOFT, 138
IRQ_M_TIMER, 139
IRQ_S_EXT, 139
IRQ_S_GEXT, 139
IRQ_S_SOFT, 139
IRQ_S_TIMER, 139
IRQ_VS_EXT, 139
IRQ_VS_SOFT, 139
IRQ_VS_TIMER, 139
LOG_REGBYTES, 139
MASK_FUNCT3, 139
MIP_MEIP, 139
MIP_MSIP, 139
MIP_MTIP, 140
MIP_SEIP, 140
MIP_SGEIP, 140
MIP_SSIP, 140
MIP_STIP, 140
MIP_VSEIP, 140
MIP_VSSIP, 140
MIP_VSTIP, 140
MSTATUS32_SD, 140
MSTATUS64_SD, 140
MSTATUS_FS, 140
MSTATUS_MIE, 141
MSTATUS_MPIE, 141
MSTATUS_MPP, 141
MSTATUS_MPP_SHIFT, 141
MSTATUS_MPRV, 141
MSTATUS_MXR, 141
MSTATUS_SD, 141
MSTATUS_SIE, 141
MSTATUS_SPIE, 141
MSTATUS_SPIE_SHIFT, 141
MSTATUS_SPP, 141
MSTATUS_SPP_SHIFT, 141
MSTATUS_SUM, 142
MSTATUS_TSR, 142
MSTATUS_TVM, 142
MSTATUS_TW, 142
MSTATUS_UBE, 142
MSTATUS_VS, 142
MSTATUS_XS, 142
MSTATUSH_MBE, 142
MSTATUSH_MPV, 142
MSTATUSH_SBE, 142
PMP_A, 142
PMP_A_NA4, 142
PMP_A_NAPOT, 143
PMP_A_TOR, 143
PMP_COUNT, 143
PMP_L, 143
PMP_R, 143
PMP_SHIFT, 143
PMP_W, 143
PMP_X, 143
PRV_M, 143
PRV_S, 143
PRV_U, 143
REG_MASK, 143
REG_OFFSET, 144
REG_PTR, 144
REGBYTES, 144
RV_X, 144
RVC_LD_IMM, 144
RVC_LDSP_IMM, 144
RVC_LW_IMM, 144
RVC_LWSP_IMM, 144
RVC_RS1S, 145
RVC_RS2, 145
RVC_RS2S, 145
RVC_SDSP_IMM, 145
RVC_SWSP_IMM, 145
SATP32_ASID, 145
SATP32_MODE, 145
SATP32_PPN, 145
SATP64_ASID, 145
SATP64_MODE, 146
SATP64_PPN, 146
SATP_MODE, 146
SATP_MODE_OFF, 146
SATP_MODE_SV32, 146
SATP_MODE_SV39, 146
SATP_MODE_SV48, 146

SATP_MODE_SV57, 146
SATP_MODE_SV64, 146
SET_RD, 146
SH_RD, 146
SH_RS1, 147
SH_RS2, 147
SH_RS2C, 147
SHIFT_RIGHT, 147
SIP_SSIP, 147
SIP_STIP, 147
SSTATUS32_SD, 147
SSTATUS64_SD, 147
SSTATUS64_UXL, 147
SSTATUS_FS, 147
SSTATUS_MXR, 147
SSTATUS_SD, 147
SSTATUS_SIE, 148
SSTATUS_SPIE, 148
SSTATUS_SPIE_SHIFT, 148
SSTATUS_SPP, 148
SSTATUS_SPP_SHIFT, 148
SSTATUS_SUM, 148
SSTATUS_VS, 148
SSTATUS_XS, 148
RISCV_FENCE
riscv_barrier.h, 97
riscv_fp.h
GET_PRECISION, 149
GET_RM, 150
PRECISION_D, 150
PRECISION_S, 150
riscv_io.h
__io_ar, 151
__io_aw, 151
__io_br, 151
__io_bw, 151
__io_rar, 151
__io_raw, 151
__io_rbr, 152
__io_rbw, 152
__raw_readb, 154
__raw_readl, 154
__raw_readq, 154
__raw_readw, 154
__raw_writeb, 154
__raw_writel, 154
__raw_writeq, 154
__raw_writew, 154
readb, 152
readb_relaxed, 152
readl, 152
readl_relaxed, 152
readq, 152
readq_relaxed, 152
readw, 152
readw_relaxed, 152
writeb, 153
writeb_relaxed, 153
writel, 153
writel_relaxed, 153
writeq, 153
writeq_relaxed, 153
writew, 153
writew_relaxed, 153
riscv_locks.c
spin_lock, 345
spin_lock_check, 346
spin_trylock, 347
spin_unlock, 347
riscv_locks.h
__RISCV_SPIN_UNLOCKED, 155
spin_lock, 155
spin_lock_check, 156
SPIN_LOCK_INIT, 155
SPIN_LOCK_INITIALIZER, 155
spin_trylock, 157
spin_unlock, 157
RISCV_RELEASE_BARRIER
riscv_barrier.h, 97
rmb
riscv_barrier.h, 97
ROUNDDOWN
sbi_types.h, 295
ROUNDUP
sbi_types.h, 295
RV_X
riscv_encoding.h, 144
RVC_LD_IMM
riscv_encoding.h, 144
RVC_LDSP_IMM
riscv_encoding.h, 144
RVC_LW_IMM
riscv_encoding.h, 144
RVC_LWSP_IMM
riscv_encoding.h, 144
RVC_RS1S
riscv_encoding.h, 145
RVC_RS2
riscv_encoding.h, 145
RVC_RS2S
riscv_encoding.h, 145
RVC_SDSP_IMM
riscv_encoding.h, 145
RVC_SWSP_IMM
riscv_encoding.h, 145

s0
sbi_trap_regs, 76
s1
sbi_trap_regs, 76
s10
sbi_trap_regs, 77
s11
sbi_trap_regs, 77
s16
sbi_types.h, 296
s2

sbi_trap_regs, 77
s3
sbi_trap_regs, 77
s32
sbi_types.h, 296
s4
sbi_trap_regs, 77
s5
sbi_trap_regs, 77
s6
sbi_trap_regs, 77
s7
sbi_trap_regs, 77
s8
sbi_trap_regs, 77
sbi_types.h, 296
s9
sbi_trap_regs, 77
SATP32_ASID
riscv_encoding.h, 145
SATP32_MODE
riscv_encoding.h, 145
SATP32_PPN
riscv_encoding.h, 145
SATP64_ASID
riscv_encoding.h, 145
SATP64_MODE
riscv_encoding.h, 146
SATP64_PPN
riscv_encoding.h, 146
SATP_MODE
riscv_encoding.h, 146
SATP_MODE_OFF
riscv_encoding.h, 146
SATP_MODE_SV32
riscv_encoding.h, 146
SATP_MODE_SV39
riscv_encoding.h, 146
SATP_MODE_SV48
riscv_encoding.h, 146
SATP_MODE_SV57
riscv_encoding.h, 146
SATP_MODE_SV64
riscv_encoding.h, 146
sbi_bitmap.c
__bitmap_and, 349
__bitmap_or, 349
__bitmap_xor, 349
sbi_bitmap.h
__bitmap_and, 160
__bitmap_or, 160
__bitmap_xor, 160
bitmap_and, 161
bitmap_clear, 161
bitmap_copy, 162
bitmap_estimate_size, 162
bitmap_fill, 162
BITMAP_FIRST_WORD_MASK, 159
BITMAP_LAST_WORD_MASK, 159
bitmap_or, 162
bitmap_set, 163
bitmap_xor, 163
bitmap_zero, 163
bitmap_zero_except, 164
DECLARE_BITMAP, 159
DEFINE_BITMAP, 160
small_const_nbits, 160
sbi_bitops.c
BITOP_WORD, 350
find_first_bit, 351
find_first_zero_bit, 351
find_last_bit, 351
find_next_bit, 351
find_next_zero_bit, 352
sbi_bitops.h
__change_bit, 167
__clear_bit, 167
__ffs, 167
__fls, 168
__set_bit, 168
__test_and_clear_bit, 168
__test_and_set_bit, 169
__test_bit, 169
BIT, 165
BIT_MASK, 165
BIT_WORD, 165
BIT_WORD_OFFSET, 166
BITS_TO_LONGS, 166
EXTRACT_FIELD, 166
ffs, 169
ffz, 166
find_first_bit, 169
find_first_zero_bit, 170
find_last_bit, 170
find_next_bit, 170
find_next_zero_bit, 170
fls, 171
for_each_clear_bit, 166
for_each_clear_bit_from, 166
for_each_set_bit, 166
for_each_set_bit_from, 167
GENMASK, 167
INSERT_FIELD, 167
sbi_boot_prints
sbi_init.c, 406
sbi_console.c
console_out_lock, 361
console_plat, 361
PAD_ALTERNATE, 353
PAD_RIGHT, 353
PAD_ZERO, 353
print, 354
PRINT_BUF_LEN, 353
printc, 355
printi, 355
prints, 356

sbi_console_init, 356
sbi_dprintf, 357
sbi_getc, 358
sbi_gets, 358
sbi_isprintable, 358
sbi_printf, 358
sbi_putc, 359
sbi_puts, 360
sbi_snprintf, 360
sbi_sprintf, 361
va_arg, 353
va_end, 353
va_list, 354
va_start, 354
sbi_console.h
__printf, 172
format, 175
out_sz, 175
sbi_console_init, 172
sbi_getc, 173
sbi_gets, 173
sbi_isprintable, 174
sbi_putc, 174
sbi_puts, 174
sbi_console_init
sbi_console.c, 356
sbi_console.h, 172
sbi_const.h
_AC, 176
_AT, 176
_BITUL, 176
_BITULL, 176
_UL, 176
_ULL, 176
__AC, 175
__STR, 176
STRINGIFY, 176
UL, 176
ULL, 177
sbi_csr_detect.h
csr_read_allowed, 178
csr_write_allowed, 178
sbi_dlist, 64
next, 64
prev, 64
sbi_dprintf
sbi_console.c, 357
SBI_EALREADY
sbi_error.h, 192
sbi_ecall.c
ecall_impid, 365
sbi_ecall_find_extension, 362
sbi_ecall_get_impid, 362
sbi_ecall_handler, 363
sbi_ecall_init, 363
sbi_ecall_register_extension, 364
sbi_ecall_set_impid, 364
sbi_ecall_unregister_extension, 364
sbi_ecall_version_major, 365
sbi_ecall_version_minor, 365
SBI_LIST_HEAD, 365
sbi_ecall.h
ecall_base, 183
ecall_hsm, 183
ecall_ipi, 183
ecall_legacy, 183
ecall_rfence, 183
ecall_time, 183
ecall_vendor, 183
sbi_ecall_find_extension, 180
sbi_ecall_get_impid, 180
sbi_ecall_handler, 181
sbi_ecall_init, 181
sbi_ecall_register_extension, 181
sbi_ecall_set_impid, 182
sbi_ecall_unregister_extension, 182
SBI_ECALL_VERSION_MAJOR, 180
sbi_ecall_version_major, 182
SBI_ECALL_VERSION_MINOR, 180
sbi_ecall_version_minor, 183
SBI_OPENSBI_IMPID, 180
sbi_ecall_base.c
ecall_base, 367
sbi_ecall_base_handler, 366
sbi_ecall_base_probe, 366
sbi_ecall_base_handler
sbi_ecall_base.c, 366
sbi_ecall_base_probe
sbi_ecall_base.c, 366
sbi_ecall_extension, 64
extid_end, 64
extid_start, 64
handle, 64
head, 65
probe, 65
sbi_ecall_find_extension
sbi_ecall.c, 362
sbi_ecall.h, 180
sbi_ecall_get_impid
sbi_ecall.c, 362
sbi_ecall.h, 180
sbi_ecall_handler
sbi_ecall.c, 363
sbi_ecall.h, 181
sbi_ecall_hsm.c
ecall_hsm, 368
sbi_ecall_hsm_handler, 368
sbi_ecall_hsm_handler
sbi_ecall_hsm.c, 368
sbi_ecall_init
sbi_ecall.c, 363
sbi_ecall.h, 181
sbi_ecall_interface.h
SBI_ERR_ALREADY_AVAILABLE, 185
SBI_ERR_DENIED, 185
SBI_ERR_FAILED, 185

SBI_ERR_INVALID_ADDRESS, 185
SBI_ERR_INVALID_PARAM, 185
SBI_ERR_NOT_SUPPORTED, 185
SBI_EXT_0_1_CLEAR_IPI, 185
SBI_EXT_0_1_CONSOLE_GETCHAR, 185
SBI_EXT_0_1_CONSOLE_PUTCHAR, 185
SBI_EXT_0_1_REMOTE_FENCE_I, 186
SBI_EXT_0_1_REMOTE_SFENCE_VMA, 186
SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID, 186
SBI_EXT_0_1_SEND_IPI, 186
SBI_EXT_0_1_SET_TIMER, 186
SBI_EXT_0_1_SHUTDOWN, 186
SBI_EXT_BASE, 186
SBI_EXT_BASE_GET_IMP_ID, 186
SBI_EXT_BASE_GET_IMP_VERSION, 186
SBI_EXT_BASE_GET_MARCHID, 186
SBI_EXT_BASE_GET_MIMPID, 186
SBI_EXT_BASE_GET_MVENDORID, 186
SBI_EXT_BASE_GET_SPEC_VERSION, 187
SBI_EXT_BASE_PROBE_EXT, 187
SBI_EXT_FIRMWARE_END, 187
SBI_EXT_FIRMWARE_START, 187
SBI_EXT_HSM, 187
SBI_EXT_HSM_HART_GET_STATUS, 187
SBI_EXT_HSM_HART_START, 187
SBI_EXT_HSM_HART_STOP, 187
SBI_EXT_IPI, 187
SBI_EXT_IPI_SEND_IPI, 187
SBI_EXT_RFENCE, 187
SBI_EXT_RFENCE_REMOTE_FENCE_I, 187
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA, 188
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID, 188
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA, 188
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID, 188
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA, 188
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID, 188
SBI_EXT_TIME, 188
SBI_EXT_TIME_SET_TIMER, 188
SBI_EXT_VENDOR_END, 188
SBI_EXT_VENDOR_START, 188
SBI_HSM_HART_STATUS_START_PENDING, 188
SBI_HSM_HART_STATUS_STARTED, 188
SBI_HSM_HART_STATUS_STOP_PENDING, 189
SBI_HSM_HART_STATUS_STOPPED, 189
SBI_LAST_ERR, 189
SBI_SPEC_VERSION_MAJOR_MASK, 189
SBI_SPEC_VERSION_MAJOR_OFFSET, 189
SBI_SPEC_VERSION_MINOR_MASK, 189
SBI_SUCCESS, 189
sbi_ecall_ipi_handler
sbi_ecall_replace.c, 371
sbi_ecall_legacy.c
ecall_legacy, 370
sbi_ecall_legacy_handler, 369
sbi_load_hart_mask_unpriv, 370
sbi_ecall_legacy_handler
sbi_ecall_legacy.c, 369
sbi_ecall_register_extension
sbi_ecall.c, 364
sbi_ecall.h, 181
sbi_ecall_replace.c
ecall_ipi, 372
ecall_rfence, 373
ecall_time, 373
sbi_ecall_ipi_handler, 371
sbi_ecall_rfence_handler, 372
sbi_ecall_time_handler, 372
sbi_ecall_rfence_handler
sbi_ecall_replace.c, 372
sbi_ecall_set_impid
sbi_ecall.c, 364
sbi_ecall.h, 182
sbi_ecall_time_handler
sbi_ecall_replace.c, 372
sbi_ecall_unregister_extension
sbi_ecall.c, 364
sbi_ecall.h, 182
sbi_ecall_vendor.c
ecall_vendor, 374
sbi_ecall_vendor_handler, 374
sbi_ecall_vendor_probe, 374
sbi_ecall_vendor_handler
sbi_ecall_vendor.c, 374
sbi_ecall_vendor_probe
sbi_ecall_vendor.c, 374
SBI_ECALL_VERSION_MAJOR
sbi_ecall.h, 180
sbi_ecall_version_major
sbi_ecall.c, 365
sbi_ecall.h, 182
SBI_ECALL_VERSION_MINOR
sbi_ecall.h, 180
sbi_ecall_version_minor
sbi_ecall.c, 365
sbi_ecall.h, 183
SBI_EDENIED
sbi_error.h, 192
SBI_EFAIL
sbi_error.h, 192
SBI_EILL
sbi_error.h, 192
SBI_EINVAL
sbi_error.h, 192
SBI_EINVALID_ADDR
sbi_error.h, 192
SBI_EIO
sbi_error.h, 192
sbi_emulate_csr.c

sbi_emulate_csr_read, 375
sbi_emulate_csr_write, 376
sbi_emulate_csr.h
sbi_emulate_csr_read, 190
sbi_emulate_csr_write, 190
sbi_emulate_csr_read
sbi_emulate_csr.c, 375
sbi_emulate_csr.h, 190
sbi_emulate_csr_write
sbi_emulate_csr.c, 376
sbi_emulate_csr.h, 190
SBI_ENODEV
sbi_error.h, 192
SBI_ENOENT
sbi_error.h, 193
SBI_ENOMEM
sbi_error.h, 193
SBI_ENOSPC
sbi_error.h, 193
SBI_ENOSYS
sbi_error.h, 193
SBI_ENOTSUPP
sbi_error.h, 193
SBI_ERR_ALREADY_AVAILABLE
sbi_ecall_interface.h, 185
SBI_ERR_DENIED
sbi_ecall_interface.h, 185
SBI_ERR_FAILED
sbi_ecall_interface.h, 185
SBI_ERR_INVALID_ADDRESS
sbi_ecall_interface.h, 185
SBI_ERR_INVALID_PARAM
sbi_ecall_interface.h, 185
SBI_ERR_NOT_SUPPORTED
sbi_ecall_interface.h, 185
sbi_error.h
SBI_EALREADY, 192
SBI_EDENIED, 192
SBI_EFAIL, 192
SBI_EILL, 192
SBI_EINVAL, 192
SBI_EINVALID_ADDR, 192
SBI_EIO, 192
SBI_ENODEV, 192
SBI_ENOENT, 193
SBI_ENOMEM, 193
SBI_ENOSPC, 193
SBI_ENOSYS, 193
SBI_ENOTSUPP, 193
SBI_ETIMEDOUT, 193
SBI_ETRAP, 193
SBI_EUNKNOWN, 193
SBI_OK, 193
SBI_ETIMEDOUT
sbi_error.h, 193
SBI_ETRAP
sbi_error.h, 193
SBI_EUNKNOWN
sbi_error.h, 193
sbi_exit
sbi_init.c, 406
sbi_init.h, 226
SBI_EXT_0_1_CLEAR_IPI
sbi_ecall_interface.h, 185
SBI_EXT_0_1_CONSOLE_GETCHAR
sbi_ecall_interface.h, 185
SBI_EXT_0_1_CONSOLE_PUTCHAR
sbi_ecall_interface.h, 185
SBI_EXT_0_1_REMOTE_FENCE_I
sbi_ecall_interface.h, 186
SBI_EXT_0_1_REMOTE_SFENCE_VMA
sbi_ecall_interface.h, 186
SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID
sbi_ecall_interface.h, 186
SBI_EXT_0_1_SEND_IPI
sbi_ecall_interface.h, 186
SBI_EXT_0_1_SET_TIMER
sbi_ecall_interface.h, 186
SBI_EXT_0_1_SHUTDOWN
sbi_ecall_interface.h, 186
SBI_EXT_BASE
sbi_ecall_interface.h, 186
SBI_EXT_BASE_GET_IMP_ID
sbi_ecall_interface.h, 186
SBI_EXT_BASE_GET_IMP_VERSION
sbi_ecall_interface.h, 186
SBI_EXT_BASE_GET_MARCHID
sbi_ecall_interface.h, 186
SBI_EXT_BASE_GET_MIMPID
sbi_ecall_interface.h, 186
SBI_EXT_BASE_GET_MVENDORID
sbi_ecall_interface.h, 186
SBI_EXT_BASE_GET_SPEC_VERSION
sbi_ecall_interface.h, 187
SBI_EXT_BASE_PROBE_EXT
sbi_ecall_interface.h, 187
SBI_EXT_FIRMWARE_END
sbi_ecall_interface.h, 187
SBI_EXT_FIRMWARE_START
sbi_ecall_interface.h, 187
SBI_EXT_HSM
sbi_ecall_interface.h, 187
SBI_EXT_HSM_HART_GET_STATUS
sbi_ecall_interface.h, 187
SBI_EXT_HSM_HART_START
sbi_ecall_interface.h, 187
SBI_EXT_HSM_HART_STOP
sbi_ecall_interface.h, 187
SBI_EXT_IPI
sbi_ecall_interface.h, 187
SBI_EXT_IPI_SEND_IPI
sbi_ecall_interface.h, 187
SBI_EXT_RFENCE
sbi_ecall_interface.h, 187
SBI_EXT_RFENCE_REMOTE_FENCE_I
sbi_ecall_interface.h, 187

SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA
sbi_ecall_interface.h, 188
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID
sbi_ecall_interface.h, 188
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA
sbi_ecall_interface.h, 188
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID
sbi_ecall_interface.h, 188
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA
sbi_ecall_interface.h, 188
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID
sbi_ecall_interface.h, 188
SBI_EXT_TIME
sbi_ecall_interface.h, 188
SBI_EXT_TIME_SET_TIMER
sbi_ecall_interface.h, 188
SBI_EXT_VENDOR_END
sbi_ecall_interface.h, 188
SBI_EXT_VENDOR_START
sbi_ecall_interface.h, 188
sbi_fifo, 65
avail, 65
entry_size, 65
num_entries, 65
qlock, 65
queue, 65
tail, 65
sbi_fifo.c
__sbi_fifo_enqueue, 377
__sbi_fifo_is_empty, 378
__sbi_fifo_is_full, 378
__sbi_fifo_reset, 378
sbi_fifo_avail, 379
sbi_fifo_dequeue, 379
sbi_fifo_enqueue, 380
sbi_fifo_init, 381
sbi_fifo_inplace_update, 381
sbi_fifo_is_empty, 382
sbi_fifo_is_full, 382
sbi_fifo_reset, 383
sbi_fifo.h
sbi_fifo_avail, 196
sbi_fifo_dequeue, 196
sbi_fifo_enqueue, 197
sbi_fifo_init, 197
sbi_fifo_inplace_update, 198
sbi_fifo_inplace_update_types, 194
sbi_fifo_is_empty, 199
sbi_fifo_is_full, 199
SBI_FIFO_SKIP, 196
SBI_FIFO_UNCHANGED, 196
SBI_FIFO_UPDATED, 196
sbi_fifo_avail
sbi_fifo.c, 379
sbi_fifo.h, 196
sbi_fifo_dequeue
sbi_fifo.c, 379
sbi_fifo.h, 196
sbi_fifo_enqueue
sbi_fifo.c, 380
sbi_fifo.h, 197
sbi_fifo_init
sbi_fifo.c, 381
sbi_fifo.h, 197
sbi_fifo_inplace_update
sbi_fifo.c, 381
sbi_fifo.h, 198
sbi_fifo_inplace_update_types
sbi_fifo.h, 194
sbi_fifo_is_empty
sbi_fifo.c, 382
sbi_fifo.h, 199
sbi_fifo_is_full
sbi_fifo.c, 382
sbi_fifo.h, 199
sbi_fifo_reset
sbi_fifo.c, 383
SBI_FIFO_SKIP
sbi_fifo.h, 196
SBI_FIFO_UNCHANGED
sbi_fifo.h, 196
SBI_FIFO_UPDATED
sbi_fifo.h, 196
sbi_get_insn
sbi_unpriv.c, 451
sbi_unpriv.h, 299
sbi_getc
sbi_console.c, 358
sbi_console.h, 173
sbi_gets
sbi_console.c, 358
sbi_console.h, 173
sbi_hart.c
__attribute__, 385
__detect_pmp, 384
__sbi_expected_trap, 385
__sbi_expected_trap_hext, 385
delegate_traps, 385
fp_init, 385
hart_detect_features, 386
hart_features_offset, 393
hart_get_features, 386
mstatus_init, 386
pmp_init, 387
sbi_hart_delegation_dump, 388
sbi_hart_expected_trap, 393
sbi_hart_feature_id2string, 388
sbi_hart_get_features_str, 388
sbi_hart_has_feature, 389
sbi_hart_init, 390
sbi_hart_pmp_check_addr, 391
sbi_hart_pmp_count, 391
sbi_hart_pmp_dump, 392
sbi_hart_pmp_get, 392
sbi_hart.h
__attribute__, 201

arg1, 207
next_addr, 207
next_mode, 207
next_virt, 207
sbi_hart_delegation_dump, 201
sbi_hart_expected_trap, 207
sbi_hart_expected_trap_addr, 201
sbi_hart_features, 200
sbi_hart_get_features_str, 202
sbi_hart_has_feature, 202
SBI_HART_HAS_LAST_FEATURE, 201
SBI_HART_HAS_MCOUNTEREN, 201
SBI_HART_HAS_PMP, 201
SBI_HART_HAS_SCOUNTEREN, 201
SBI_HART_HAS_TIME, 201
sbi_hart_init, 204
sbi_hart_pmp_check_addr, 205
sbi_hart_pmp_count, 205
sbi_hart_pmp_dump, 206
sbi_hart_pmp_get, 206
sbi_hart_delegation_dump
sbi_hart.c, 388
sbi_hart.h, 201
sbi_hart_expected_trap
sbi_hart.c, 393
sbi_hart.h, 207
sbi_hart_expected_trap_addr
sbi_hart.h, 201
sbi_hart_feature_id2string
sbi_hart.c, 388
sbi_hart_features
sbi_hart.h, 200
sbi_hart_get_features_str
sbi_hart.c, 388
sbi_hart.h, 202
sbi_hart_has_feature
sbi_hart.c, 389
sbi_hart.h, 202
SBI_HART_HAS_LAST_FEATURE
sbi_hart.h, 201
SBI_HART_HAS_MCOUNTEREN
sbi_hart.h, 201
SBI_HART_HAS_PMP
sbi_hart.h, 201
SBI_HART_HAS_SCOUNTEREN
sbi_hart.h, 201
SBI_HART_HAS_TIME
sbi_hart.h, 201
sbi_hart_init
sbi_hart.c, 390
sbi_hart.h, 204
sbi_hart_pmp_check_addr
sbi_hart.c, 391
sbi_hart.h, 205
sbi_hart_pmp_count
sbi_hart.c, 391
sbi_hart.h, 205
sbi_hart_pmp_dump
sbi_hart.c, 392
sbi_hart.h, 206
sbi_hart_pmp_get
sbi_hart.c, 392
sbi_hart.h, 206
SBI_HART_STARTED
sbi_hsm.h, 217
SBI_HART_STARTING
sbi_hsm.h, 217
SBI_HART_STOPPED
sbi_hsm.h, 218
SBI_HART_STOPPING
sbi_hsm.h, 218
SBI_HART_UNKNOWN
sbi_hsm.h, 218
sbi_hartid_to_scratch
sbi_scratch.h, 268
sbi_hartmask, 66
DECLARE_BITMAP, 66
sbi_hartmask.h
sbi_hartmask_and, 209
sbi_hartmask_bits, 209
sbi_hartmask_clear_all, 210
sbi_hartmask_clear_hart, 210
sbi_hartmask_for_each_hart, 209
SBI_HARTMASK_INIT, 209
SBI_HARTMASK_INIT_EXCEPT, 209
SBI_HARTMASK_MAX_BITS, 209
sbi_hartmask_or, 211
sbi_hartmask_set_all, 212
sbi_hartmask_set_hart, 212
sbi_hartmask_test_hart, 213
sbi_hartmask_xor, 213
sbi_hartmask_and
sbi_hartmask.h, 209
sbi_hartmask_bits
sbi_hartmask.h, 209
sbi_hartmask_clear_all
sbi_hartmask.h, 210
sbi_hartmask_clear_hart
sbi_hartmask.h, 210
sbi_hartmask_for_each_hart
sbi_hartmask.h, 209
SBI_HARTMASK_INIT
sbi_hartmask.h, 209
SBI_HARTMASK_INIT_EXCEPT
sbi_hartmask.h, 209
SBI_HARTMASK_MAX_BITS
sbi_hartmask.h, 209
sbi_hartmask_or
sbi_hartmask.h, 211
sbi_hartmask_set_all
sbi_hartmask.h, 212
sbi_hartmask_set_hart
sbi_hartmask.h, 212
sbi_hartmask_test_hart
sbi_hartmask.h, 213
sbi_hartmask_xor

sbi_hartmask.h, 213
sbi_hfence.h
__sbi_hfence_gvma_all, 214
__sbi_hfence_gvma_gpa, 215
__sbi_hfence_gvma_vmid, 215
__sbi_hfence_gvma_vmid_gpa, 215
__sbi_hfence_vvma_all, 215
__sbi_hfence_vvma_asid, 216
__sbi_hfence_vvma_asid_va, 216
__sbi_hfence_vvma_va, 216
sbi_hsm.c
hart_data_offset, 400
sbi_hsm_exit, 394
sbi_hsm_hart_get_state, 395
sbi_hsm_hart_start, 395
sbi_hsm_hart_started, 396
sbi_hsm_hart_started_mask, 396
sbi_hsm_hart_state_to_status, 397
sbi_hsm_hart_stop, 397
sbi_hsm_hart_wait, 398
sbi_hsm_init, 399
sbi_hsm_prepare_next_jump, 399
sbi_hsm.h
SBI_HART_STARTED, 217
SBI_HART_STARTING, 217
SBI_HART_STOPPED, 218
SBI_HART_STOPPING, 218
SBI_HART_UNKNOWN, 218
sbi_hsm_exit, 218
sbi_hsm_hart_get_state, 218
sbi_hsm_hart_start, 219
sbi_hsm_hart_started, 220
sbi_hsm_hart_started_mask, 220
sbi_hsm_hart_state_to_status, 221
sbi_hsm_hart_stop, 221
sbi_hsm_init, 222
sbi_hsm_prepare_next_jump, 223
sbi_hsm_exit
sbi_hsm.c, 394
sbi_hsm.h, 218
sbi_hsm_hart_get_state
sbi_hsm.c, 395
sbi_hsm.h, 218
sbi_hsm_hart_start
sbi_hsm.c, 395
sbi_hsm.h, 219
sbi_hsm_hart_started
sbi_hsm.c, 396
sbi_hsm.h, 220
sbi_hsm_hart_started_mask
sbi_hsm.c, 396
sbi_hsm.h, 220
sbi_hsm_hart_state_to_status
sbi_hsm.c, 397
sbi_hsm.h, 221
SBI_HSM_HART_STATUS_START_PENDING
sbi_ecall_interface.h, 188
SBI_HSM_HART_STATUS_STARTED
sbi_ecall_interface.h, 188
SBI_HSM_HART_STATUS_STOP_PENDING
sbi_ecall_interface.h, 189
SBI_HSM_HART_STATUS_STOPPED
sbi_ecall_interface.h, 189
sbi_hsm_hart_stop
sbi_hsm.c, 397
sbi_hsm.h, 221
sbi_hsm_hart_wait
sbi_hsm.c, 398
sbi_hsm_init
sbi_hsm.c, 399
sbi_hsm.h, 222
sbi_hsm_prepare_next_jump
sbi_hsm.c, 399
sbi_hsm.h, 223
sbi_illegal_insn.c
illegal_insn_func, 401
illegal_insn_table, 402
sbi_illegal_insn_handler, 401
system_opcode_insn, 401
truly_illegal_insn, 402
sbi_illegal_insn.h
sbi_illegal_insn_handler, 224
sbi_illegal_insn_handler
sbi_illegal_insn.c, 401
sbi_illegal_insn.h, 224
sbi_init
sbi_init.c, 408
sbi_init.h, 227
sbi_init.c
BANNER, 403
coldboot_done, 411
coldboot_lock, 411
coldboot_lottery, 411
coldboot_wait_hmask, 411
init_coldboot, 404
init_count_offset, 411
init_warmboot, 405
sbi_boot_prints, 406
sbi_exit, 406
sbi_init, 408
sbi_init_count, 409
wait_for_coldboot, 410
wake_coldboot_harts, 410
sbi_init.h
sbi_exit, 226
sbi_init, 227
sbi_init_count, 228
sbi_init_count
sbi_init.c, 409
sbi_init.h, 228
SBI_INIT_LIST_HEAD
sbi_list.h, 235
sbi_ipi.c
ipi_data_off, 418
ipi_halt_event, 418
ipi_halt_ops, 418

ipi_ops_array, 418
ipi_smode_event, 418
ipi_smode_ops, 418
sbi_ipi_clear_smode, 412
sbi_ipi_event_create, 413
sbi_ipi_event_destroy, 413
sbi_ipi_exit, 413
sbi_ipi_init, 414
sbi_ipi_process, 414
sbi_ipi_process_halt, 415
sbi_ipi_process_smode, 415
sbi_ipi_send, 415
sbi_ipi_send_halt, 416
sbi_ipi_send_many, 416
sbi_ipi_send_smode, 417
sbi_ipi.h
sbi_ipi_clear_smode, 230
sbi_ipi_event_create, 230
sbi_ipi_event_destroy, 230
SBI_IPI_EVENT_MAX, 229
sbi_ipi_exit, 230
sbi_ipi_init, 231
sbi_ipi_process, 232
sbi_ipi_send_halt, 232
sbi_ipi_send_many, 233
sbi_ipi_send_smode, 233
sbi_ipi_clear_smode
sbi_ipi.c, 412
sbi_ipi.h, 230
sbi_ipi_event_create
sbi_ipi.c, 413
sbi_ipi.h, 230
sbi_ipi_event_destroy
sbi_ipi.c, 413
sbi_ipi.h, 230
SBI_IPI_EVENT_MAX
sbi_ipi.h, 229
sbi_ipi_event_ops, 66
name, 66
process, 66
sync, 67
update, 67
sbi_ipi_exit
sbi_ipi.c, 413
sbi_ipi.h, 230
sbi_ipi_init
sbi_ipi.c, 414
sbi_ipi.h, 231
sbi_ipi_process
sbi_ipi.c, 414
sbi_ipi.h, 232
sbi_ipi_process_halt
sbi_ipi.c, 415
sbi_ipi_process_smode
sbi_ipi.c, 415
sbi_ipi_send
sbi_ipi.c, 415
sbi_ipi_send_halt
sbi_ipi.c, 416
sbi_ipi.h, 232
sbi_ipi_send_many
sbi_ipi.c, 416
sbi_ipi.h, 233
sbi_ipi_send_smode
sbi_ipi.c, 417
sbi_ipi.h, 233
sbi_isprintable
sbi_console.c, 358
sbi_console.h, 174
SBI_ITLB_FLUSH
sbi_tlb.h, 284
SBI_LAST_ERR
sbi_ecall_interface.h, 189
sbi_list.h
__sbi_list_add, 237
__sbi_list_del, 237
__sbi_list_del_entry, 238
SBI_INIT_LIST_HEAD, 235
sbi_list_add, 238
sbi_list_add_tail, 239
sbi_list_del, 239
sbi_list_del_init, 240
sbi_list_entry, 235
sbi_list_first_entry, 235
sbi_list_for_each, 236
sbi_list_for_each_entry, 236
SBI_LIST_HEAD, 236
SBI_LIST_HEAD_INIT, 236
sbi_list_last_entry, 237
SBI_LIST_POISON_NEXT, 237
SBI_LIST_POISON_PREV, 237
sbi_list_add
sbi_list.h, 238
sbi_list_add_tail
sbi_list.h, 239
sbi_list_del
sbi_list.h, 239
sbi_list_del_init
sbi_list.h, 240
sbi_list_entry
sbi_list.h, 235
sbi_list_first_entry
sbi_list.h, 235
sbi_list_for_each
sbi_list.h, 236
sbi_list_for_each_entry
sbi_list.h, 236
SBI_LIST_HEAD
sbi_ecall.c, 365
sbi_list.h, 236
SBI_LIST_HEAD_INIT
sbi_list.h, 236
sbi_list_last_entry
sbi_list.h, 237
SBI_LIST_POISON_NEXT
sbi_list.h, 237

SBI_LIST_POISON_PREV
sbi_list.h, 237
sbi_load_hart_mask_unpriv
sbi_ecall_legacy.c, 370
sbi_load_u64
sbi_unpriv.c, 452
sbi_math.c
log2roundup, 419
sbi_math.h
log2roundup, 241
sbi_memchr
sbi_string.c, 426
sbi_string.h, 273
sbi_memcmp
sbi_string.c, 427
sbi_string.h, 273
sbi_memcpy
sbi_string.c, 427
sbi_string.h, 273
sbi_memmove
sbi_string.c, 427
sbi_string.h, 273
sbi_memset
sbi_string.c, 427
sbi_string.h, 274
sbi_misaligned_ldst.c
sbi_misaligned_load_handler, 420
sbi_misaligned_store_handler, 420
sbi_misaligned_ldst.h
sbi_misaligned_load_handler, 242
sbi_misaligned_store_handler, 243
sbi_misaligned_load_handler
sbi_misaligned_ldst.c, 420
sbi_misaligned_ldst.h, 242
sbi_misaligned_store_handler
sbi_misaligned_ldst.c, 420
sbi_misaligned_ldst.h, 243
SBI_OK
sbi_error.h, 193
SBI_OPENSBI_IMPID
sbi_ecall.h, 180
sbi_platform, 67
features, 67
firmware_context, 67
hart_count, 67
hart_index2id, 68
hart_stack_size, 68
name, 68
opensbi_version, 68
platform_ops_addr, 68
platform_version, 68
sbi_platform.c
sbi_platform_feature_id2string, 421
sbi_platform_get_features_str, 422
sbi_platform_hart_index, 422
sbi_platform.h
__packed, 266
sbi_platform_console_getc, 248
sbi_platform_console_init, 249
sbi_platform_console_putc, 249
SBI_PLATFORM_DEFAULT_FEATURES, 246
SBI_PLATFORM_DEFAULT_HART_STACK_SIZE, 246
sbi_platform_early_exit, 250
sbi_platform_early_init, 250
sbi_platform_features, 248
SBI_PLATFORM_FEATURES_OFFSET, 246
sbi_platform_final_exit, 251
sbi_platform_final_init, 251
SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET, 246
sbi_platform_get_features, 252
sbi_platform_get_features_str, 252
sbi_platform_hart_count, 253
SBI_PLATFORM_HART_COUNT_OFFSET, 246
sbi_platform_hart_index, 253
SBI_PLATFORM_HART_INDEX2ID_OFFSET, 246
sbi_platform_hart_invalid, 254
sbi_platform_hart_stack_size, 254
SBI_PLATFORM_HART_STACK_SIZE_OFFSET, 246
sbi_platform_hart_start, 255
sbi_platform_hart_stop, 255
SBI_PLATFORM_HAS_HART_HOTPLUG, 248
sbi_platform_has_hart_hotplug, 246
SBI_PLATFORM_HAS_HART_SECONDARY_BOOT, 248
sbi_platform_has_hart_secondary_boot, 246
SBI_PLATFORM_HAS_LAST_FEATURE, 248
SBI_PLATFORM_HAS_MFAULTS_DELEGATION, 248
sbi_platform_has_mfaults_delegation, 246
SBI_PLATFORM_HAS_TIMER_VALUE, 248
sbi_platform_has_timer_value, 246
sbi_platform_ipi_clear, 256
sbi_platform_ipi_exit, 256
sbi_platform_ipi_init, 257
sbi_platform_ipi_send, 257
sbi_platform_irqchip_exit, 257
sbi_platform_irqchip_init, 258
sbi_platform_misa_extension, 258
sbi_platform_misa_xlen, 260
sbi_platform_name, 260
SBI_PLATFORM_NAME_OFFSET, 247
SBI_PLATFORM_OPENSBI_VERSION_OFFSET, 247
sbi_platform_ops, 247
SBI_PLATFORM_OPS_OFFSET, 247
sbi_platform_pmp_region_count, 261
sbi_platform_pmp_region_info, 261
sbi_platform_ptr, 247
SBI_PLATFORM_RESET_COLD, 247
SBI_PLATFORM_RESET_SHUTDOWN, 247
SBI_PLATFORM_RESET_WARM, 247
sbi_platform_system_reset, 262

sbi_platform_thishart_ptr, 247
sbi_platform_timer_event_start, 262
sbi_platform_timer_event_stop, 263
sbi_platform_timer_exit, 263
sbi_platform_timer_init, 264
sbi_platform_timer_value, 264
SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT, 247
sbi_platform_tlbr_flush_limit, 265
sbi_platform_vendor_ext_check, 265
sbi_platform_vendor_ext_provider, 266
SBI_PLATFORM_VERSION, 247
SBI_PLATFORM_VERSION_OFFSET, 248
sbi_platform_console_getc
sbi_platform.h, 248
sbi_platform_console_init
sbi_platform.h, 249
sbi_platform_console_putc
sbi_platform.h, 249
SBI_PLATFORM_DEFAULT_FEATURES
sbi_platform.h, 246
SBI_PLATFORM_DEFAULT_HART_STACK_SIZE
sbi_platform.h, 246
sbi_platform_early_exit
sbi_platform.h, 250
sbi_platform_early_init
sbi_platform.h, 250
sbi_platform_feature_id2string
sbi_platform.c, 421
sbi_platform_features
sbi_platform.h, 248
SBI_PLATFORM_FEATURES_OFFSET
sbi_platform.h, 246
sbi_platform_final_exit
sbi_platform.h, 251
sbi_platform_final_init
sbi_platform.h, 251
SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET
sbi_platform.h, 246
sbi_platform_get_features
sbi_platform.h, 252
sbi_platform_get_features_str
sbi_platform.c, 422
sbi_platform.h, 252
sbi_platform_hart_count
sbi_platform.h, 253
SBI_PLATFORM_HART_COUNT_OFFSET
sbi_platform.h, 246
sbi_platform_hart_index
sbi_platform.c, 422
sbi_platform.h, 253
SBI_PLATFORM_HART_INDEX2ID_OFFSET
sbi_platform.h, 246
sbi_platform_hart_invalid
sbi_platform.h, 254
sbi_platform_hart_stack_size
sbi_platform.h, 254
SBI_PLATFORM_HART_STACK_SIZE_OFFSET
sbi_platform.h, 246
sbi_platform_hart_start
sbi_platform.h, 255
sbi_platform_hart_stop
sbi_platform.h, 255
SBI_PLATFORM_HAS_HART_HOTPLUG
sbi_platform.h, 248
sbi_platform_has_hart_hotplug
sbi_platform.h, 246
SBI_PLATFORM_HAS_HART_SECONDARY_BOOT
sbi_platform.h, 248
sbi_platform_has_hart_secondary_boot
sbi_platform.h, 246
SBI_PLATFORM_HAS_LAST_FEATURE
sbi_platform.h, 248
SBI_PLATFORM_HAS_MFAULTS_DELEGATION
sbi_platform.h, 248
sbi_platform_has_mfaults_delegation
sbi_platform.h, 246
SBI_PLATFORM_HAS_TIMER_VALUE
sbi_platform.h, 248
sbi_platform_has_timer_value
sbi_platform.h, 246
sbi_platform_ipi_clear
sbi_platform.h, 256
sbi_platform_ipi_exit
sbi_platform.h, 256
sbi_platform_ipi_init
sbi_platform.h, 257
sbi_platform_ipi_send
sbi_platform.h, 257
sbi_platform_irqchip_exit
sbi_platform.h, 257
sbi_platform_irqchip_init
sbi_platform.h, 258
sbi_platform_misa_extension
sbi_platform.h, 258
sbi_platform_misa_xlen
sbi_platform.h, 260
sbi_platform_name
sbi_platform.h, 260
SBI_PLATFORM_NAME_OFFSET
sbi_platform.h, 247
SBI_PLATFORM_OPENSBI_VERSION_OFFSET
sbi_platform.h, 247
sbi_platform_operations, 68
console_getc, 69
console_init, 69
console_putc, 69
early_exit, 69
early_init, 69
final_exit, 69
final_init, 70
get_tlbr_flush_limit, 70
hart_start, 70
hart_stop, 70
ipi_clear, 70
ipi_exit, 70

ipi_init, 70
ipi_send, 70
irqchip_exit, 70
irqchip_init, 70
misa_check_extension, 70
misa_get_xlen, 70
pmp_region_count, 71
pmp_region_info, 71
system_reset, 71
timer_event_start, 71
timer_event_stop, 71
timer_exit, 71
timer_init, 71
timer_value, 71
vendor_ext_check, 71
vendor_ext_provider, 71
sbi_platform_ops
sbi_platform.h, 247
SBI_PLATFORM_OPS_OFFSET
sbi_platform.h, 247
sbi_platform_pmp_region_count
sbi_platform.h, 261
sbi_platform_pmp_region_info
sbi_platform.h, 261
sbi_platform_ptr
sbi_platform.h, 247
SBI_PLATFORM_RESET_COLD
sbi_platform.h, 247
SBI_PLATFORM_RESET_SHUTDOWN
sbi_platform.h, 247
SBI_PLATFORM_RESET_WARM
sbi_platform.h, 247
sbi_platform_system_reset
sbi_platform.h, 262
sbi_platform_thishart_ptr
sbi_platform.h, 247
sbi_platform_timer_event_start
sbi_platform.h, 262
sbi_platform_timer_event_stop
sbi_platform.h, 263
sbi_platform_timer_exit
sbi_platform.h, 263
sbi_platform_timer_init
sbi_platform.h, 264
sbi_platform_timer_value
sbi_platform.h, 264
SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT
sbi_platform.h, 247
sbi_platform_tlbr_flush_limit
sbi_platform.h, 265
sbi_platform_vendor_ext_check
sbi_platform.h, 265
sbi_platform_vendor_ext_provider
sbi_platform.h, 266
SBI_PLATFORM_VERSION
sbi_platform.h, 247
SBI_PLATFORM_VERSION_OFFSET
sbi_platform.h, 248
sbi_printf
sbi_console.c, 358
sbi_putc
sbi_console.c, 359
sbi_console.h, 174
sbi_puts
sbi_console.c, 360
sbi_console.h, 174
sbi_scratch, 72
fw_size, 72
fw_start, 72
hartid_to_scratch, 72
next_addr, 72
next_arg1, 72
next_mode, 72
options, 72
platform_addr, 72
tmp0, 73
warmboot_addr, 73
sbi_scratch.c
extra_lock, 425
extra_offset, 425
hartid2scratch, 424
hartid_to_scratch_table, 425
last_hartid_having_scratch, 426
sbi_scratch_alloc_offset, 424
sbi_scratch_free_offset, 424
sbi_scratch_init, 425
sbi_scratch.h
__packed, 272
hartid_to_scratch_table, 272
last_hartid_having_scratch, 272
sbi_hartid_to_scratch, 268
sbi_scratch_alloc_offset, 270
SBI_SCRATCH_DEBUG_PRINTS, 270
SBI_SCRATCH_EXTRA_SPACE_OFFSET, 268
sbi_scratch_free_offset, 271
SBI_SCRATCH_FW_SIZE_OFFSET, 268
SBI_SCRATCH_FW_START_OFFSET, 268
SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET, 268
sbi_scratch_init, 271
sbi_scratch_last_hartid, 268
SBI_SCRATCH_NEXT_ADDR_OFFSET, 269
SBI_SCRATCH_NEXT_ARG1_OFFSET, 269
SBI_SCRATCH_NEXT_MODE_OFFSET, 269
SBI_SCRATCH_NO_BOOT_PRINTS, 270
sbi_scratch_offset_ptr, 269
sbi_scratch_options, 270
SBI_SCRATCH_OPTIONS_OFFSET, 269
SBI_SCRATCH_PLATFORM_ADDR_OFFSET, 269
SBI_SCRATCH_SIZE, 269
sbi_scratch_thishart_arg1_ptr, 269
sbi_scratch_thishart_offset_ptr, 269
sbi_scratch_thishart_ptr, 269
SBI_SCRATCH_TMP0_OFFSET, 269

SBI_SCRATCH_WARMBOOT_ADDR_OFFSET, 270
sbi_scratch_alloc_offset
sbi_scratch.c, 424
sbi_scratch.h, 270
SBI_SCRATCH_DEBUG_PRINTS
sbi_scratch.h, 270
SBI_SCRATCH_EXTRA_SPACE_OFFSET
sbi_scratch.h, 268
sbi_scratch_free_offset
sbi_scratch.c, 424
sbi_scratch.h, 271
SBI_SCRATCH_FW_SIZE_OFFSET
sbi_scratch.h, 268
SBI_SCRATCH_FW_START_OFFSET
sbi_scratch.h, 268
SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET
sbi_scratch.h, 268
sbi_scratch_init
sbi_scratch.c, 425
sbi_scratch.h, 271
sbi_scratch_last_hartid
sbi_scratch.h, 268
SBI_SCRATCH_NEXT_ADDR_OFFSET
sbi_scratch.h, 269
SBI_SCRATCH_NEXT_ARG1_OFFSET
sbi_scratch.h, 269
SBI_SCRATCH_NEXT_MODE_OFFSET
sbi_scratch.h, 269
SBI_SCRATCH_NO_BOOT_PRINTS
sbi_scratch.h, 270
sbi_scratch_offset_ptr
sbi_scratch.h, 269
sbi_scratch_options
sbi_scratch.h, 270
SBI_SCRATCH_OPTIONS_OFFSET
sbi_scratch.h, 269
SBI_SCRATCH_PLATFORM_ADDR_OFFSET
sbi_scratch.h, 269
SBI_SCRATCH_SIZE
sbi_scratch.h, 269
sbi_scratch_thishart_arg1_ptr
sbi_scratch.h, 269
sbi_scratch_thishart_offset_ptr
sbi_scratch.h, 269
sbi_scratch_thishart_ptr
sbi_scratch.h, 269
SBI_SCRATCH_TMP0_OFFSET
sbi_scratch.h, 269
SBI_SCRATCH_WARMBOOT_ADDR_OFFSET
sbi_scratch.h, 270
sbi_snprintf
sbi_console.c, 360
SBI_SPEC_VERSION_MAJOR_MASK
sbi_ecall_interface.h, 189
SBI_SPEC_VERSION_MAJOR_OFFSET
sbi_ecall_interface.h, 189
SBI_SPEC_VERSION_MINOR_MASK
sbi_ecall_interface.h, 189
sbi_sprintf
sbi_console.c, 361
sbi_store_u64
sbi_unpriv.c, 452
sbi_strchr
sbi_string.c, 428
sbi_string.h, 274
sbi_strcmp
sbi_string.c, 428
sbi_string.h, 274
sbi_strcpy
sbi_string.c, 428
sbi_string.h, 275
sbi_string.c
sbi_memchr, 426
sbi_memcmp, 427
sbi_memcpy, 427
sbi_memmove, 427
sbi_memset, 427
sbi_strchr, 428
sbi_strcmp, 428
sbi_strcpy, 428
sbi_strlen, 428
sbi_strncpy, 429
sbi_strnlen, 429
sbi_strrchr, 429
sbi_string.h
sbi_memchr, 273
sbi_memcmp, 273
sbi_memcpy, 273
sbi_memmove, 273
sbi_memset, 274
sbi_strchr, 274
sbi_strcmp, 274
sbi_strcpy, 275
sbi_strlen, 275
sbi_strncpy, 275
sbi_strnlen, 275
sbi_strrchr, 276
sbi_strlen
sbi_string.c, 428
sbi_string.h, 275
sbi_strncpy
sbi_string.c, 429
sbi_string.h, 275
sbi_strnlen
sbi_string.c, 429
sbi_string.h, 275
sbi_strrchr
sbi_string.c, 429
sbi_string.h, 276
SBI_SUCCESS
sbi_ecall_interface.h, 189
sbi_system.c
sbi_system_reset, 430
sbi_system.h
sbi_system_reset, 277

sbi_system_reset
sbi_system.c, 430
sbi_system.h, 277
sbi_timer.c
get_ticks, 432
get_time_val, 436
sbi_timer_event_start, 432
sbi_timer_exit, 432
sbi_timer_get_delta, 433
sbi_timer_init, 433
sbi_timer_process, 434
sbi_timer_set_delta, 434
sbi_timer_set_delta_upper, 435
sbi_timer_value, 435
sbi_timer_virt_value, 435
time_delta_off, 436
sbi_timer.h
sbi_timer_event_start, 278
sbi_timer_exit, 278
sbi_timer_get_delta, 279
sbi_timer_init, 279
sbi_timer_process, 280
sbi_timer_set_delta, 280
sbi_timer_set_delta_upper, 281
sbi_timer_value, 281
sbi_timer_virt_value, 281
sbi_timer_event_start
sbi_timer.c, 432
sbi_timer.h, 278
sbi_timer_exit
sbi_timer.c, 432
sbi_timer.h, 278
sbi_timer_get_delta
sbi_timer.c, 433
sbi_timer.h, 279
sbi_timer_init
sbi_timer.c, 433
sbi_timer.h, 279
sbi_timer_process
sbi_timer.c, 434
sbi_timer.h, 280
sbi_timer_set_delta
sbi_timer.c, 434
sbi_timer.h, 280
sbi_timer_set_delta_upper
sbi_timer.c, 435
sbi_timer.h, 281
sbi_timer_value
sbi_timer.c, 435
sbi_timer.h, 281
sbi_timer_virt_value
sbi_timer.c, 435
sbi_timer.h, 281
sbi_tlb.c
__sbi_tlb_range_check, 437
sbi_tlb_entry_process, 437
sbi_tlb_flush_all, 438
sbi_tlb_hfence_gvma, 438
sbi_tlb_hfence_gvma_vmid, 439
sbi_tlb_hfence_vvma, 439
sbi_tlb_hfence_vvma_asid, 440
sbi_tlb_init, 441
sbi_tlb_local_flush, 441
sbi_tlb_process, 442
sbi_tlb_process_count, 442
sbi_tlb_request, 443
sbi_tlb_sfence_vma, 444
sbi_tlb_sfence_vma_asid, 444
sbi_tlb_sync, 445
sbi_tlb_update, 445
sbi_tlb_update_cb, 446
tlb_event, 446
tlb_fifo_mem_off, 446
tlb_fifo_off, 447
tlb_ops, 447
tlb_range_flush_limit, 447
tlb_sync_off, 447
sbi_tlb.h
SBI_ITLB_FLUSH, 284
SBI_TLB_FIFO_NUM_ENTRIES, 283
SBI_TLB_FLUSH_ALL, 283
SBI_TLB_FLUSH_GVMA, 284
SBI_TLB_FLUSH_GVMA_VMID, 284
SBI_TLB_FLUSH_VMA, 284
SBI_TLB_FLUSH_VMA_ASID, 284
SBI_TLB_FLUSH_VVMA, 284
SBI_TLB_FLUSH_VVMA_ASID, 284
SBI_TLB_INFO_INIT, 283
SBI_TLB_INFO_SIZE, 283
sbi_tlb_info_types, 284
sbi_tlb_init, 284
sbi_tlb_request, 285
sbi_tlb_entry_process
sbi_tlb.c, 437
SBI_TLB_FIFO_NUM_ENTRIES
sbi_tlb.h, 283
SBI_TLB_FLUSH_ALL
sbi_tlb.h, 283
sbi_tlb_flush_all
sbi_tlb.c, 438
SBI_TLB_FLUSH_GVMA
sbi_tlb.h, 284
SBI_TLB_FLUSH_GVMA_VMID
sbi_tlb.h, 284
SBI_TLB_FLUSH_VMA
sbi_tlb.h, 284
SBI_TLB_FLUSH_VMA_ASID
sbi_tlb.h, 284
SBI_TLB_FLUSH_VVMA
sbi_tlb.h, 284
SBI_TLB_FLUSH_VVMA_ASID
sbi_tlb.h, 284
sbi_tlb_hfence_gvma
sbi_tlb.c, 438
sbi_tlb_hfence_gvma_vmid
sbi_tlb.c, 439

sbi_tlb_hfence_vvma
sbi_tlb.c, 439
sbi_tlb_hfence_vvma_asid
sbi_tlb.c, 440
sbi_tlb_info, 73
asid, 73
size, 73
smask, 73
start, 73
type, 73
vmid, 73
SBI_TLB_INFO_INIT
sbi_tlb.h, 283
SBI_TLB_INFO_SIZE
sbi_tlb.h, 283
sbi_tlb_info_types
sbi_tlb.h, 284
sbi_tlb_init
sbi_tlb.c, 441
sbi_tlb.h, 284
sbi_tlb_local_flush
sbi_tlb.c, 441
sbi_tlb_process
sbi_tlb.c, 442
sbi_tlb_process_count
sbi_tlb.c, 442
sbi_tlb_request
sbi_tlb.c, 443
sbi_tlb.h, 285
sbi_tlb_sfence_vma
sbi_tlb.c, 444
sbi_tlb_sfence_vma_asid
sbi_tlb.c, 444
sbi_tlb_sync
sbi_tlb.c, 445
sbi_tlb_update
sbi_tlb.c, 445
sbi_tlb_update_cb
sbi_tlb.c, 446
sbi_trap.c
sbi_trap_error, 448
sbi_trap_handler, 448
sbi_trap_redirect, 449
sbi_trap.h
__packed, 292
sbi_trap_handler, 291
SBI_TRAP_INFO_cause, 287
SBI_TRAP_INFO_epc, 287
SBI_TRAP_INFO_last, 287
SBI_TRAP_INFO_OFFSET, 287
SBI_TRAP_INFO_SIZE, 287
SBI_TRAP_INFO_tinst, 288
SBI_TRAP_INFO_tval, 288
SBI_TRAP_INFO_tval2, 288
sbi_trap_redirect, 292
SBI_TRAP_REGS_a0, 288
SBI_TRAP_REGS_a1, 288
SBI_TRAP_REGS_a2, 288
SBI_TRAP_REGS_a3, 288
SBI_TRAP_REGS_a4, 288
SBI_TRAP_REGS_a5, 288
SBI_TRAP_REGS_a6, 288
SBI_TRAP_REGS_a7, 288
SBI_TRAP_REGS_gp, 288
SBI_TRAP_REGS_last, 289
SBI_TRAP_REGS_mepc, 289
SBI_TRAP_REGS_mstatus, 289
SBI_TRAP_REGS_mstatusH, 289
SBI_TRAP_REGS_OFFSET, 289
SBI_TRAP_REGS_ra, 289
SBI_TRAP_REGS_s0, 289
SBI_TRAP_REGS_s1, 289
SBI_TRAP_REGS_s10, 289
SBI_TRAP_REGS_s11, 289
SBI_TRAP_REGS_s2, 289
SBI_TRAP_REGS_s3, 289
SBI_TRAP_REGS_s4, 290
SBI_TRAP_REGS_s5, 290
SBI_TRAP_REGS_s6, 290
SBI_TRAP_REGS_s7, 290
SBI_TRAP_REGS_s8, 290
SBI_TRAP_REGS_s9, 290
SBI_TRAP_REGS_SIZE, 290
SBI_TRAP_REGS_sp, 290
SBI_TRAP_REGS_t0, 290
SBI_TRAP_REGS_t1, 290
SBI_TRAP_REGS_t2, 290
SBI_TRAP_REGS_t3, 290
SBI_TRAP_REGS_t4, 291
SBI_TRAP_REGS_t5, 291
SBI_TRAP_REGS_t6, 291
SBI_TRAP_REGS_tp, 291
SBI_TRAP_REGS_zero, 291
sbi_trap_error
sbi_trap.c, 448
sbi_trap_handler
sbi_trap.c, 448
sbi_trap.h, 291
sbi_trap_info, 74
cause, 74
epc, 74
tinst, 74
tval, 74
tval2, 74
SBI_TRAP_INFO_cause
sbi_trap.h, 287
SBI_TRAP_INFO_epc
sbi_trap.h, 287
SBI_TRAP_INFO_last
sbi_trap.h, 287
SBI_TRAP_INFO_OFFSET
sbi_trap.h, 287
SBI_TRAP_INFO_SIZE
sbi_trap.h, 287
SBI_TRAP_INFO_tinst
sbi_trap.h, 288

SBI_TRAP_INFO_tval
sbi_trap.h, 288
SBI_TRAP_INFO_tval2
sbi_trap.h, 288
sbi_trap_redirect
sbi_trap.c, 449
sbi_trap.h, 292
sbi_trap_regs, 74
a0, 75
a1, 75
a2, 75
a3, 76
a4, 76
a5, 76
a6, 76
a7, 76
gp, 76
mepc, 76
mstatus, 76
mstatusH, 76
ra, 76
s0, 76
s1, 76
s10, 77
s11, 77
s2, 77
s3, 77
s4, 77
s5, 77
s6, 77
s7, 77
s8, 77
s9, 77
sp, 77
t0, 77
t1, 78
t2, 78
t3, 78
t4, 78
t5, 78
t6, 78
tp, 78
zero, 78
SBI_TRAP_REGS_a0
sbi_trap.h, 288
SBI_TRAP_REGS_a1
sbi_trap.h, 288
SBI_TRAP_REGS_a2
sbi_trap.h, 288
SBI_TRAP_REGS_a3
sbi_trap.h, 288
SBI_TRAP_REGS_a4
sbi_trap.h, 288
SBI_TRAP_REGS_a5
sbi_trap.h, 288
SBI_TRAP_REGS_a6
sbi_trap.h, 288
SBI_TRAP_REGS_a7
sbi_trap.h, 288
SBI_TRAP_REGS_gp
sbi_trap.h, 288
SBI_TRAP_REGS_last
sbi_trap.h, 289
SBI_TRAP_REGS_mepc
sbi_trap.h, 289
SBI_TRAP_REGS_mstatus
sbi_trap.h, 289
SBI_TRAP_REGS_mstatusH
sbi_trap.h, 289
SBI_TRAP_REGS_OFFSET
sbi_trap.h, 289
SBI_TRAP_REGS_ra
sbi_trap.h, 289
SBI_TRAP_REGS_s0
sbi_trap.h, 289
SBI_TRAP_REGS_s1
sbi_trap.h, 289
SBI_TRAP_REGS_s10
sbi_trap.h, 289
SBI_TRAP_REGS_s11
sbi_trap.h, 289
SBI_TRAP_REGS_s2
sbi_trap.h, 289
SBI_TRAP_REGS_s3
sbi_trap.h, 289
SBI_TRAP_REGS_s4
sbi_trap.h, 290
SBI_TRAP_REGS_s5
sbi_trap.h, 290
SBI_TRAP_REGS_s6
sbi_trap.h, 290
SBI_TRAP_REGS_s7
sbi_trap.h, 290
SBI_TRAP_REGS_s8
sbi_trap.h, 290
SBI_TRAP_REGS_s9
sbi_trap.h, 290
SBI_TRAP_REGS_SIZE
sbi_trap.h, 290
SBI_TRAP_REGS_sp
sbi_trap.h, 290
SBI_TRAP_REGS_t0
sbi_trap.h, 290
SBI_TRAP_REGS_t1
sbi_trap.h, 290
SBI_TRAP_REGS_t2
sbi_trap.h, 290
SBI_TRAP_REGS_t3
sbi_trap.h, 290
SBI_TRAP_REGS_t4
sbi_trap.h, 291
SBI_TRAP_REGS_t5
sbi_trap.h, 291
SBI_TRAP_REGS_t6
sbi_trap.h, 291
SBI_TRAP_REGS_tp

sbi_trap.h, 291
SBI_TRAP_REGS_zero
sbi_trap.h, 291
sbi_types.h
__noreturn, 294
__packed, 294
array_size, 294
bool, 296
CLAMP, 294
container_of, 294
FALSE, 294
false, 294
int16_t, 296
int32_t, 296
likely, 294
MAX, 294
MIN, 294
NULL, 295
offsetof, 295
physical_addr_t, 296
physical_size_t, 296
ROUNDDOWN, 295
ROUNDUP, 295
s16, 296
s32, 296
s8, 296
size_t, 296
ssize_t, 296
STR, 295
TRUE, 295
true, 295
u16, 296
u32, 296
u8, 297
uint16_t, 297
uint32_t, 297
uint8_t, 297
uintptr_t, 297
ulong, 297
unlikely, 295
virtual_addr_t, 297
virtual_size_t, 297
XSTR, 295
sbi_unpriv.c
DEFINE_UNPRIVILEGED_LOAD_FUNCTION, 450
DEFINE_UNPRIVILEGED_STORE_FUNCTION, 451
sbi_get_insn, 451
sbi_load_u64, 452
sbi_store_u64, 452
sbi_unpriv.h
DECLARE_UNPRIVILEGED_LOAD_FUNCTION, 298
DECLARE_UNPRIVILEGED_STORE_FUNCTION, 298
sbi_get_insn, 299
sbi_version.h
OPENSBI_VERSION, 299
OPENSBI_VERSION_MAJOR, 300
OPENSBI_VERSION_MINOR, 300
send
fdt_ipi, 56
serial_drivers
fdt_serial.c, 654
serial_htif_match
fdt_serial_htif.c, 655
serial_shakti_init
fdt_serial_shakti.c, 656
serial_shakti_match
fdt_serial_shakti.c, 657
serial_sifive_init
fdt_serial_sifive.c, 657
serial_sifive_match
fdt_serial_sifive.c, 658
serial_uart8250_init
fdt_serial_uart8250.c, 659
serial_uart8250_match
fdt_serial_uart8250.c, 659
SET_RD
riscv_encoding.h, 146
set_reg
sifive-uart.c, 664
uart8250.c, 670
SH_RD
riscv_encoding.h, 146
SH_RS1
riscv_encoding.h, 147
SH_RS2
riscv_encoding.h, 147
SH_RS2C
riscv_encoding.h, 147
shakti-uart.c
REG_BAUD, 660
REG_CONTROL, 660
REG_DELAY, 660
REG_INT_EN, 661
REG_IQ_CYCLES, 661
REG_RX, 661
REG_RX_THRES, 661
REG_STATUS, 661
REG_TX, 661
shakti_uart_getc, 661
shakti_uart_init, 661
shakti_uart_putc, 662
uart_base, 662
shakti-uart.h
shakti_uart_getc, 321
shakti_uart_init, 321
shakti_uart_putc, 322
shakti_uart_getc
shakti-uart.c, 661
shakti-uart.h, 321
shakti_uart_init
shakti-uart.c, 661
shakti-uart.h, 321

shakti_uart_putc
shakti-uart.c, 662
shakti-uart.h, 322
SHIFT_RIGHT
riscv_encoding.h, 147
sifive-uart.c
get_reg, 664
set_reg, 664
sifive_uart_getc, 665
sifive_uart_init, 665
sifive_uart_putc, 666
uart_base, 666
uart_baudrate, 667
uart_in_freq, 667
uart_min_clk_divisor, 666
UART_REG_DIV, 663
UART_REG_IE, 663
UART_REG_IP, 663
UART_REG_RXCTRL, 663
UART_REG_RXFIFO, 663
UART_REG_TXCTRL, 663
UART_REG_TXFIFO, 663
UART_RXCTRL_RXEN, 664
UART_RXFIFO_DATA, 664
UART_RXFIFO_EMPTY, 664
UART_TXCTRL_TXEN, 664
UART_TXFIFO_FULL, 664
sifive-uart.h
sifive_uart_getc, 323
sifive_uart_init, 323
sifive_uart_putc, 324
sifive_test.c
FINISHER_FAIL, 684
FINISHER_PASS, 684
FINISHER_RESET, 684
sifive_test_base, 685
sifive_test_init, 684
sifive_test_system_reset, 685
sifive_test.h
sifive_test_init, 332
sifive_test_system_reset, 333
sifive_test_base
sifive_test.c, 685
sifive_test_init
sifive_test.c, 684
sifive_test.h, 332
sifive_test_reset_init
fdt_reset_sifive.c, 651
sifive_test_reset_match
fdt_reset_sifive.c, 652
sifive_test_system_reset
sifive_test.c, 685
sifive_test.h, 333
sifive_uart_getc
sifive-uart.c, 665
sifive-uart.h, 323
sifive_uart_init
sifive-uart.c, 665
sifive-uart.h, 323
sifive_uart_putc
sifive-uart.c, 666
sifive-uart.h, 324
SIP_SSIP
riscv_encoding.h, 147
SIP_STIP
riscv_encoding.h, 147
size
fdt_reserve_entry, 59
sbi_tlb_info, 73
size_dt_strings
fdt_header, 55
size_dt_struct
fdt_header, 55
size_t
sbi_types.h, 296
small_const_nbits
sbi_bitmap.h, 160
smask
sbi_tlb_info, 73
smp_mb
riscv_barrier.h, 97
smp_rmb
riscv_barrier.h, 97
smp_wmb
riscv_barrier.h, 98
sp
sbi_trap_regs, 77
spin_lock
riscv_locks.c, 345
riscv_locks.h, 155
spin_lock_check
riscv_locks.c, 346
riscv_locks.h, 156
SPIN_LOCK_INIT
riscv_locks.h, 155
SPIN_LOCK_INITIALIZER
riscv_locks.h, 155
spin_trylock
riscv_locks.c, 347
riscv_locks.h, 157
spin_unlock
riscv_locks.c, 347
riscv_locks.h, 157
spinlock_t, 78
lock, 78
ssize_t
sbi_types.h, 296
SSTATUS32_SD
riscv_encoding.h, 147
SSTATUS64_SD
riscv_encoding.h, 147
SSTATUS64_UXL
riscv_encoding.h, 147
SSTATUS_FS
riscv_encoding.h, 147
SSTATUS_MXR

riscv_encoding.h, 147
SSTATUS_SD
riscv_encoding.h, 147
SSTATUS_SIE
riscv_encoding.h, 148
SSTATUS_SPIE
riscv_encoding.h, 148
SSTATUS_SPIE_SHIFT
riscv_encoding.h, 148
SSTATUS_SPP
riscv_encoding.h, 148
SSTATUS_SPP_SHIFT
riscv_encoding.h, 148
SSTATUS_SUM
riscv_encoding.h, 148
SSTATUS_VS
riscv_encoding.h, 148
SSTATUS_XS
riscv_encoding.h, 148
start
sbi_tlb_info, 73
STR
sbi_types.h, 295
strchr
libfdt_env.h, 637
strcmp
libfdt_env.h, 637
strcpy
libfdt_env.h, 637
STRINGIFY
sbi_const.h, 176
strlen
libfdt_env.h, 637
strnlen
libfdt_env.h, 637
strrchr
libfdt_env.h, 637
sw_flags
fdt_sw.c, 564
sync
sbi_ipi_event_ops, 67
system_opcode_insn
sbi_illegal_insn.c, 401
system_reset
fdt_reset, 60
sbi_platform_operations, 71
SZREG
riscv_asm.h, 88

t0
sbi_trap_regs, 77
t1
sbi_trap_regs, 78
t2
sbi_trap_regs, 78
t3
sbi_trap_regs, 78
t4
sbi_trap_regs, 78
t5
sbi_trap_regs, 78
t6
sbi_trap_regs, 78
tag
fdt_node_header, 58
fdt_property, 59
tail
sbi_fifo, 65
time_cmp
clint_data, 54
time_delta
clint_data, 54
time_delta_computed
clint_data, 54
time_delta_off
sbi_timer.c, 436
time_delta_reference
clint_data, 54
time_rd
clint_data, 54
time_val
clint_data, 54
time_wr
clint_data, 54
timer_clint_cold_init
fdt_timer_clint.c, 690
timer_clint_match
fdt_timer_clint.c, 690
timer_drivers
fdt_timer.c, 689
timer_event_start
sbi_platform_operations, 71
timer_event_stop
sbi_platform_operations, 71
timer_exit
sbi_platform_operations, 71
timer_init
sbi_platform_operations, 71
timer_value
sbi_platform_operations, 71
tinst
sbi_trap_info, 74
tlb_event
sbi_tlb.c, 446
tlb_fifo_mem_off
sbi_tlb.c, 446
tlb_fifo_off
sbi_tlb.c, 447
tlb_ops
sbi_tlb.c, 447
tlb_range_flush_limit
sbi_tlb.c, 447
tlb_sync_off
sbi_tlb.c, 447
tmp0
sbi_scratch, 73
TOHOST_CMD

htif.c, 681
totalsize
fdt_header, 55
tp
sbi_trap_regs, 78
TRUE
sbi_types.h, 295
true
sbi_types.h, 295
truly_illegal_insn
sbi_illegal_insn.c, 402
tval
sbi_trap_info, 74
tval2
sbi_trap_info, 74
type
sbi_tlb_info, 73

u16
sbi_types.h, 296
u32
sbi_types.h, 296
u8
sbi_types.h, 297
uart8250.c
get_reg, 670
set_reg, 670
uart8250_base, 672
uart8250_baudrate, 672
uart8250_getc, 671
uart8250_in_freq, 672
uart8250_init, 671
uart8250_putc, 672
uart8250_reg_shift, 672
uart8250_reg_width, 672
UART_DLL_OFFSET, 668
UART_DLM_OFFSET, 668
UART_FCR_OFFSET, 668
UART_IER_OFFSET, 668
UART_IIR_OFFSET, 668
UART_LCR_OFFSET, 668
UART_LSR_BI, 668
UART_LSR_BRK_ERROR_BITS, 669
UART_LSR_DR, 669
UART_LSR_FE, 669
UART_LSR_FIFOE, 669
UART_LSR_OE, 669
UART_LSR_OFFSET, 669
UART_LSR_PE, 669
UART_LSR_TEMT, 669
UART_LSR_THRE, 669
UART_MCR_OFFSET, 669
UART_MDR1_OFFSET, 669
UART_MSR_OFFSET, 669
UART_RBR_OFFSET, 670
UART_SCR_OFFSET, 670
UART_THR_OFFSET, 670
uart8250.h
uart8250_getc, 325
uart8250_init, 325
uart8250_putc, 326
uart8250_base
uart8250.c, 672
uart8250_baudrate
uart8250.c, 672
uart8250_getc
uart8250.c, 671
uart8250.h, 325
uart8250_in_freq
uart8250.c, 672
uart8250_init
uart8250.c, 671
uart8250.h, 325
uart8250_putc
uart8250.c, 672
uart8250.h, 326
uart8250_reg_shift
uart8250.c, 672
uart8250_reg_width
uart8250.c, 672
uart_base
shakti-uart.c, 662
sifive-uart.c, 666
uart_baudrate
sifive-uart.c, 667
UART_DLL_OFFSET
uart8250.c, 668
UART_DLM_OFFSET
uart8250.c, 668
UART_FCR_OFFSET
uart8250.c, 668
UART_IER_OFFSET
uart8250.c, 668
UART_IIR_OFFSET
uart8250.c, 668
uart_in_freq
sifive-uart.c, 667
UART_LCR_OFFSET
uart8250.c, 668
UART_LSR_BI
uart8250.c, 668
UART_LSR_BRK_ERROR_BITS
uart8250.c, 669
UART_LSR_DR
uart8250.c, 669
UART_LSR_FE
uart8250.c, 669
UART_LSR_FIFOE
uart8250.c, 669
UART_LSR_OE
uart8250.c, 669
UART_LSR_OFFSET
uart8250.c, 669
UART_LSR_PE
uart8250.c, 669
UART_LSR_TEMT
uart8250.c, 669

UART_LSR_THRE
uart8250.c, 669
UART_MCR_OFFSET
uart8250.c, 669
UART_MDR1_OFFSET
uart8250.c, 669
uart_min_clk_divisor
sifive-uart.c, 666
UART_MSR_OFFSET
uart8250.c, 669
UART_RBR_OFFSET
uart8250.c, 670
UART_REG_DIV
sifive-uart.c, 663
UART_REG_IE
sifive-uart.c, 663
UART_REG_IP
sifive-uart.c, 663
UART_REG_RXCTRL
sifive-uart.c, 663
UART_REG_RXFIFO
sifive-uart.c, 663
UART_REG_TXCTRL
sifive-uart.c, 663
UART_REG_TXFIFO
sifive-uart.c, 663
UART_RXCTRL_RXEN
sifive-uart.c, 664
UART_RXFIFO_DATA
sifive-uart.c, 664
UART_RXFIFO_EMPTY
sifive-uart.c, 664
UART_SCR_OFFSET
uart8250.c, 670
UART_THR_OFFSET
uart8250.c, 670
UART_TXCTRL_TXEN
sifive-uart.c, 664
UART_TXFIFO_FULL
sifive-uart.c, 664
uint16_t
sbi_types.h, 297
UINT32_MAX
libfdt_env.h, 638
uint32_t
sbi_types.h, 297
uint8_t
sbi_types.h, 297
UINT_MAX
libfdt_env.h, 638
uintptr_t
sbi_types.h, 297
UL
sbi_const.h, 176
ULL
sbi_const.h, 177
ulong
sbi_types.h, 297
unlikely
sbi_types.h, 295
update
sbi_ipi_event_ops, 67

va_arg
sbi_console.c, 353
va_end
sbi_console.c, 353
va_list
sbi_console.c, 354
va_start
sbi_console.c, 354
value
fdt_timer, 61
vendor_ext_check
sbi_platform_operations, 71
vendor_ext_provider
sbi_platform_operations, 71
version
fdt_header, 56
fw_dynamic_info, 62
virtual_addr_t
sbi_types.h, 297
virtual_size_t
sbi_types.h, 297
vmid
sbi_tlb_info, 73

wait_for_coldboot
sbi_init.c, 410
wake_coldboot_harts
sbi_init.c, 410
warm_init
fdt_ipi, 56
fdt_irqchip, 57
fdt_timer, 61
warmboot_addr
sbi_scratch, 73
wfi
riscv_asm.h, 88
wmb
riscv_barrier.h, 98
writeb
riscv_io.h, 153
writeb_relaxed
riscv_io.h, 153
writel
riscv_io.h, 153
writel_relaxed
riscv_io.h, 153
writeq
riscv_io.h, 153
writeq_relaxed
riscv_io.h, 153
writew
riscv_io.h, 153
writew_relaxed
riscv_io.h, 153

xchg
riscv_atomic.c, 341
XSTR
sbi_types.h, 295

zero
sbi_trap_regs, 78