

RISC-V OpenSBI

v0.6

Generated by Doxygen 1.8.13

Contents

1	README	1
2	OpenSBI Contribution Guideline	5
3	OpenSBI Platform Support Guideline	7
4	OpenSBI Library Usage	9
5	OpenSBI Platform Firmwares	11
6	OpenSBI Firmware with Dynamic Information *FW_DYNAMIC*	13
7	OpenSBI Firmware with Jump Address *FW_JUMP*	15
8	OpenSBI Firmware with Payload *FW_PAYLOAD*	17
9	Linux as a direct payload to OpenSBI	19
10	U-Boot as a payload to OpenSBI	21
11	Andes AE350 SoC Platform	23
12	Ariane FPGA SoC Platform	25
13	OpenSBI Supported Platforms	27
14	QEMU RISC-V Virt Machine Platform	29
15	SiFive FU540 SoC Platform	33
16	Spike Simulator Platform	37

17 T-HEAD C910 Processor	39
18 Data Structure Index	41
18.1 Data Structures	41
19 File Index	43
19.1 File List	43
20 Data Structure Documentation	45
20.1 atomic_t Struct Reference	45
20.1.1 Field Documentation	45
20.1.1.1 counter	45
20.2 fdt_header Struct Reference	45
20.2.1 Field Documentation	46
20.2.1.1 boot_cpuid_phys	46
20.2.1.2 last_comp_version	46
20.2.1.3 magic	46
20.2.1.4 off_dt_strings	46
20.2.1.5 off_dt_struct	46
20.2.1.6 off_mem_rsvmap	47
20.2.1.7 size_dt_strings	47
20.2.1.8 size_dt_struct	47
20.2.1.9 totalsize	47
20.2.1.10 version	47
20.3 fdt_node_header Struct Reference	47
20.3.1 Field Documentation	47
20.3.1.1 name	48
20.3.1.2 tag	48
20.4 fdt_property Struct Reference	48
20.4.1 Field Documentation	48
20.4.1.1 data	48
20.4.1.2 len	48

20.4.1.3	nameoff	49
20.4.1.4	tag	49
20.5	fdt_reserve_entry Struct Reference	49
20.5.1	Field Documentation	49
20.5.1.1	address	49
20.5.1.2	size	49
20.6	fw_dynamic_info Struct Reference	49
20.6.1	Detailed Description	50
20.6.2	Field Documentation	50
20.6.2.1	boot_hart	50
20.6.2.2	magic	50
20.6.2.3	next_addr	50
20.6.2.4	next_mode	50
20.6.2.5	options	51
20.6.2.6	version	51
20.7	sbi_dlist Struct Reference	51
20.7.1	Field Documentation	51
20.7.1.1	next	51
20.7.1.2	prev	51
20.8	sbi_ecall_extension Struct Reference	52
20.8.1	Field Documentation	52
20.8.1.1	extid_end	52
20.8.1.2	extid_start	52
20.8.1.3	handle	52
20.8.1.4	head	52
20.8.1.5	probe	53
20.9	sbi_fifo Struct Reference	53
20.9.1	Field Documentation	53
20.9.1.1	avail	53
20.9.1.2	entry_size	53

20.9.1.3	num_entries	53
20.9.1.4	qlock	54
20.9.1.5	queue	54
20.9.1.6	tail	54
20.10	sbi_ipi_event_ops Struct Reference	54
20.10.1	Detailed Description	54
20.10.2	Field Documentation	54
20.10.2.1	name	54
20.10.2.2	process	55
20.10.2.3	sync	55
20.10.2.4	update	55
20.11	sbi_platform Struct Reference	55
20.11.1	Detailed Description	55
20.11.2	Field Documentation	56
20.11.2.1	disabled_hart_mask	56
20.11.2.2	features	56
20.11.2.3	firmware_context	56
20.11.2.4	hart_count	56
20.11.2.5	hart_stack_size	56
20.11.2.6	name	56
20.11.2.7	opensbi_version	56
20.11.2.8	platform_ops_addr	57
20.11.2.9	platform_version	57
20.12	sbi_platform_operations Struct Reference	57
20.12.1	Detailed Description	58
20.12.2	Field Documentation	58
20.12.2.1	console_getc	58
20.12.2.2	console_init	58
20.12.2.3	console_putc	58
20.12.2.4	early_exit	58

20.12.2.5 early_init	58
20.12.2.6 final_exit	58
20.12.2.7 final_init	59
20.12.2.8 get_tlbr_flush_limit	59
20.12.2.9 ipi_clear	59
20.12.2.10 ipi_exit	59
20.12.2.11 ipi_init	59
20.12.2.12 ipi_send	59
20.12.2.13 irqchip_exit	59
20.12.2.14 irqchip_init	59
20.12.2.15 misa_check_extension	60
20.12.2.16 misa_get_xlen	60
20.12.2.17 pmp_region_count	60
20.12.2.18 pmp_region_info	60
20.12.2.19 system_reboot	60
20.12.2.20 system_shutdown	60
20.12.2.21 timer_event_start	60
20.12.2.22 timer_event_stop	61
20.12.2.23 timer_exit	61
20.12.2.24 timer_init	61
20.12.2.25 timer_value	61
20.12.2.26 vendor_ext_check	61
20.12.2.27 vendor_ext_provider	61
20.13 sbi_scratch Struct Reference	61
20.13.1 Detailed Description	62
20.13.2 Field Documentation	62
20.13.2.1 fw_size	62
20.13.2.2 fw_start	62
20.13.2.3 hartid_to_scratch	62
20.13.2.4 next_addr	62

20.13.2.5 next_arg1	63
20.13.2.6 next_mode	63
20.13.2.7 options	63
20.13.2.8 platform_addr	63
20.13.2.9 tmp0	63
20.13.2.10 warmboot_addr	63
20.14 sbi_tlb_info Struct Reference	63
20.14.1 Field Documentation	64
20.14.1.1 asid	64
20.14.1.2 shart_mask	64
20.14.1.3 size	64
20.14.1.4 start	64
20.14.1.5 type	64
20.15 sbi_trap_info Struct Reference	65
20.15.1 Detailed Description	65
20.15.2 Field Documentation	65
20.15.2.1 cause	65
20.15.2.2 epc	65
20.15.2.3 tinst	65
20.15.2.4 tval	65
20.15.2.5 tval2	66
20.16 sbi_trap_regs Struct Reference	66
20.16.1 Detailed Description	67
20.16.2 Field Documentation	67
20.16.2.1 a0	67
20.16.2.2 a1	67
20.16.2.3 a2	67
20.16.2.4 a3	67
20.16.2.5 a4	67
20.16.2.6 a5	67

20.16.2.7 a6	68
20.16.2.8 a7	68
20.16.2.9 gp	68
20.16.2.10 mepc	68
20.16.2.11 mstatus	68
20.16.2.12 mstatusH	68
20.16.2.13 ra	68
20.16.2.14 s0	68
20.16.2.15 s1	69
20.16.2.16 s10	69
20.16.2.17 s11	69
20.16.2.18 s2	69
20.16.2.19 s3	69
20.16.2.20 s4	69
20.16.2.21 s5	69
20.16.2.22 s6	69
20.16.2.23 s7	70
20.16.2.24 s8	70
20.16.2.25 s9	70
20.16.2.26 sp	70
20.16.2.27 0	70
20.16.2.28 1	70
20.16.2.29 2	70
20.16.2.30 3	70
20.16.2.31 t4	71
20.16.2.32 5	71
20.16.2.33 6	71
20.16.2.34 tp	71
20.16.2.35 zero	71
20.17 spinlock_t Struct Reference	71
20.17.1 Field Documentation	71
20.17.1.1 lock	71

21 File Documentation	73
21.1 docs/contributing.md File Reference	73
21.2 docs/firmware/fw.md File Reference	73
21.3 docs/firmware/fw_dynamic.md File Reference	73
21.4 docs/firmware/fw_jump.md File Reference	73
21.5 docs/firmware/fw_payload.md File Reference	73
21.6 docs/firmware/payload_linux.md File Reference	73
21.7 docs/firmware/payload_uboot.md File Reference	73
21.8 docs/library_usage.md File Reference	73
21.9 docs/platform/andes-ae350.md File Reference	73
21.10 docs/platform/ariane-fpga.md File Reference	73
21.11 docs/platform/platform.md File Reference	73
21.12 docs/platform/qemu_virt.md File Reference	73
21.13 docs/platform/sifive_fu540.md File Reference	74
21.14 docs/platform/spike.md File Reference	74
21.15 docs/platform/thead-c910.md File Reference	74
21.16 docs/platform_guide.md File Reference	74
21.17 include/sbi/fw_dynamic.h File Reference	74
21.17.1 Macro Definition Documentation	75
21.17.1.1 FW_DYNAMIC_INFO_BOOT_HART_OFFSET	75
21.17.1.2 FW_DYNAMIC_INFO_MAGIC_OFFSET	75
21.17.1.3 FW_DYNAMIC_INFO_MAGIC_VALUE	75
21.17.1.4 FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET	75
21.17.1.5 FW_DYNAMIC_INFO_NEXT_MODE_M	75
21.17.1.6 FW_DYNAMIC_INFO_NEXT_MODE_OFFSET	75
21.17.1.7 FW_DYNAMIC_INFO_NEXT_MODE_S	76
21.17.1.8 FW_DYNAMIC_INFO_NEXT_MODE_U	76
21.17.1.9 FW_DYNAMIC_INFO_OPTIONS_OFFSET	76
21.17.1.10 FW_DYNAMIC_INFO_VERSION_MAX	76
21.17.1.11 FW_DYNAMIC_INFO_VERSION_OFFSET	76

21.17.2 Variable Documentation	76
21.17.2.1 <code>__packed</code>	76
21.18 <code>include/sbi/riscv_asm.h</code> File Reference	77
21.18.1 Macro Definition Documentation	78
21.18.1.1 <code>__ASM_STR</code>	78
21.18.1.2 <code>csr_clear</code>	78
21.18.1.3 <code>csr_read</code>	78
21.18.1.4 <code>csr_read_clear</code>	79
21.18.1.5 <code>csr_read_set</code>	79
21.18.1.6 <code>csr_set</code>	79
21.18.1.7 <code>csr_swap</code>	80
21.18.1.8 <code>csr_write</code>	80
21.18.1.9 <code>LGREG</code>	80
21.18.1.10 <code>misa_extension</code>	80
21.18.1.11 <code>PAGE_MASK</code>	81
21.18.1.12 <code>PAGE_SHIFT</code>	81
21.18.1.13 <code>PAGE_SIZE</code>	81
21.18.1.14 <code>REG_L</code>	81
21.18.1.15 <code>REG_S</code>	81
21.18.1.16 <code>SZREG</code>	81
21.18.1.17 <code>wfi</code>	81
21.18.2 Function Documentation	82
21.18.2.1 <code>csr_read_num()</code>	82
21.18.2.2 <code>csr_write_num()</code>	82
21.18.2.3 <code>misa_extension_imp()</code>	82
21.18.2.4 <code>misa_string()</code>	83
21.18.2.5 <code>misa_xlen()</code>	84
21.18.2.6 <code>pmp_get()</code>	84
21.18.2.7 <code>pmp_set()</code>	85
21.19 <code>include/sbi/riscv_atomic.h</code> File Reference	86

21.19.1 Macro Definition Documentation	86
21.19.1.1 ATOMIC_INIT	86
21.19.1.2 ATOMIC_INITIALIZER	87
21.19.2 Function Documentation	87
21.19.2.1 arch_atomic_cmpxchg()	87
21.19.2.2 arch_atomic_xchg()	87
21.19.2.3 atomic_add_return()	87
21.19.2.4 atomic_clear_bit()	88
21.19.2.5 atomic_raw_clear_bit()	88
21.19.2.6 atomic_raw_set_bit()	88
21.19.2.7 atomic_raw_xchg_uint()	89
21.19.2.8 atomic_raw_xchg_ulong()	89
21.19.2.9 atomic_read()	89
21.19.2.10atomic_set_bit()	90
21.19.2.11atomic_sub_return()	90
21.19.2.12atomic_write()	90
21.20include/sbi/riscv_barrier.h File Reference	90
21.20.1 Macro Definition Documentation	91
21.20.1.1 __smp_load_acquire	91
21.20.1.2 __smp_store_release	91
21.20.1.3 cpu_relax	92
21.20.1.4 mb	92
21.20.1.5 RISCV_ACQUIRE_BARRIER	92
21.20.1.6 RISCV_FENCE	92
21.20.1.7 RISCV_RELEASE_BARRIER	92
21.20.1.8 rmb	92
21.20.1.9 smp_mb	92
21.20.1.10smp_rmb	93
21.20.1.11smp_wmb	93
21.20.1.12wmb	93

21.21 include/sbi/riscv_encoding.h File Reference	93
21.21.1 Macro Definition Documentation	102
21.21.1.1 CAUSE_BREAKPOINT	102
21.21.1.2 CAUSE_FETCH_ACCESS	102
21.21.1.3 CAUSE_FETCH_GUEST_PAGE_FAULT	103
21.21.1.4 CAUSE_FETCH_PAGE_FAULT	103
21.21.1.5 CAUSE_HYPERVISOR_ECALL	103
21.21.1.6 CAUSE_ILLEGAL_INSTRUCTION	103
21.21.1.7 CAUSE_LOAD_ACCESS	103
21.21.1.8 CAUSE_LOAD_GUEST_PAGE_FAULT	103
21.21.1.9 CAUSE_LOAD_PAGE_FAULT	103
21.21.1.10 CAUSE_MACHINE_ECALL	103
21.21.1.11 CAUSE_MISALIGNED_FETCH	104
21.21.1.12 CAUSE_MISALIGNED_LOAD	104
21.21.1.13 CAUSE_MISALIGNED_STORE	104
21.21.1.14 CAUSE_STORE_ACCESS	104
21.21.1.15 CAUSE_STORE_GUEST_PAGE_FAULT	104
21.21.1.16 CAUSE_STORE_PAGE_FAULT	104
21.21.1.17 CAUSE_SUPERVISOR_ECALL	104
21.21.1.18 CAUSE_USER_ECALL	104
21.21.1.19 CSR_CYCLE	105
21.21.1.20 CSR_CYCLEH	105
21.21.1.21 CSR_DCSR	105
21.21.1.22 CSR_DPC	105
21.21.1.23 CSR_DSCRATCH	105
21.21.1.24 CSR_FCSR	105
21.21.1.25 CSR_FFLAGS	105
21.21.1.26 CSR_FRM	105
21.21.1.27 CSR_HCOUNTERNEN	106
21.21.1.28 CSR_HEDELEG	106

21.21.1.29	CSR_HGATP	106
21.21.1.30	CSR_HGEIE	106
21.21.1.31	CSR_HGEIP	106
21.21.1.32	CSR_HIDELEG	106
21.21.1.33	CSR_HIE	106
21.21.1.34	CSR_HIP	106
21.21.1.35	CSR_HPMCOUNTER10	107
21.21.1.36	CSR_HPMCOUNTER10H	107
21.21.1.37	CSR_HPMCOUNTER11	107
21.21.1.38	CSR_HPMCOUNTER11H	107
21.21.1.39	CSR_HPMCOUNTER12	107
21.21.1.40	CSR_HPMCOUNTER12H	107
21.21.1.41	CSR_HPMCOUNTER13	107
21.21.1.42	CSR_HPMCOUNTER13H	107
21.21.1.43	CSR_HPMCOUNTER14	108
21.21.1.44	CSR_HPMCOUNTER14H	108
21.21.1.45	CSR_HPMCOUNTER15	108
21.21.1.46	CSR_HPMCOUNTER15H	108
21.21.1.47	CSR_HPMCOUNTER16	108
21.21.1.48	CSR_HPMCOUNTER16H	108
21.21.1.49	CSR_HPMCOUNTER17	108
21.21.1.50	CSR_HPMCOUNTER17H	108
21.21.1.51	CSR_HPMCOUNTER18	109
21.21.1.52	CSR_HPMCOUNTER18H	109
21.21.1.53	CSR_HPMCOUNTER19	109
21.21.1.54	CSR_HPMCOUNTER19H	109
21.21.1.55	CSR_HPMCOUNTER20	109
21.21.1.56	CSR_HPMCOUNTER20H	109
21.21.1.57	CSR_HPMCOUNTER21	109
21.21.1.58	CSR_HPMCOUNTER21H	109

21.21.1.59CSR_HPMCounter22	110
21.21.1.60CSR_HPMCounter22H	110
21.21.1.61CSR_HPMCounter23	110
21.21.1.62CSR_HPMCounter23H	110
21.21.1.63CSR_HPMCounter24	110
21.21.1.64CSR_HPMCounter24H	110
21.21.1.65CSR_HPMCounter25	110
21.21.1.66CSR_HPMCounter25H	110
21.21.1.67CSR_HPMCounter26	111
21.21.1.68CSR_HPMCounter26H	111
21.21.1.69CSR_HPMCounter27	111
21.21.1.70CSR_HPMCounter27H	111
21.21.1.71CSR_HPMCounter28	111
21.21.1.72CSR_HPMCounter28H	111
21.21.1.73CSR_HPMCounter29	111
21.21.1.74CSR_HPMCounter29H	111
21.21.1.75CSR_HPMCounter3	112
21.21.1.76CSR_HPMCounter30	112
21.21.1.77CSR_HPMCounter30H	112
21.21.1.78CSR_HPMCounter31	112
21.21.1.79CSR_HPMCounter31H	112
21.21.1.80CSR_HPMCounter3H	112
21.21.1.81CSR_HPMCounter4	112
21.21.1.82CSR_HPMCounter4H	112
21.21.1.83CSR_HPMCounter5	113
21.21.1.84CSR_HPMCounter5H	113
21.21.1.85CSR_HPMCounter6	113
21.21.1.86CSR_HPMCounter6H	113
21.21.1.87CSR_HPMCounter7	113
21.21.1.88CSR_HPMCounter7H	113

21.21.1.89	CSR_HPMCOUNTER8	113
21.21.1.90	CSR_HPMCOUNTER8H	113
21.21.1.91	CSR_HPMCOUNTER9	114
21.21.1.92	CSR_HPMCOUNTER9H	114
21.21.1.93	CSR_HSTATUS	114
21.21.1.94	CSR_HTIMEDELTA	114
21.21.1.95	CSR_HTIMEDELTAH	114
21.21.1.96	CSR_HTINST	114
21.21.1.97	CSR_HTVAL	114
21.21.1.98	CSR_INSTRET	114
21.21.1.99	CSR_INSTRETH	115
21.21.1.100	SR_MARCHID	115
21.21.1.101	SR_MCAUSE	115
21.21.1.102	SR_MCOUNTEREN	115
21.21.1.103	SR_MCYCLE	115
21.21.1.104	SR_MCYCLEH	115
21.21.1.105	SR_MEDELEG	115
21.21.1.106	SR_MEPC	115
21.21.1.107	SR_MHARTID	116
21.21.1.108	SR_MHPMCOUNTER10	116
21.21.1.109	SR_MHPMCOUNTER10H	116
21.21.1.110	SR_MHPMCOUNTER11	116
21.21.1.111	SR_MHPMCOUNTER11H	116
21.21.1.112	SR_MHPMCOUNTER12	116
21.21.1.113	SR_MHPMCOUNTER12H	116
21.21.1.114	SR_MHPMCOUNTER13	116
21.21.1.115	SR_MHPMCOUNTER13H	117
21.21.1.116	SR_MHPMCOUNTER14	117
21.21.1.117	SR_MHPMCOUNTER14H	117
21.21.1.118	SR_MHPMCOUNTER15	117

21.21.1.110SR_MHPMCOUNTER15H	117
21.21.1.120SR_MHPMCOUNTER16	117
21.21.1.120SR_MHPMCOUNTER16H	117
21.21.1.122SR_MHPMCOUNTER17	117
21.21.1.123SR_MHPMCOUNTER17H	118
21.21.1.124SR_MHPMCOUNTER18	118
21.21.1.125SR_MHPMCOUNTER18H	118
21.21.1.126SR_MHPMCOUNTER19	118
21.21.1.127SR_MHPMCOUNTER19H	118
21.21.1.128SR_MHPMCOUNTER20	118
21.21.1.129SR_MHPMCOUNTER20H	118
21.21.1.130SR_MHPMCOUNTER21	118
21.21.1.133SR_MHPMCOUNTER21H	119
21.21.1.132SR_MHPMCOUNTER22	119
21.21.1.133SR_MHPMCOUNTER22H	119
21.21.1.134SR_MHPMCOUNTER23	119
21.21.1.135SR_MHPMCOUNTER23H	119
21.21.1.136SR_MHPMCOUNTER24	119
21.21.1.137SR_MHPMCOUNTER24H	119
21.21.1.138SR_MHPMCOUNTER25	119
21.21.1.139SR_MHPMCOUNTER25H	120
21.21.1.140SR_MHPMCOUNTER26	120
21.21.1.140SR_MHPMCOUNTER26H	120
21.21.1.142SR_MHPMCOUNTER27	120
21.21.1.143SR_MHPMCOUNTER27H	120
21.21.1.144SR_MHPMCOUNTER28	120
21.21.1.145SR_MHPMCOUNTER28H	120
21.21.1.146SR_MHPMCOUNTER29	120
21.21.1.147SR_MHPMCOUNTER29H	121
21.21.1.148SR_MHPMCOUNTER3	121

21.21.1.149SR_MHPMCOUNTER30	121
21.21.1.150SR_MHPMCOUNTER30H	121
21.21.1.151SR_MHPMCOUNTER31	121
21.21.1.152SR_MHPMCOUNTER31H	121
21.21.1.153SR_MHPMCOUNTER3H	121
21.21.1.154SR_MHPMCOUNTER4	121
21.21.1.155SR_MHPMCOUNTER4H	122
21.21.1.156SR_MHPMCOUNTER5	122
21.21.1.157SR_MHPMCOUNTER5H	122
21.21.1.158SR_MHPMCOUNTER6	122
21.21.1.159SR_MHPMCOUNTER6H	122
21.21.1.160SR_MHPMCOUNTER7	122
21.21.1.161SR_MHPMCOUNTER7H	122
21.21.1.162SR_MHPMCOUNTER8	122
21.21.1.163SR_MHPMCOUNTER8H	123
21.21.1.164SR_MHPMCOUNTER9	123
21.21.1.165SR_MHPMCOUNTER9H	123
21.21.1.166SR_MHPMEVENT10	123
21.21.1.167SR_MHPMEVENT11	123
21.21.1.168SR_MHPMEVENT12	123
21.21.1.169SR_MHPMEVENT13	123
21.21.1.170SR_MHPMEVENT14	123
21.21.1.171SR_MHPMEVENT15	124
21.21.1.172SR_MHPMEVENT16	124
21.21.1.173SR_MHPMEVENT17	124
21.21.1.174SR_MHPMEVENT18	124
21.21.1.175SR_MHPMEVENT19	124
21.21.1.176SR_MHPMEVENT20	124
21.21.1.177SR_MHPMEVENT21	124
21.21.1.178SR_MHPMEVENT22	124

21.21.1.179SR_MHPMEVENT23	125
21.21.1.180SR_MHPMEVENT24	125
21.21.1.181SR_MHPMEVENT25	125
21.21.1.182SR_MHPMEVENT26	125
21.21.1.183SR_MHPMEVENT27	125
21.21.1.184SR_MHPMEVENT28	125
21.21.1.185SR_MHPMEVENT29	125
21.21.1.186SR_MHPMEVENT3	125
21.21.1.187SR_MHPMEVENT30	126
21.21.1.188SR_MHPMEVENT31	126
21.21.1.189SR_MHPMEVENT4	126
21.21.1.190SR_MHPMEVENT5	126
21.21.1.191SR_MHPMEVENT6	126
21.21.1.192SR_MHPMEVENT7	126
21.21.1.193SR_MHPMEVENT8	126
21.21.1.194SR_MHPMEVENT9	126
21.21.1.195SR_MIDELEG	127
21.21.1.196SR_MIE	127
21.21.1.197SR_MIMPID	127
21.21.1.198SR_MINSTRET	127
21.21.1.199SR_MINSTRETH	127
21.21.1.200SR_MIP	127
21.21.1.201SR_MISA	127
21.21.1.202SR_MSCRATCH	127
21.21.1.203SR_MSTATUS	128
21.21.1.204SR_MSTATUSH	128
21.21.1.205SR_MTINST	128
21.21.1.206SR_MTVAL	128
21.21.1.207SR_MTVAL2	128
21.21.1.208SR_MTVEC	128

21.21.1.209SR_MVENDORID	128
21.21.1.210SR_PMPADDR0	128
21.21.1.211SR_PMPADDR1	129
21.21.1.212SR_PMPADDR10	129
21.21.1.213SR_PMPADDR11	129
21.21.1.214SR_PMPADDR12	129
21.21.1.215SR_PMPADDR13	129
21.21.1.216SR_PMPADDR14	129
21.21.1.217SR_PMPADDR15	129
21.21.1.218SR_PMPADDR2	129
21.21.1.219SR_PMPADDR3	130
21.21.1.220SR_PMPADDR4	130
21.21.1.221SR_PMPADDR5	130
21.21.1.222SR_PMPADDR6	130
21.21.1.223SR_PMPADDR7	130
21.21.1.224SR_PMPADDR8	130
21.21.1.225SR_PMPADDR9	130
21.21.1.226SR_PMPCFG0	130
21.21.1.227SR_PMPCFG1	131
21.21.1.228SR_PMPCFG2	131
21.21.1.229SR_PMPCFG3	131
21.21.1.230SR_SATP	131
21.21.1.231SR_SCAUSE	131
21.21.1.232SR_SCOUNTEREN	131
21.21.1.233SR_SEPC	131
21.21.1.234SR_SIE	131
21.21.1.235SR_SIP	132
21.21.1.236SR_SSCRATCH	132
21.21.1.237SR_SSTATUS	132
21.21.1.238SR_STVAL	132

21.21.1.239	SR_STVEC	132
21.21.1.240	SR_TDATA1	132
21.21.1.241	SR_TDATA2	132
21.21.1.242	SR_TDATA3	132
21.21.1.243	SR_TIME	133
21.21.1.244	SR_TIMEH	133
21.21.1.245	SR_TSELECT	133
21.21.1.246	SR_UCAUSE	133
21.21.1.247	SR_UEPC	133
21.21.1.248	SR_UIE	133
21.21.1.249	SR_UIP	133
21.21.1.250	SR_USCRATCH	133
21.21.1.251	SR_USTATUS	134
21.21.1.252	SR_UTVAL	134
21.21.1.253	SR_UTVEC	134
21.21.1.254	SR_VSATP	134
21.21.1.255	SR_VSCAUSE	134
21.21.1.256	SR_VSEPC	134
21.21.1.257	SR_VSIE	134
21.21.1.258	SR_VSIP	134
21.21.1.259	SR_VSSCRATCH	135
21.21.1.260	SR_VSSTATUS	135
21.21.1.261	SR_VSTVAL	135
21.21.1.262	SR_VSTVEC	135
21.21.1.263	ET_RM	135
21.21.1.264	ET_RS1	135
21.21.1.265	ET_RS1S	135
21.21.1.266	ET_RS2	136
21.21.1.267	ET_RS2C	136
21.21.1.268	ET_RS2S	136

21.21.1.269	GET_SP	136
21.21.1.270	STATUS_SP2P	136
21.21.1.271	STATUS_SP2V	136
21.21.1.272	STATUS_SPRV	136
21.21.1.273	STATUS_SPV	137
21.21.1.274	STATUS_VTSR	137
21.21.1.275	STATUS_VTVM	137
21.21.1.276	MM_I	137
21.21.1.277	MM_S	137
21.21.1.278	NSN_16BIT_MASK	137
21.21.1.279	NSN_32BIT_MASK	137
21.21.1.280	NSN_IS_16BIT	138
21.21.1.281	NSN_IS_32BIT	138
21.21.1.282	NSN_LEN	138
21.21.1.283	NSN_MASK_C_FLD	138
21.21.1.284	NSN_MASK_C_FLDSP	138
21.21.1.285	NSN_MASK_C_FLW	138
21.21.1.286	NSN_MASK_C_FLWSP	139
21.21.1.287	NSN_MASK_C_FSD	139
21.21.1.288	NSN_MASK_C_FSDSP	139
21.21.1.289	NSN_MASK_C_FSW	139
21.21.1.290	NSN_MASK_C_FSWSP	139
21.21.1.291	NSN_MASK_C_LD	139
21.21.1.292	NSN_MASK_C_LDSP	139
21.21.1.293	NSN_MASK_C_LW	139
21.21.1.294	NSN_MASK_C_LWSP	140
21.21.1.295	NSN_MASK_C_SD	140
21.21.1.296	NSN_MASK_C_SDSP	140
21.21.1.297	NSN_MASK_C_SW	140
21.21.1.298	NSN_MASK_C_SWSP	140

21.21.1.299	INSN_MASK_FLD	140
21.21.1.300	INSN_MASK_FLQ	140
21.21.1.301	INSN_MASK_FLW	140
21.21.1.302	INSN_MASK_FSD	141
21.21.1.303	INSN_MASK_FSQ	141
21.21.1.304	INSN_MASK_FSW	141
21.21.1.305	INSN_MASK_LB	141
21.21.1.306	INSN_MASK_LBU	141
21.21.1.307	INSN_MASK_LD	141
21.21.1.308	INSN_MASK_LH	141
21.21.1.309	INSN_MASK_LHU	141
21.21.1.310	INSN_MASK_LW	142
21.21.1.311	INSN_MASK_LWU	142
21.21.1.312	INSN_MASK_SB	142
21.21.1.313	INSN_MASK_SD	142
21.21.1.314	INSN_MASK_SH	142
21.21.1.315	INSN_MASK_SW	142
21.21.1.316	INSN_MASK_WFI	142
21.21.1.317	INSN_MATCH_C_FLD	142
21.21.1.318	INSN_MATCH_C_FLDSP	143
21.21.1.319	INSN_MATCH_C_FLW	143
21.21.1.320	INSN_MATCH_C_FLWSP	143
21.21.1.321	INSN_MATCH_C_FSD	143
21.21.1.322	INSN_MATCH_C_FSDSP	143
21.21.1.323	INSN_MATCH_C_FSW	143
21.21.1.324	INSN_MATCH_C_FSWSP	143
21.21.1.325	INSN_MATCH_C_LD	143
21.21.1.326	INSN_MATCH_C_LDSP	144
21.21.1.327	INSN_MATCH_C_LW	144
21.21.1.328	INSN_MATCH_C_LWSP	144

21.21.1.329	INSN_MATCH_C_SD	144
21.21.1.330	INSN_MATCH_C_SDSP	144
21.21.1.331	INSN_MATCH_C_SW	144
21.21.1.332	INSN_MATCH_C_SWSP	144
21.21.1.333	INSN_MATCH_FLD	144
21.21.1.334	INSN_MATCH_FLQ	145
21.21.1.335	INSN_MATCH_FLW	145
21.21.1.336	INSN_MATCH_FSD	145
21.21.1.337	INSN_MATCH_FSQ	145
21.21.1.338	INSN_MATCH_FSW	145
21.21.1.339	INSN_MATCH_LB	145
21.21.1.340	INSN_MATCH_LBU	145
21.21.1.341	INSN_MATCH_LD	145
21.21.1.342	INSN_MATCH_LH	146
21.21.1.343	INSN_MATCH_LHU	146
21.21.1.344	INSN_MATCH_LW	146
21.21.1.345	INSN_MATCH_LWU	146
21.21.1.346	INSN_MATCH_SB	146
21.21.1.347	INSN_MATCH_SD	146
21.21.1.348	INSN_MATCH_SH	146
21.21.1.349	INSN_MATCH_SW	146
21.21.1.350	INSN_MATCH_WFI	147
21.21.1.351	RQ_M_EXT	147
21.21.1.352	RQ_M_SOFT	147
21.21.1.353	RQ_M_TIMER	147
21.21.1.354	RQ_S_EXT	147
21.21.1.355	RQ_S_GEXT	147
21.21.1.356	RQ_S_SOFT	147
21.21.1.357	RQ_S_TIMER	147
21.21.1.358	RQ_VS_EXT	148

21.21.1.359	IRQ_VS_SOFT	148
21.21.1.360	IRQ_VS_TIMER	148
21.21.1.361	IOG_REGBYTES	148
21.21.1.362	IRQ_MASK_FUNCT3	148
21.21.1.363	IRQ_MEIP	148
21.21.1.364	IRQ_MSIP	148
21.21.1.365	IRQ_MTIP	148
21.21.1.366	IRQ_SEIP	149
21.21.1.367	IRQ_SGEIP	149
21.21.1.368	IRQ_SSIP	149
21.21.1.369	IRQ_STIP	149
21.21.1.370	IRQ_VSEIP	149
21.21.1.371	IRQ_VSSIP	149
21.21.1.372	IRQ_VSTIP	149
21.21.1.373	STATUS32_SD [1/2]	149
21.21.1.374	STATUS32_SD [2/2]	150
21.21.1.375	STATUS64_SD	150
21.21.1.376	STATUS_FS	150
21.21.1.377	STATUS_MIE	150
21.21.1.378	STATUS_MPIE	150
21.21.1.379	STATUS_MPP	150
21.21.1.380	STATUS_MPP_SHIFT	150
21.21.1.381	STATUS_MPRV	150
21.21.1.382	STATUS_MXR	151
21.21.1.383	STATUS_SD	151
21.21.1.384	STATUS_SIE	151
21.21.1.385	STATUS_SPIE	151
21.21.1.386	STATUS_SPIE_SHIFT	151
21.21.1.387	STATUS_SPP	151
21.21.1.388	STATUS_SPP_SHIFT	151

21.21.1.390	STATUS_SUM	151
21.21.1.390	STATUS_TSR	152
21.21.1.391	STATUS_TVM	152
21.21.1.392	STATUS_TW	152
21.21.1.393	STATUS_UBE	152
21.21.1.394	STATUS_XS	152
21.21.1.395	STATUSH_MBE	152
21.21.1.396	STATUSH_MPV	152
21.21.1.397	STATUSH_SBE	152
21.21.1.398	BMP_A	153
21.21.1.399	BMP_A_NA4	153
21.21.1.400	BMP_A_NAPOT	153
21.21.1.401	BMP_A_TOR	153
21.21.1.402	BMP_COUNT	153
21.21.1.403	BMP_L	153
21.21.1.404	BMP_R	153
21.21.1.405	BMP_SHIFT	153
21.21.1.406	BMP_W	154
21.21.1.407	BMP_X	154
21.21.1.408	BRV_M	154
21.21.1.409	BRV_S	154
21.21.1.410	BRV_U	154
21.21.1.411	BTE_A	154
21.21.1.412	BTE_D	154
21.21.1.413	BTE_G	154
21.21.1.414	BTE_PPN_SHIFT	155
21.21.1.415	BTE_R	155
21.21.1.416	BTE_SOFT	155
21.21.1.417	BTE_TABLE	155
21.21.1.418	BTE_U	155

21.21.1.41	OTE_V	155
21.21.1.42	OTE_W	155
21.21.1.42	OTE_X	156
21.21.1.42	REG_MASK	156
21.21.1.42	REG_OFFSET	156
21.21.1.42	REG_PTR	156
21.21.1.42	REGBYTES	156
21.21.1.42	RISCV_PGLEVEL_BITS	156
21.21.1.42	RISCV_PGSHIFT	156
21.21.1.42	RISCV_PGSIZE	157
21.21.1.42	RV_X	157
21.21.1.43	RV_LD_IMM	157
21.21.1.43	RV_LDSP_IMM	157
21.21.1.43	RV_LW_IMM	157
21.21.1.43	RV_LWSP_IMM	158
21.21.1.43	RV_RS1S	158
21.21.1.43	RV_RS2	158
21.21.1.43	RV_RS2S	158
21.21.1.43	RV_SDSP_IMM	158
21.21.1.43	RV_SWSP_IMM	159
21.21.1.43	SATP32_ASID	159
21.21.1.44	SATP32_MODE	159
21.21.1.44	SATP32_PPN	159
21.21.1.44	SATP64_ASID	159
21.21.1.44	SATP64_MODE	159
21.21.1.44	SATP64_PPN	159
21.21.1.44	SATP_MODE	160
21.21.1.44	SATP_MODE_OFF	160
21.21.1.44	SATP_MODE_SV32	160
21.21.1.44	SATP_MODE_SV39	160

21.21.1.449	ATP_MODE_SV48	160
21.21.1.450	ATP_MODE_SV57	160
21.21.1.453	ATP_MODE_SV64	160
21.21.1.452	ET_RD	161
21.21.1.453	H_RD	161
21.21.1.453	H_RS1	161
21.21.1.453	H_RS2	161
21.21.1.456	H_RS2C	161
21.21.1.457	SHIFT_RIGHT	161
21.21.1.458	IP_SSIP	161
21.21.1.459	IP_STIP	162
21.21.1.460	STATUS32_SD	162
21.21.1.463	STATUS64_SD	162
21.21.1.463	STATUS64_UXL	162
21.21.1.463	STATUS_FS	162
21.21.1.463	STATUS_MXR	162
21.21.1.465	STATUS_SD	162
21.21.1.466	STATUS_SIE	162
21.21.1.467	STATUS_SPIE	163
21.21.1.468	STATUS_SPIE_SHIFT	163
21.21.1.469	STATUS_SPP	163
21.21.1.470	STATUS_SPP_SHIFT	163
21.21.1.473	STATUS_SUM	163
21.21.1.473	STATUS_XS	163
21.22	include/sbi/riscv_fp.h File Reference	164
21.22.1	Macro Definition Documentation	165
21.22.1.1	GET_PRECISION	165
21.22.1.2	GET_RM	165
21.22.1.3	PRECISION_D	165
21.22.1.4	PRECISION_S	165

21.23include/sbi/riscv_io.h File Reference	165
21.23.1 Macro Definition Documentation	167
21.23.1.1 __io_ar	167
21.23.1.2 __io_aw	167
21.23.1.3 __io_br	167
21.23.1.4 __io_bw	167
21.23.1.5 __io_rar	167
21.23.1.6 __io_raw	167
21.23.1.7 __io_rbr	167
21.23.1.8 __io_rbw	168
21.23.1.9 readb	168
21.23.1.10readb_relaxed	168
21.23.1.11readl	168
21.23.1.12readl_relaxed	168
21.23.1.13readq	168
21.23.1.14readq_relaxed	168
21.23.1.15readw	169
21.23.1.16readw_relaxed	169
21.23.1.17writeb	169
21.23.1.18writeb_relaxed	169
21.23.1.19writel	169
21.23.1.20writel_relaxed	169
21.23.1.21writeq	170
21.23.1.22writeq_relaxed	170
21.23.1.23writew	170
21.23.1.24writew_relaxed	170
21.23.2 Function Documentation	170
21.23.2.1 __raw_readb()	170
21.23.2.2 __raw_readl()	170
21.23.2.3 __raw_readq()	171

21.23.2.4	<code>__raw_readw()</code>	171
21.23.2.5	<code>__raw_writeb()</code>	171
21.23.2.6	<code>__raw_writel()</code>	171
21.23.2.7	<code>__raw_writew()</code>	171
21.23.2.8	<code>__raw_writel()</code>	171
21.24	<code>include/sbi/riscv_locks.h</code> File Reference	172
21.24.1	Macro Definition Documentation	172
21.24.1.1	<code>__RISCV_SPIN_UNLOCKED</code>	172
21.24.1.2	<code>SPIN_LOCK_INIT</code>	172
21.24.1.3	<code>SPIN_LOCK_INITIALIZER</code>	173
21.24.2	Function Documentation	173
21.24.2.1	<code>spin_lock()</code>	173
21.24.2.2	<code>spin_lock_check()</code>	174
21.24.2.3	<code>spin_trylock()</code>	175
21.24.2.4	<code>spin_unlock()</code>	176
21.25	<code>include/sbi/sbi_bitops.h</code> File Reference	177
21.25.1	Macro Definition Documentation	179
21.25.1.1	<code>ffz</code>	179
21.25.2	Function Documentation	179
21.25.2.1	<code>__ffs()</code>	179
21.25.2.2	<code>__fls()</code>	179
21.25.2.3	<code>ffs()</code>	180
21.25.2.4	<code>fls()</code>	180
21.26	<code>include/sbi/sbi_bits.h</code> File Reference	180
21.26.1	Macro Definition Documentation	180
21.26.1.1	<code>BIT_MASK</code>	180
21.26.1.2	<code>BIT_WORD</code>	181
21.26.1.3	<code>EXTRACT_FIELD</code>	181
21.26.1.4	<code>INSERT_FIELD</code>	181
21.27	<code>include/sbi/sbi_console.h</code> File Reference	181

21.27.1 Macro Definition Documentation	182
21.27.1.1 __printf	182
21.27.2 Function Documentation	182
21.27.2.1 __printf() [1/3]	182
21.27.2.2 __printf() [2/3]	183
21.27.2.3 __printf() [3/3]	183
21.27.2.4 sbi_console_init()	183
21.27.2.5 sbi_getc()	184
21.27.2.6 sbi_gets()	184
21.27.2.7 sbi_isprintable()	185
21.27.2.8 sbi_putc()	185
21.27.2.9 sbi_puts()	185
21.27.3 Variable Documentation	186
21.27.3.1 format	186
21.27.3.2 out_sz	186
21.28include/sbi/sbi_const.h File Reference	186
21.28.1 Macro Definition Documentation	187
21.28.1.1 __AC	187
21.28.1.2 __STR	187
21.28.1.3 _AC	187
21.28.1.4 _AT	187
21.28.1.5 _BITUL	188
21.28.1.6 _BITULL	188
21.28.1.7 _UL	188
21.28.1.8 _ULL	188
21.28.1.9 STRINGIFY	188
21.28.1.10UL	188
21.28.1.11ULL	188
21.29include/sbi/sbi_ecall.h File Reference	189
21.29.1 Macro Definition Documentation	190

21.29.1.1 SBI_ECALL_VERSION_MAJOR	190
21.29.1.2 SBI_ECALL_VERSION_MINOR	190
21.29.1.3 SBI_OPENSBI_IMPID	190
21.29.2 Function Documentation	190
21.29.2.1 sbi_ecall_find_extension()	190
21.29.2.2 sbi_ecall_handler()	191
21.29.2.3 sbi_ecall_init()	191
21.29.2.4 sbi_ecall_register_extension()	192
21.29.2.5 sbi_ecall_unregister_extension()	192
21.29.2.6 sbi_ecall_version_major()	193
21.29.2.7 sbi_ecall_version_minor()	193
21.29.3 Variable Documentation	194
21.29.3.1 ecall_base	194
21.29.3.2 ecall_ipi	194
21.29.3.3 ecall_legacy	194
21.29.3.4 ecall_rfence	194
21.29.3.5 ecall_time	194
21.29.3.6 ecall_vendor	194
21.30include/sbi/sbi_ecall_interface.h File Reference	194
21.30.1 Macro Definition Documentation	195
21.30.1.1 SBI_EXT_0_1_CLEAR_IPI	195
21.30.1.2 SBI_EXT_0_1_CONSOLE_GETCHAR	195
21.30.1.3 SBI_EXT_0_1_CONSOLE_PUTCHAR	196
21.30.1.4 SBI_EXT_0_1_REMOTE_FENCE_I	196
21.30.1.5 SBI_EXT_0_1_REMOTE_SFENCE_VMA	196
21.30.1.6 SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID	196
21.30.1.7 SBI_EXT_0_1_SEND_IPI	196
21.30.1.8 SBI_EXT_0_1_SET_TIMER	196
21.30.1.9 SBI_EXT_0_1_SHUTDOWN	196
21.30.1.10SBI_EXT_BASE	196

21.30.1.11	SBI_EXT_BASE_GET_IMP_ID	197
21.30.1.12	SBI_EXT_BASE_GET_IMP_VERSION	197
21.30.1.13	SBI_EXT_BASE_GET_MARCHID	197
21.30.1.14	SBI_EXT_BASE_GET_MIMPID	197
21.30.1.15	SBI_EXT_BASE_GET_MVENDORID	197
21.30.1.16	SBI_EXT_BASE_GET_SPEC_VERSION	197
21.30.1.17	SBI_EXT_BASE_PROBE_EXT	197
21.30.1.18	SBI_EXT_IPI	197
21.30.1.19	SBI_EXT_IPI_SEND_IPI	198
21.30.1.20	SBI_EXT_RFENCE	198
21.30.1.21	SBI_EXT_RFENCE_REMOTE_FENCE_I	198
21.30.1.22	SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA	198
21.30.1.23	SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID	198
21.30.1.24	SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA	198
21.30.1.25	SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID	198
21.30.1.26	SBI_EXT_RFENCE_REMOTE_SFENCE_VMA	198
21.30.1.27	SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID	199
21.30.1.28	SBI_EXT_TIME	199
21.30.1.29	SBI_EXT_TIME_SET_TIMER	199
21.30.1.30	SBI_EXT_VENDOR_END	199
21.30.1.31	SBI_EXT_VENDOR_START	199
21.30.1.32	SBI_SPEC_VERSION_MAJOR_MASK	199
21.30.1.33	SBI_SPEC_VERSION_MAJOR_OFFSET	199
21.30.1.34	SBI_SPEC_VERSION_MINOR_MASK	199
21.31	include/sbi/sbi_emulate_csr.h File Reference	200
21.31.1	Function Documentation	200
21.31.1.1	sbi_emulate_csr_read()	201
21.31.1.2	sbi_emulate_csr_write()	201
21.32	include/sbi/sbi_error.h File Reference	202
21.32.1	Macro Definition Documentation	202

21.32.1.1 SBI_DENIED	202
21.32.1.2 SBI_EFAIL	203
21.32.1.3 SBI_EILL	203
21.32.1.4 SBI_EINVAL	203
21.32.1.5 SBI_EIO	203
21.32.1.6 SBI_ENODEV	203
21.32.1.7 SBI_ENOENT	203
21.32.1.8 SBI_ENOMEM	203
21.32.1.9 SBI_ENOSPC	203
21.32.1.10 SBI_ENOSYS	204
21.32.1.11 SBI_ENOTSUPP	204
21.32.1.12 SBI_ETIMEDOUT	204
21.32.1.13 SBI_ETRAP	204
21.32.1.14 SBI_EUNKNOWN	204
21.32.1.15 SBI_INVALID_ADDR	204
21.32.1.16 SBI_OK	204
21.33 include/sbi/sbi_fifo.h File Reference	205
21.33.1 Enumeration Type Documentation	206
21.33.1.1 sbi_fifo_inplace_update_types	206
21.33.2 Function Documentation	206
21.33.2.1 sbi_fifo_avail()	206
21.33.2.2 sbi_fifo_dequeue()	206
21.33.2.3 sbi_fifo_enqueue()	207
21.33.2.4 sbi_fifo_init()	208
21.33.2.5 sbi_fifo_inplace_update()	209
21.33.2.6 sbi_fifo_is_empty()	210
21.33.2.7 sbi_fifo_is_full()	210
21.34 include/sbi/sbi_hart.h File Reference	211
21.34.1 Function Documentation	212
21.34.1.1 __attribute__((__aligned__))	212

21.34.1.2 <code>sbi_current_hartid()</code>	212
21.34.1.3 <code>sbi_hart_available_mask()</code>	212
21.34.1.4 <code>sbi_hart_delegation_dump()</code>	213
21.34.1.5 <code>sbi_hart_get_trap_info()</code>	214
21.34.1.6 <code>sbi_hart_id_to_scratch()</code>	214
21.34.1.7 <code>sbi_hart_init()</code>	214
21.34.1.8 <code>sbi_hart_mark_available()</code>	215
21.34.1.9 <code>sbi_hart_pmp_dump()</code>	216
21.34.1.10 <code>sbi_hart_set_trap_info()</code>	217
21.34.1.11 <code>sbi_hart_unmark_available()</code>	217
21.34.1.12 <code>sbi_hart_wait_for_coldboot()</code>	218
21.34.1.13 <code>sbi_hart_wake_coldboot_harts()</code>	218
21.34.2 Variable Documentation	219
21.34.2.1 <code>arg1</code>	219
21.34.2.2 <code>next_addr</code>	219
21.34.2.3 <code>next_mode</code>	219
21.34.2.4 <code>next_virt</code>	219
21.35 <code>include/sbi/sbi_hfence.h</code> File Reference	220
21.35.1 Function Documentation	220
21.35.1.1 <code>__sbi_hfence_gvma_all()</code>	220
21.35.1.2 <code>__sbi_hfence_gvma_gpa()</code>	221
21.35.1.3 <code>__sbi_hfence_gvma_vmid()</code>	221
21.35.1.4 <code>__sbi_hfence_gvma_vmid_gpa()</code>	221
21.35.1.5 <code>__sbi_hfence_vvma_all()</code>	222
21.35.1.6 <code>__sbi_hfence_vvma_asid()</code>	222
21.35.1.7 <code>__sbi_hfence_vvma_asid_va()</code>	222
21.35.1.8 <code>__sbi_hfence_vvma_va()</code>	223
21.36 <code>include/sbi/sbi_illegal_insn.h</code> File Reference	223
21.36.1 Function Documentation	224
21.36.1.1 <code>sbi_illegal_insn_handler()</code>	224

21.37include/sbi/sbi_init.h File Reference	225
21.37.1 Function Documentation	225
21.37.1.1 sbi_exit()	226
21.37.1.2 sbi_init()	227
21.37.1.3 sbi_init_count()	228
21.38include/sbi/sbi_ipi.h File Reference	229
21.38.1 Macro Definition Documentation	230
21.38.1.1 SBI_IPI_EVENT_MAX	230
21.38.2 Function Documentation	230
21.38.2.1 sbi_ipi_clear_smode()	230
21.38.2.2 sbi_ipi_event_create()	231
21.38.2.3 sbi_ipi_event_destroy()	231
21.38.2.4 sbi_ipi_exit()	231
21.38.2.5 sbi_ipi_init()	232
21.38.2.6 sbi_ipi_process()	233
21.38.2.7 sbi_ipi_send_halt()	233
21.38.2.8 sbi_ipi_send_many()	234
21.38.2.9 sbi_ipi_send_smode()	235
21.39include/sbi/sbi_list.h File Reference	236
21.39.1 Macro Definition Documentation	237
21.39.1.1 SBI_INIT_LIST_HEAD	237
21.39.1.2 sbi_list_entry	237
21.39.1.3 sbi_list_first_entry	237
21.39.1.4 sbi_list_for_each	238
21.39.1.5 sbi_list_for_each_entry	238
21.39.1.6 SBI_LIST_HEAD	238
21.39.1.7 SBI_LIST_HEAD_INIT	239
21.39.1.8 sbi_list_last_entry	239
21.39.1.9 SBI_LIST_POISON_NEXT	239
21.39.1.10SBI_LIST_POISON_PREV	239

21.39.2 Function Documentation	239
21.39.2.1 __sbi_list_add()	239
21.39.2.2 __sbi_list_del()	240
21.39.2.3 __sbi_list_del_entry()	240
21.39.2.4 sbi_list_add()	241
21.39.2.5 sbi_list_add_tail()	241
21.39.2.6 sbi_list_del()	242
21.39.2.7 sbi_list_del_init()	242
21.40 include/sbi/sbi_misaligned_ldst.h File Reference	243
21.40.1 Function Documentation	244
21.40.1.1 sbi_misaligned_load_handler()	244
21.40.1.2 sbi_misaligned_store_handler()	245
21.41 include/sbi/sbi_platform.h File Reference	246
21.41.1 Macro Definition Documentation	248
21.41.1.1 SBI_PLATFORM_DEFAULT_FEATURES	248
21.41.1.2 SBI_PLATFORM_DISABLED_HART_OFFSET	248
21.41.1.3 SBI_PLATFORM_FEATURES_OFFSET	248
21.41.1.4 SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET	248
21.41.1.5 SBI_PLATFORM_HART_COUNT_OFFSET	248
21.41.1.6 SBI_PLATFORM_HART_STACK_SIZE_OFFSET	249
21.41.1.7 sbi_platform_has_hart_hotplug	249
21.41.1.8 sbi_platform_has_mcounteren	249
21.41.1.9 sbi_platform_has_mfaults_delegation	249
21.41.1.10 sbi_platform_has_pmp	249
21.41.1.11 sbi_platform_has_scounteren	249
21.41.1.12 sbi_platform_has_timer_value	249
21.41.1.13 SBI_PLATFORM_NAME_OFFSET	250
21.41.1.14 SBI_PLATFORM_OPENSBI_VERSION_OFFSET	250
21.41.1.15 sbi_platform_ops	250
21.41.1.16 SBI_PLATFORM_OPS_OFFSET	250

21.41.1.17	<code>sbi_platform_ptr</code>	250
21.41.1.18	<code>sbi_platform_thishart_ptr</code>	250
21.41.1.19	<code>SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT</code>	250
21.41.1.20	<code>SBI_PLATFORM_VERSION</code>	251
21.41.1.21	<code>SBI_PLATFORM_VERSION_OFFSET</code>	251
21.41.2	Enumeration Type Documentation	251
21.41.2.1	<code>sbi_platform_features</code>	251
21.41.3	Function Documentation	251
21.41.3.1	<code>sbi_platform_console_getc()</code>	251
21.41.3.2	<code>sbi_platform_console_init()</code>	252
21.41.3.3	<code>sbi_platform_console_putc()</code>	253
21.41.3.4	<code>sbi_platform_early_exit()</code>	253
21.41.3.5	<code>sbi_platform_early_init()</code>	254
21.41.3.6	<code>sbi_platform_final_exit()</code>	254
21.41.3.7	<code>sbi_platform_final_init()</code>	255
21.41.3.8	<code>sbi_platform_hart_count()</code>	255
21.41.3.9	<code>sbi_platform_hart_disabled()</code>	256
21.41.3.10	<code>sbi_platform_hart_stack_size()</code>	257
21.41.3.11	<code>sbi_platform_ipi_clear()</code>	257
21.41.3.12	<code>sbi_platform_ipi_exit()</code>	258
21.41.3.13	<code>sbi_platform_ipi_init()</code>	258
21.41.3.14	<code>sbi_platform_ipi_send()</code>	259
21.41.3.15	<code>sbi_platform_irqchip_exit()</code>	259
21.41.3.16	<code>sbi_platform_irqchip_init()</code>	260
21.41.3.17	<code>sbi_platform_misa_extension()</code>	261
21.41.3.18	<code>sbi_platform_misa_xlen()</code>	261
21.41.3.19	<code>sbi_platform_name()</code>	262
21.41.3.20	<code>sbi_platform_pmp_region_count()</code>	262
21.41.3.21	<code>sbi_platform_pmp_region_info()</code>	263
21.41.3.22	<code>sbi_platform_system_reboot()</code>	264

21.41.3.23	sbi_platform_system_shutdown()	264
21.41.3.24	sbi_platform_timer_event_start()	265
21.41.3.25	sbi_platform_timer_event_stop()	265
21.41.3.26	sbi_platform_timer_exit()	266
21.41.3.27	sbi_platform_timer_init()	266
21.41.3.28	sbi_platform_timer_value()	267
21.41.3.29	sbi_platform_tlbr_flush_limit()	268
21.41.3.30	sbi_platform_vendor_ext_check()	268
21.41.3.31	sbi_platform_vendor_ext_provider()	269
21.41.4	Variable Documentation	270
21.41.4.1	__packed	270
21.42	include/sbi/sbi_scratch.h File Reference	270
21.42.1	Macro Definition Documentation	271
21.42.1.1	SBI_SCRATCH_EXTRA_SPACE_OFFSET	272
21.42.1.2	SBI_SCRATCH_FW_SIZE_OFFSET	272
21.42.1.3	SBI_SCRATCH_FW_START_OFFSET	272
21.42.1.4	SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET	272
21.42.1.5	SBI_SCRATCH_NEXT_ADDR_OFFSET	272
21.42.1.6	SBI_SCRATCH_NEXT_ARG1_OFFSET	272
21.42.1.7	SBI_SCRATCH_NEXT_MODE_OFFSET	272
21.42.1.8	sbi_scratch_offset_ptr	273
21.42.1.9	SBI_SCRATCH_OPTIONS_OFFSET	273
21.42.1.10	SBI_SCRATCH_PLATFORM_ADDR_OFFSET	273
21.42.1.11	SBI_SCRATCH_SIZE	273
21.42.1.12	sbi_scratch_thishart_arg1_ptr	273
21.42.1.13	sbi_scratch_thishart_offset_ptr	273
21.42.1.14	sbi_scratch_thishart_ptr	273
21.42.1.15	SBI_SCRATCH_TMP0_OFFSET	274
21.42.1.16	SBI_SCRATCH_WARMBOOT_ADDR_OFFSET	274
21.42.2	Enumeration Type Documentation	274

21.42.2.1 sbi_scratch_options	274
21.42.3 Function Documentation	274
21.42.3.1 sbi_scratch_alloc_offset()	274
21.42.3.2 sbi_scratch_free_offset()	275
21.42.4 Variable Documentation	276
21.42.4.1 __packed	276
21.43include/sbi/sbi_string.h File Reference	276
21.43.1 Function Documentation	277
21.43.1.1 sbi_memchr()	277
21.43.1.2 sbi_memcmp()	277
21.43.1.3 sbi_memcpy()	277
21.43.1.4 sbi_memmove()	277
21.43.1.5 sbi_memset()	278
21.43.1.6 sbi_strchr()	278
21.43.1.7 sbi_strcmp()	278
21.43.1.8 sbi_strcpy()	278
21.43.1.9 sbi_strlen()	279
21.43.1.10sbi_strncpy()	279
21.43.1.11sbi_strnlen()	279
21.43.1.12sbi_strrchr()	279
21.44include/sbi/sbi_system.h File Reference	280
21.44.1 Function Documentation	280
21.44.1.1 sbi_system_early_exit()	281
21.44.1.2 sbi_system_early_init()	281
21.44.1.3 sbi_system_final_exit()	282
21.44.1.4 sbi_system_final_init()	282
21.44.1.5 sbi_system_reboot()	283
21.44.1.6 sbi_system_shutdown()	283
21.45include/sbi/sbi_timer.h File Reference	284
21.45.1 Function Documentation	285

21.45.1.1 sbi_timer_event_start()	285
21.45.1.2 sbi_timer_exit()	285
21.45.1.3 sbi_timer_get_delta()	286
21.45.1.4 sbi_timer_init()	287
21.45.1.5 sbi_timer_process()	287
21.45.1.6 sbi_timer_set_delta()	288
21.45.1.7 sbi_timer_set_delta_upper()	288
21.45.1.8 sbi_timer_value()	288
21.45.1.9 sbi_timer_virt_value()	289
21.46 include/sbi/sbi_tlb.h File Reference	289
21.46.1 Macro Definition Documentation	291
21.46.1.1 SBI_TLB_FIFO_NUM_ENTRIES	291
21.46.1.2 SBI_TLB_FLUSH_ALL	291
21.46.1.3 SBI_TLB_INFO_SIZE	291
21.46.2 Enumeration Type Documentation	291
21.46.2.1 sbi_tlb_info_types	291
21.46.3 Function Documentation	291
21.46.3.1 sbi_tlb_init()	292
21.46.3.2 sbi_tlb_request()	292
21.47 include/sbi/sbi_trap.h File Reference	293
21.47.1 Macro Definition Documentation	295
21.47.1.1 SBI_TRAP_REGS_a0	295
21.47.1.2 SBI_TRAP_REGS_a1	295
21.47.1.3 SBI_TRAP_REGS_a2	295
21.47.1.4 SBI_TRAP_REGS_a3	295
21.47.1.5 SBI_TRAP_REGS_a4	295
21.47.1.6 SBI_TRAP_REGS_a5	296
21.47.1.7 SBI_TRAP_REGS_a6	296
21.47.1.8 SBI_TRAP_REGS_a7	296
21.47.1.9 SBI_TRAP_REGS_gp	296

21.47.1.10	SBI_TRAP_REGS_last	296
21.47.1.11	SBI_TRAP_REGS_mepc	296
21.47.1.12	SBI_TRAP_REGS_mstatus	296
21.47.1.13	SBI_TRAP_REGS_mstatusH	296
21.47.1.14	SBI_TRAP_REGS_OFFSET	297
21.47.1.15	SBI_TRAP_REGS_ra	297
21.47.1.16	SBI_TRAP_REGS_s0	297
21.47.1.17	SBI_TRAP_REGS_s1	297
21.47.1.18	SBI_TRAP_REGS_s10	297
21.47.1.19	SBI_TRAP_REGS_s11	297
21.47.1.20	SBI_TRAP_REGS_s2	297
21.47.1.21	SBI_TRAP_REGS_s3	298
21.47.1.22	SBI_TRAP_REGS_s4	298
21.47.1.23	SBI_TRAP_REGS_s5	298
21.47.1.24	SBI_TRAP_REGS_s6	298
21.47.1.25	SBI_TRAP_REGS_s7	298
21.47.1.26	SBI_TRAP_REGS_s8	298
21.47.1.27	SBI_TRAP_REGS_s9	298
21.47.1.28	SBI_TRAP_REGS_SIZE	298
21.47.1.29	SBI_TRAP_REGS_sp	299
21.47.1.30	SBI_TRAP_REGS_t0	299
21.47.1.31	SBI_TRAP_REGS_t1	299
21.47.1.32	SBI_TRAP_REGS_t2	299
21.47.1.33	SBI_TRAP_REGS_t3	299
21.47.1.34	SBI_TRAP_REGS_t4	299
21.47.1.35	SBI_TRAP_REGS_t5	299
21.47.1.36	SBI_TRAP_REGS_t6	299
21.47.1.37	SBI_TRAP_REGS_tp	300
21.47.1.38	SBI_TRAP_REGS_zero	300
21.47.2	Function Documentation	300

21.47.2.1 sbi_trap_handler()	300
21.47.2.2 sbi_trap_redirect()	301
21.47.3 Variable Documentation	302
21.47.3.1 __packed	302
21.48include/sbi/sbi_types.h File Reference	302
21.48.1 Macro Definition Documentation	303
21.48.1.1 __noreturn	303
21.48.1.2 __packed	303
21.48.1.3 CLAMP	304
21.48.1.4 container_of	304
21.48.1.5 FALSE	304
21.48.1.6 likely	304
21.48.1.7 MAX	304
21.48.1.8 MIN	304
21.48.1.9 NULL	305
21.48.1.10offsetof	305
21.48.1.11ROUNDDOWN	305
21.48.1.12ROUNDUP	305
21.48.1.13STR	305
21.48.1.14TRUE	305
21.48.1.15unlikely	305
21.48.1.16XSTR	306
21.48.2 Typedef Documentation	306
21.48.2.1 bool	306
21.48.2.2 int16_t	306
21.48.2.3 int32_t	306
21.48.2.4 physical_addr_t	306
21.48.2.5 physical_size_t	306
21.48.2.6 s16	306
21.48.2.7 s32	307

21.48.2.8 s8	307
21.48.2.9 size_t	307
21.48.2.10 ssize_t	307
21.48.2.11 u16	307
21.48.2.12 i32	307
21.48.2.13 i8	307
21.48.2.14 uint16_t	307
21.48.2.15 uint32_t	308
21.48.2.16 uint8_t	308
21.48.2.17 uintptr_t	308
21.48.2.18 long	308
21.48.2.19 virtual_addr_t	308
21.48.2.20 virtual_size_t	308
21.49 include/sbi/sbi_unpriv.h File Reference	309
21.49.1 Macro Definition Documentation	309
21.49.1.1 DECLARE_UNPRIVILEGED_LOAD_FUNCTION	310
21.49.1.2 DECLARE_UNPRIVILEGED_STORE_FUNCTION	310
21.49.2 Function Documentation	310
21.49.2.1 sbi_get_insn()	310
21.50 include/sbi/sbi_version.h File Reference	311
21.50.1 Macro Definition Documentation	311
21.50.1.1 OPENSBI_VERSION	311
21.50.1.2 OPENSBI_VERSION_MAJOR	312
21.50.1.3 OPENSBI_VERSION_MINOR	312
21.51 include/sbi_utils/irqchip/plic.h File Reference	312
21.51.1 Function Documentation	313
21.51.1.1 plic_cold_irqchip_init()	313
21.51.1.2 plic_fdt_fixup()	313
21.51.1.3 plic_set_ie()	314
21.51.1.4 plic_set_thresh()	314

21.51.1.5 plic_warm_irqchip_init()	314
21.52include/sbi_utils/serial/sifive-uart.h File Reference	315
21.52.1 Function Documentation	316
21.52.1.1 sifive_uart_getc()	316
21.52.1.2 sifive_uart_init()	316
21.52.1.3 sifive_uart_putc()	317
21.53include/sbi_utils/serial/uart8250.h File Reference	317
21.53.1 Function Documentation	318
21.53.1.1 uart8250_getc()	318
21.53.1.2 uart8250_init()	319
21.53.1.3 uart8250_putc()	319
21.54include/sbi_utils/sys/clint.h File Reference	320
21.54.1 Function Documentation	320
21.54.1.1 clint_cold_ipi_init()	321
21.54.1.2 clint_cold_timer_init()	321
21.54.1.3 clint_ipi_clear()	321
21.54.1.4 clint_ipi_send()	322
21.54.1.5 clint_ipi_sync()	322
21.54.1.6 clint_timer_event_start()	322
21.54.1.7 clint_timer_event_stop()	322
21.54.1.8 clint_timer_value()	323
21.54.1.9 clint_warm_ipi_init()	323
21.54.1.10clint_warm_timer_init()	323
21.55include/sbi_utils/sys/htif.h File Reference	324
21.55.1 Function Documentation	324
21.55.1.1 htif_getc()	325
21.55.1.2 htif_putc()	325
21.55.1.3 htif_system_down()	326
21.56lib/sbi/riscv_asm.c File Reference	326
21.56.1 Function Documentation	326

21.56.1.1	<code>csr_read_num()</code>	327
21.56.1.2	<code>csr_write_num()</code>	327
21.56.1.3	<code>ctz()</code>	327
21.56.1.4	<code>misa_extension_imp()</code>	328
21.56.1.5	<code>misa_xlen()</code>	328
21.56.1.6	<code>pmp_get()</code>	329
21.56.1.7	<code>pmp_set()</code>	329
21.57	<code>lib/sbi/riscv_atomic.c</code> File Reference	330
21.57.1	Macro Definition Documentation	331
21.57.1.1	<code>__atomic_op_bit</code>	331
21.57.1.2	<code>__atomic_op_bit_ord</code>	331
21.57.1.3	<code>__axchg</code>	332
21.57.1.4	<code>__cmpxchg</code>	332
21.57.1.5	<code>__NOP</code>	332
21.57.1.6	<code>__NOT</code>	332
21.57.1.7	<code>__xchg</code>	333
21.57.1.8	<code>axchg</code>	333
21.57.1.9	<code>cmpxchg</code>	333
21.57.1.10	<code>xchg</code>	333
21.57.2	Function Documentation	334
21.57.2.1	<code>arch_atomic_cmpxchg()</code>	334
21.57.2.2	<code>arch_atomic_xchg()</code>	334
21.57.2.3	<code>atomic_add_return()</code>	334
21.57.2.4	<code>atomic_clear_bit()</code>	335
21.57.2.5	<code>atomic_raw_clear_bit()</code>	335
21.57.2.6	<code>atomic_raw_set_bit()</code>	335
21.57.2.7	<code>atomic_raw_xchg_uint()</code>	336
21.57.2.8	<code>atomic_raw_xchg_ulong()</code>	336
21.57.2.9	<code>atomic_read()</code>	336
21.57.2.10	<code>atomic_set_bit()</code>	337

21.57.2.1	<code>atomic_sub_return()</code>	337
21.57.2.12	<code>atomic_write()</code>	337
21.58	<code>lib/sbi/riscv_locks.c</code> File Reference	337
21.58.1	Function Documentation	338
21.58.1.1	<code>spin_lock()</code>	338
21.58.1.2	<code>spin_lock_check()</code>	339
21.58.1.3	<code>spin_trylock()</code>	340
21.58.1.4	<code>spin_unlock()</code>	341
21.59	<code>lib/sbi/sbi_console.c</code> File Reference	342
21.59.1	Macro Definition Documentation	344
21.59.1.1	<code>PAD_ALTERNATE</code>	344
21.59.1.2	<code>PAD_RIGHT</code>	344
21.59.1.3	<code>PAD_ZERO</code>	344
21.59.1.4	<code>PRINT_BUF_LEN</code>	344
21.59.1.5	<code>va_arg</code>	344
21.59.1.6	<code>va_end</code>	344
21.59.1.7	<code>va_start</code>	345
21.59.2	Typedef Documentation	345
21.59.2.1	<code>va_list</code>	345
21.59.3	Function Documentation	345
21.59.3.1	<code>print()</code>	345
21.59.3.2	<code>putc()</code>	346
21.59.3.3	<code>printi()</code>	346
21.59.3.4	<code>prints()</code>	347
21.59.3.5	<code>sbi_console_init()</code>	348
21.59.3.6	<code>sbi_dprintf()</code>	348
21.59.3.7	<code>sbi_getc()</code>	349
21.59.3.8	<code>sbi_gets()</code>	350
21.59.3.9	<code>sbi_isprintable()</code>	350
21.59.3.10	<code>sbi_printf()</code>	350

21.59.3.11	<code>sbi_putc()</code>	351
21.59.3.12	<code>sbi_puts()</code>	352
21.59.3.13	<code>sbi_snprintf()</code>	352
21.59.3.14	<code>sbi_sprintf()</code>	353
21.59.4	Variable Documentation	353
21.59.4.1	<code>console_out_lock</code>	353
21.59.4.2	<code>console_plat</code>	353
21.60	<code>lib/sbi/sbi_ecall.c</code> File Reference	353
21.60.1	Function Documentation	354
21.60.1.1	<code>sbi_ecall_find_extension()</code>	354
21.60.1.2	<code>sbi_ecall_handler()</code>	355
21.60.1.3	<code>sbi_ecall_init()</code>	355
21.60.1.4	<code>sbi_ecall_register_extension()</code>	356
21.60.1.5	<code>sbi_ecall_unregister_extension()</code>	356
21.60.1.6	<code>sbi_ecall_version_major()</code>	357
21.60.1.7	<code>sbi_ecall_version_minor()</code>	357
21.60.1.8	<code>SBI_LIST_HEAD()</code>	358
21.61	<code>lib/sbi/sbi_ecall_base.c</code> File Reference	358
21.61.1	Function Documentation	358
21.61.1.1	<code>sbi_ecall_base_handler()</code>	359
21.61.1.2	<code>sbi_ecall_base_probe()</code>	359
21.61.2	Variable Documentation	360
21.61.2.1	<code>ecall_base</code>	360
21.62	<code>lib/sbi/sbi_ecall_legacy.c</code> File Reference	360
21.62.1	Function Documentation	361
21.62.1.1	<code>sbi_ecall_legacy_handler()</code>	361
21.62.1.2	<code>sbi_load_hart_mask_unpriv()</code>	361
21.62.2	Variable Documentation	362
21.62.2.1	<code>ecall_legacy</code>	362
21.63	<code>lib/sbi/sbi_ecall_replace.c</code> File Reference	362

21.63.1 Function Documentation	363
21.63.1.1 sbi_ecall_ipi_handler()	363
21.63.1.2 sbi_ecall_rfence_handler()	364
21.63.1.3 sbi_ecall_time_handler()	364
21.63.2 Variable Documentation	364
21.63.2.1 ecall_ipi	365
21.63.2.2 ecall_rfence	365
21.63.2.3 ecall_time	365
21.64lib/sbi/sbi_ecall_vendor.c File Reference	366
21.64.1 Function Documentation	366
21.64.1.1 sbi_ecall_vendor_handler()	367
21.64.1.2 sbi_ecall_vendor_probe()	367
21.64.2 Variable Documentation	367
21.64.2.1 ecall_vendor	368
21.65lib/sbi/sbi_emulate_csr.c File Reference	368
21.65.1 Function Documentation	368
21.65.1.1 sbi_emulate_csr_read()	369
21.65.1.2 sbi_emulate_csr_write()	369
21.66lib/sbi/sbi_fifo.c File Reference	370
21.66.1 Function Documentation	371
21.66.1.1 __sbi_fifo_enqueue()	371
21.66.1.2 __sbi_fifo_is_empty()	371
21.66.1.3 __sbi_fifo_is_full()	372
21.66.1.4 __sbi_fifo_reset()	372
21.66.1.5 sbi_fifo_avail()	373
21.66.1.6 sbi_fifo_dequeue()	373
21.66.1.7 sbi_fifo_enqueue()	374
21.66.1.8 sbi_fifo_init()	375
21.66.1.9 sbi_fifo_inplace_update()	375
21.66.1.10sbi_fifo_is_empty()	376

21.66.1.11	<code>sbi_fifo_is_full()</code>	377
21.66.1.12	<code>sbi_fifo_reset()</code>	377
21.67	<code>lib/sbi/sbi_hart.c</code> File Reference	378
21.67.1	Macro Definition Documentation	379
21.67.1.1	<code>COLDBOOT_WAIT_BITMAP_SIZE</code>	379
21.67.2	Typedef Documentation	379
21.67.2.1	<code>h2s</code>	379
21.67.3	Function Documentation	379
21.67.3.1	<code>__attribute__()</code>	379
21.67.3.2	<code>delegate_traps()</code>	380
21.67.3.3	<code>fp_init()</code>	380
21.67.3.4	<code>log2roundup()</code>	380
21.67.3.5	<code>mstatus_init()</code>	381
21.67.3.6	<code>pmp_init()</code>	381
21.67.3.7	<code>sbi_current_hartid()</code>	382
21.67.3.8	<code>sbi_hart_available_mask()</code>	382
21.67.3.9	<code>sbi_hart_delegation_dump()</code>	383
21.67.3.10	<code>sbi_hart_get_trap_info()</code>	383
21.67.3.11	<code>sbi_hart_id_to_scratch()</code>	384
21.67.3.12	<code>sbi_hart_init()</code>	384
21.67.3.13	<code>sbi_hart_mark_available()</code>	385
21.67.3.14	<code>sbi_hart_pmp_dump()</code>	386
21.67.3.15	<code>sbi_hart_set_trap_info()</code>	387
21.67.3.16	<code>sbi_hart_unmark_available()</code>	387
21.67.3.17	<code>sbi_hart_wait_for_coldboot()</code>	388
21.67.3.18	<code>sbi_hart_wake_coldboot_harts()</code>	388
21.67.4	Variable Documentation	389
21.67.4.1	<code>avail_hart_mask</code>	389
21.67.4.2	<code>avail_hart_mask_lock</code>	389
21.67.4.3	<code>coldboot_done</code>	389

21.67.4.4 coldboot_lock	389
21.67.4.5 coldboot_wait_bitmap	389
21.67.4.6 trap_info_offset	390
21.68lib/sbi/sbi_illegal_insn.c File Reference	390
21.68.1 Typedef Documentation	390
21.68.1.1 illegal_insn_func	391
21.68.2 Function Documentation	391
21.68.2.1 sbi_illegal_insn_handler()	391
21.68.2.2 system_opcode_insn()	392
21.68.2.3 truly_illegal_insn()	392
21.68.3 Variable Documentation	393
21.68.3.1 illegal_insn_table	393
21.69lib/sbi/sbi_init.c File Reference	393
21.69.1 Macro Definition Documentation	394
21.69.1.1 BANNER	394
21.69.2 Function Documentation	394
21.69.2.1 init_coldboot()	394
21.69.2.2 init_warmboot()	396
21.69.2.3 sbi_boot_prints()	396
21.69.2.4 sbi_exit()	397
21.69.2.5 sbi_init()	398
21.69.2.6 sbi_init_count()	399
21.69.3 Variable Documentation	400
21.69.3.1 coldboot_lottery	400
21.69.3.2 init_count_offset	400
21.70lib/sbi/sbi_ipi.c File Reference	400
21.70.1 Function Documentation	401
21.70.1.1 sbi_ipi_clear_smode()	401
21.70.1.2 sbi_ipi_event_create()	402
21.70.1.3 sbi_ipi_event_destroy()	402

21.70.1.4 sbi_ipi_exit()	402
21.70.1.5 sbi_ipi_init()	403
21.70.1.6 sbi_ipi_process()	404
21.70.1.7 sbi_ipi_process_halt()	404
21.70.1.8 sbi_ipi_process_smode()	405
21.70.1.9 sbi_ipi_send()	405
21.70.1.10 sbi_ipi_send_halt()	406
21.70.1.11 sbi_ipi_send_many()	407
21.70.1.12 sbi_ipi_send_smode()	407
21.70.2 Variable Documentation	408
21.70.2.1 ipi_data_off	408
21.70.2.2 ipi_halt_event	408
21.70.2.3 ipi_halt_ops	409
21.70.2.4 ipi_ops_array	409
21.70.2.5 ipi_smode_event	409
21.70.2.6 ipi_smode_ops	409
21.71 lib/sbi/sbi_misaligned_ldst.c File Reference	410
21.71.1 Function Documentation	410
21.71.1.1 sbi_misaligned_load_handler()	410
21.71.1.2 sbi_misaligned_store_handler()	411
21.72 lib/sbi/sbi_scratch.c File Reference	412
21.72.1 Function Documentation	413
21.72.1.1 sbi_scratch_alloc_offset()	413
21.72.1.2 sbi_scratch_free_offset()	414
21.72.2 Variable Documentation	414
21.72.2.1 extra_lock	414
21.72.2.2 extra_offset	414
21.73 lib/sbi/sbi_string.c File Reference	414
21.73.1 Function Documentation	415
21.73.1.1 sbi_memchr()	415

21.73.1.2 sbi_memcmp()	415
21.73.1.3 sbi_memcpy()	415
21.73.1.4 sbi_memmove()	416
21.73.1.5 sbi_memset()	416
21.73.1.6 sbi_strchr()	416
21.73.1.7 sbi_strcmp()	416
21.73.1.8 sbi_strcpy()	417
21.73.1.9 sbi_strlen()	417
21.73.1.10 sbi_strncpy()	417
21.73.1.11 sbi_strnlen()	417
21.73.1.12 sbi_strrchr()	418
21.74 lib/sbi/sbi_system.c File Reference	418
21.74.1 Function Documentation	419
21.74.1.1 sbi_system_early_exit()	419
21.74.1.2 sbi_system_early_init()	419
21.74.1.3 sbi_system_final_exit()	420
21.74.1.4 sbi_system_final_init()	420
21.74.1.5 sbi_system_reboot()	421
21.74.1.6 sbi_system_shutdown()	421
21.75 lib/sbi/sbi_timer.c File Reference	422
21.75.1 Function Documentation	423
21.75.1.1 get_ticks()	423
21.75.1.2 sbi_timer_event_start()	423
21.75.1.3 sbi_timer_exit()	424
21.75.1.4 sbi_timer_get_delta()	424
21.75.1.5 sbi_timer_init()	425
21.75.1.6 sbi_timer_process()	425
21.75.1.7 sbi_timer_set_delta()	426
21.75.1.8 sbi_timer_set_delta_upper()	426
21.75.1.9 sbi_timer_value()	426

21.75.1.10	<code>sbi_timer_virt_value()</code>	427
21.75.2	Variable Documentation	427
21.75.2.1	<code>time_delta_off</code>	427
21.76	<code>lib/sbi/sbi_tlb.c</code> File Reference	428
21.76.1	Function Documentation	429
21.76.1.1	<code>__sbi_tlb_range_check()</code>	429
21.76.1.2	<code>sbi_tlb_entry_process()</code>	429
21.76.1.3	<code>sbi_tlb_flush_all()</code>	430
21.76.1.4	<code>sbi_tlb_hfence_gvma()</code>	430
21.76.1.5	<code>sbi_tlb_hfence_gvma_vmid()</code>	431
21.76.1.6	<code>sbi_tlb_hfence_vvma()</code>	431
21.76.1.7	<code>sbi_tlb_hfence_vvma_asid()</code>	432
21.76.1.8	<code>sbi_tlb_init()</code>	433
21.76.1.9	<code>sbi_tlb_local_flush()</code>	433
21.76.1.10	<code>sbi_tlb_process()</code>	434
21.76.1.11	<code>sbi_tlb_process_count()</code>	435
21.76.1.12	<code>sbi_tlb_request()</code>	435
21.76.1.13	<code>sbi_tlb_sfence_vma()</code>	436
21.76.1.14	<code>sbi_tlb_sfence_vma_asid()</code>	437
21.76.1.15	<code>sbi_tlb_sync()</code>	437
21.76.1.16	<code>sbi_tlb_update()</code>	438
21.76.1.17	<code>sbi_tlb_update_cb()</code>	438
21.76.2	Variable Documentation	439
21.76.2.1	<code>tlb_event</code>	439
21.76.2.2	<code>tlb_fifo_mem_off</code>	439
21.76.2.3	<code>tlb_fifo_off</code>	440
21.76.2.4	<code>tlb_ops</code>	440
21.76.2.5	<code>tlb_range_flush_limit</code>	440
21.76.2.6	<code>tlb_sync_off</code>	440
21.77	<code>lib/sbi/sbi_trap.c</code> File Reference	440

21.77.1 Function Documentation	441
21.77.1.1 sbi_trap_error()	441
21.77.1.2 sbi_trap_handler()	442
21.77.1.3 sbi_trap_redirect()	442
21.78lib/sbi/sbi_unpriv.c File Reference	443
21.78.1 Macro Definition Documentation	444
21.78.1.1 DEFINE_UNPRIVILEGED_LOAD_FUNCTION	444
21.78.1.2 DEFINE_UNPRIVILEGED_STORE_FUNCTION	445
21.78.2 Function Documentation	445
21.78.2.1 sbi_get_insn()	445
21.78.2.2 sbi_load_u64()	446
21.78.2.3 sbi_store_u64()	446
21.79lib/utlis/irqchip/plic.c File Reference	447
21.79.1 Macro Definition Documentation	448
21.79.1.1 PLIC_CONTEXT_BASE	448
21.79.1.2 PLIC_CONTEXT_STRIDE	448
21.79.1.3 PLIC_ENABLE_BASE	448
21.79.1.4 PLIC_ENABLE_STRIDE	448
21.79.1.5 PLIC_PENDING_BASE	448
21.79.1.6 PLIC_PRIORITY_BASE	448
21.79.2 Function Documentation	448
21.79.2.1 plic_cold_irqchip_init()	449
21.79.2.2 plic_fdt_fixup()	449
21.79.2.3 plic_set_ie()	449
21.79.2.4 plic_set_priority()	450
21.79.2.5 plic_set_thresh()	450
21.79.2.6 plic_warm_irqchip_init()	451
21.79.3 Variable Documentation	451
21.79.3.1 plic_base	451
21.79.3.2 plic_hart_count	451

21.79.3.3 plic_num_sources	451
21.80lib/utlis/libfdt/fdt.c File Reference	452
21.80.1 Function Documentation	453
21.80.1.1 check_block_()	453
21.80.1.2 check_off_()	453
21.80.1.3 fdt_check_header()	454
21.80.1.4 fdt_check_node_offset_()	454
21.80.1.5 fdt_check_prop_offset_()	455
21.80.1.6 fdt_find_string_()	456
21.80.1.7 fdt_first_subnode()	456
21.80.1.8 fdt_header_size_()	457
21.80.1.9 fdt_move()	458
21.80.1.10fdt_next_node()	458
21.80.1.11fdt_next_subnode()	459
21.80.1.12fdt_next_tag()	460
21.80.1.13fdt_offset_ptr()	461
21.80.1.14fdt_ro_probe_()	462
21.81lib/utlis/libfdt/fdt.h File Reference	463
21.81.1 Macro Definition Documentation	463
21.81.1.1 FDT_BEGIN_NODE	463
21.81.1.2 FDT_END	463
21.81.1.3 FDT_END_NODE	464
21.81.1.4 FDT_MAGIC	464
21.81.1.5 FDT_NOP	464
21.81.1.6 FDT_PROP	464
21.81.1.7 FDT_TAGSIZE	464
21.81.1.8 FDT_V16_SIZE	464
21.81.1.9 FDT_V17_SIZE	464
21.81.1.10FDT_V1_SIZE	465
21.81.1.11FDT_V2_SIZE	465

21.81.1.12	FDT_V3_SIZE	465
21.82	lib/utils/libfdt/fdt_addresses.c File Reference	465
21.82.1	Function Documentation	466
21.82.1.1	fdt_address_cells()	466
21.82.1.2	fdt_cells()	466
21.82.1.3	fdt_size_cells()	467
21.83	lib/utils/libfdt/fdt_empty_tree.c File Reference	467
21.83.1	Function Documentation	468
21.83.1.1	fdt_create_empty_tree()	468
21.84	lib/utils/libfdt/fdt_overlay.c File Reference	469
21.84.1	Function Documentation	470
21.84.1.1	fdt_overlay_apply()	470
21.84.1.2	get_path_len()	471
21.84.1.3	overlay_adjust_local_phandles()	472
21.84.1.4	overlay_adjust_node_phandles()	472
21.84.1.5	overlay_apply_node()	473
21.84.1.6	overlay_fixup_one_phandle()	474
21.84.1.7	overlay_fixup_phandle()	475
21.84.1.8	overlay_fixup_phandles()	475
21.84.1.9	overlay_get_target()	476
21.84.1.10	overlay_get_target_phandle()	477
21.84.1.11	overlay_merge()	477
21.84.1.12	overlay_phandle_add_offset()	478
21.84.1.13	overlay_symbol_update()	479
21.84.1.14	overlay_update_local_node_references()	480
21.84.1.15	overlay_update_local_references()	480
21.85	lib/utils/libfdt/fdt_ro.c File Reference	481
21.85.1	Function Documentation	483
21.85.1.1	fdt_check_full()	483
21.85.1.2	fdt_first_property_offset()	484

21.85.1.3 fdt_get_alias()	485
21.85.1.4 fdt_get_alias_namelen()	486
21.85.1.5 fdt_get_max_phandle()	486
21.85.1.6 fdt_get_mem_rsv()	487
21.85.1.7 fdt_get_name()	488
21.85.1.8 fdt_get_path()	489
21.85.1.9 fdt_get_phandle()	489
21.85.1.10 fdt_get_property()	490
21.85.1.11 fdt_get_property_by_offset()	491
21.85.1.12 fdt_get_property_by_offset_()	492
21.85.1.13 fdt_get_property_namelen()	492
21.85.1.14 fdt_get_property_namelen_()	493
21.85.1.15 fdt_get_string()	494
21.85.1.16 fdt_getprop()	495
21.85.1.17 fdt_getprop_by_offset()	496
21.85.1.18 fdt_getprop_namelen()	497
21.85.1.19 fdt_mem_rsv()	498
21.85.1.20 fdt_next_property_offset()	499
21.85.1.21 fdt_node_check_compatible()	500
21.85.1.22 fdt_node_depth()	501
21.85.1.23 fdt_node_offset_by_compatible()	502
21.85.1.24 fdt_node_offset_by_phandle()	503
21.85.1.25 fdt_node_offset_by_prop_value()	503
21.85.1.26 fdt_nodename_eq_()	504
21.85.1.27 fdt_num_mem_rsv()	505
21.85.1.28 fdt_parent_offset()	506
21.85.1.29 fdt_path_offset()	506
21.85.1.30 fdt_path_offset_namelen()	507
21.85.1.31 fdt_string()	508
21.85.1.32 fdt_string_eq_()	509

21.85.1.33	fdt_stringlist_contains()	509
21.85.1.34	fdt_stringlist_count()	510
21.85.1.35	fdt_stringlist_get()	511
21.85.1.36	fdt_stringlist_search()	511
21.85.1.37	fdt_subnode_offset()	512
21.85.1.38	fdt_subnode_offset_namelen()	513
21.85.1.39	fdt_supernode_atdepth_offset()	513
21.85.1.40	nextprop_()	514
21.86	lib/utils/libfdt/fdt_rw.c File Reference	515
21.86.1	Macro Definition Documentation	517
21.86.1.1	FDT_RW_PROBE	517
21.86.2	Function Documentation	517
21.86.2.1	fdt_add_mem_rsv()	517
21.86.2.2	fdt_add_property_()	518
21.86.2.3	fdt_add_subnode()	518
21.86.2.4	fdt_add_subnode_namelen()	519
21.86.2.5	fdt_appendprop()	519
21.86.2.6	fdt_blocks_misordered_()	520
21.86.2.7	fdt_data_size_()	521
21.86.2.8	fdt_del_mem_rsv()	521
21.86.2.9	fdt_del_node()	522
21.86.2.10	fdt_delprop()	522
21.86.2.11	fdt_find_add_string_()	523
21.86.2.12	fdt_open_into()	523
21.86.2.13	fdt_pack()	524
21.86.2.14	fdt_packblocks_()	524
21.86.2.15	fdt_resize_property_()	525
21.86.2.16	fdt_rw_probe_()	525
21.86.2.17	fdt_set_name()	526
21.86.2.18	fdt_setprop()	526

21.86.2.19 fdt_setprop_placeholder()	527
21.86.2.20 fdt_splice_()	527
21.86.2.21 fdt_splice_mem_rsv_()	528
21.86.2.22 fdt_splice_string_()	529
21.86.2.23 fdt_splice_struct_()	529
21.87 lib/utils/libfdt/fdt_strerror.c File Reference	530
21.87.1 Macro Definition Documentation	531
21.87.1.1 FDT_ERRTABENT	531
21.87.1.2 FDT_ERRTABSIZE	531
21.87.2 Function Documentation	531
21.87.2.1 fdt_strerror()	531
21.87.3 Variable Documentation	531
21.87.3.1 fdt_errtable	532
21.88 lib/utils/libfdt/fdt_sw.c File Reference	532
21.88.1 Macro Definition Documentation	533
21.88.1.1 FDT_SW_PROBE	533
21.88.1.2 FDT_SW_PROBE_MEMRSV	533
21.88.1.3 FDT_SW_PROBE_STRUCT	534
21.88.2 Function Documentation	534
21.88.2.1 fdt_add_reservemap_entry()	534
21.88.2.2 fdt_begin_node()	535
21.88.2.3 fdt_create()	535
21.88.2.4 fdt_end_node()	536
21.88.2.5 fdt_find_add_string_()	536
21.88.2.6 fdt_finish()	537
21.88.2.7 fdt_finish_reservemap()	538
21.88.2.8 fdt_grab_space_()	538
21.88.2.9 fdt_property()	539
21.88.2.10 fdt_property_placeholder()	540
21.88.2.11 fdt_resize()	540

21.88.2.12	fdt_sw_probe_()	541
21.88.2.13	fdt_sw_probe_memrsv_()	541
21.88.2.14	fdt_sw_probe_struct_()	541
21.89	lib/utils/libfdt/fdt_wip.c File Reference	542
21.89.1	Function Documentation	542
21.89.1.1	fdt_node_end_offset_()	543
21.89.1.2	fdt_nop_node_()	543
21.89.1.3	fdt_nop_property_()	544
21.89.1.4	fdt_nop_region_()	545
21.89.1.5	fdt_setprop_inplace_()	545
21.89.1.6	fdt_setprop_inplace_namelen_partial_()	546
21.90	lib/utils/libfdt/libfdt.h File Reference	546
21.90.1	Macro Definition Documentation	552
21.90.1.1	fdt_appendprop_string	552
21.90.1.2	fdt_boot_cpuid_phys	552
21.90.1.3	FDT_ERR_BADLAYOUT	553
21.90.1.4	FDT_ERR_BADMAGIC	553
21.90.1.5	FDT_ERR_BADNCELLS	553
21.90.1.6	FDT_ERR_BADOFFSET	553
21.90.1.7	FDT_ERR_BADOVERLAY	553
21.90.1.8	FDT_ERR_BADPATH	553
21.90.1.9	FDT_ERR_BADPHANDLE	553
21.90.1.10	FDT_ERR_BADSTATE	553
21.90.1.11	FDT_ERR_BADSTRUCTURE	554
21.90.1.12	FDT_ERR_BADVALUE	554
21.90.1.13	FDT_ERR_BADVERSION	554
21.90.1.14	FDT_ERR_EXISTS	554
21.90.1.15	FDT_ERR_INTERNAL	554
21.90.1.16	FDT_ERR_MAX	554
21.90.1.17	FDT_ERR_NOPHANDLES	554

21.90.1.18	fdt_err_nospace	554
21.90.1.19	fdt_err_notfound	555
21.90.1.20	fdt_err_truncated	555
21.90.1.21	fdt_first_supported_version	555
21.90.1.22	fdt_for_each_property_offset	555
21.90.1.23	fdt_for_each_subnode	556
21.90.1.24	fdt_get_header	556
21.90.1.25	fdt_last_comp_version	556
21.90.1.26	fdt_last_supported_version	556
21.90.1.27	fdt_magic	556
21.90.1.28	fdt_max_ncells	557
21.90.1.29	fdt_off_dt_strings	557
21.90.1.30	fdt_off_dt_struct	557
21.90.1.31	fdt_off_mem_rsvmap	557
21.90.1.32	fdt_property_string	557
21.90.1.33	fdt_set_hdr	557
21.90.1.34	fdt_setprop_empty	558
21.90.1.35	fdt_setprop_string	558
21.90.1.36	fdt_size_dt_strings	558
21.90.1.37	fdt_size_dt_struct	558
21.90.1.38	fdt_totalsize	558
21.90.1.39	fdt_version	558
21.90.2	Function Documentation	558
21.90.2.1	fdt32_id()	559
21.90.2.2	fdt64_id()	559
21.90.2.3	fdt_add_mem_rsv()	560
21.90.2.4	fdt_add_reservemap_entry()	561
21.90.2.5	fdt_add_subnode()	561
21.90.2.6	fdt_add_subnode_namelen()	562
21.90.2.7	fdt_address_cells()	562

21.90.2.8 fdt_appendprop()	563
21.90.2.9 fdt_appendprop_cell()	564
21.90.2.10 fdt_appendprop_u32()	564
21.90.2.11 fdt_appendprop_u64()	565
21.90.2.12 fdt_begin_node()	565
21.90.2.13 fdt_check_full()	566
21.90.2.14 fdt_check_header()	566
21.90.2.15 fdt_create()	567
21.90.2.16 fdt_create_empty_tree()	568
21.90.2.17 fdt_del_mem_rsv()	568
21.90.2.18 fdt_del_node()	569
21.90.2.19 fdt_delprop()	569
21.90.2.20 fdt_end_node()	570
21.90.2.21 fdt_finish()	570
21.90.2.22 fdt_finish_reservemap()	571
21.90.2.23 fdt_first_property_offset()	572
21.90.2.24 fdt_first_subnode()	573
21.90.2.25 fdt_get_alias()	573
21.90.2.26 fdt_get_alias_namelen()	574
21.90.2.27 fdt_get_max_phandle()	575
21.90.2.28 fdt_get_mem_rsv()	575
21.90.2.29 fdt_get_name()	576
21.90.2.30 fdt_get_path()	577
21.90.2.31 fdt_get_phandle()	578
21.90.2.32 fdt_get_property()	579
21.90.2.33 fdt_get_property_by_offset()	579
21.90.2.34 fdt_get_property_namelen()	580
21.90.2.35 fdt_get_property_w()	581
21.90.2.36 fdt_get_string()	581
21.90.2.37 fdt_getprop()	582

21.90.2.38	fdt_getprop_by_offset()	583
21.90.2.39	fdt_getprop_namelen()	584
21.90.2.40	fdt_getprop_namelen_w()	585
21.90.2.41	fdt_getprop_w()	586
21.90.2.42	fdt_header_size()	586
21.90.2.43	fdt_header_size_()	587
21.90.2.44	fdt_move()	587
21.90.2.45	fdt_next_node()	588
21.90.2.46	fdt_next_property_offset()	589
21.90.2.47	fdt_next_subnode()	590
21.90.2.48	fdt_next_tag()	591
21.90.2.49	fdt_node_check_compatible()	592
21.90.2.50	fdt_node_depth()	593
21.90.2.51	fdt_node_offset_by_compatible()	594
21.90.2.52	fdt_node_offset_by_phandle()	595
21.90.2.53	fdt_node_offset_by_prop_value()	595
21.90.2.54	fdt_nop_node()	596
21.90.2.55	fdt_nop_property()	597
21.90.2.56	fdt_num_mem_rsv()	598
21.90.2.57	fdt_offset_ptr()	598
21.90.2.58	fdt_offset_ptr_w()	599
21.90.2.59	fdt_open_into()	600
21.90.2.60	fdt_overlay_apply()	601
21.90.2.61	fdt_pack()	601
21.90.2.62	fdt_parent_offset()	602
21.90.2.63	fdt_path_offset()	602
21.90.2.64	fdt_path_offset_namelen()	603
21.90.2.65	fdt_property()	604
21.90.2.66	fdt_property_cell()	605
21.90.2.67	fdt_property_placeholder()	605

21.90.2.68	dt_property_u32()	606
21.90.2.69	dt_property_u64()	606
21.90.2.70	dt_resize()	607
21.90.2.71	dt_set_hdr() [1/10]	607
21.90.2.72	dt_set_hdr() [2/10]	607
21.90.2.73	dt_set_hdr() [3/10]	607
21.90.2.74	dt_set_hdr() [4/10]	607
21.90.2.75	dt_set_hdr() [5/10]	608
21.90.2.76	dt_set_hdr() [6/10]	608
21.90.2.77	dt_set_hdr() [7/10]	608
21.90.2.78	dt_set_hdr() [8/10]	608
21.90.2.79	dt_set_hdr() [9/10]	608
21.90.2.80	dt_set_hdr() [10/10]	608
21.90.2.81	dt_set_name()	609
21.90.2.82	dt_setprop()	609
21.90.2.83	dt_setprop_cell()	610
21.90.2.84	dt_setprop_inplace()	610
21.90.2.85	dt_setprop_inplace_cell()	611
21.90.2.86	dt_setprop_inplace_namelen_partial()	611
21.90.2.87	dt_setprop_inplace_u32()	612
21.90.2.88	dt_setprop_inplace_u64()	612
21.90.2.89	dt_setprop_placeholder()	613
21.90.2.90	dt_setprop_u32()	613
21.90.2.91	dt_setprop_u64()	614
21.90.2.92	dt_size_cells()	614
21.90.2.93	dt_strerror()	615
21.90.2.94	dt_string()	615
21.90.2.95	dt_stringlist_contains()	616
21.90.2.96	dt_stringlist_count()	616
21.90.2.97	dt_stringlist_get()	617

21.90.2.98	fdt_stringlist_search()	618
21.90.2.99	fdt_subnode_offset()	618
21.90.2.100	fdt_subnode_offset_namelen()	619
21.90.2.101	fdt_supernode_atdepth_offset()	620
21.91	lib/utils/libfdt/libfdt_env.h File Reference	620
21.91.1	Macro Definition Documentation	622
21.91.1.1	CPU_TO_FDT16	622
21.91.1.2	CPU_TO_FDT32	622
21.91.1.3	CPU_TO_FDT64	622
21.91.1.4	EXTRACT_BYTE	623
21.91.1.5	FDT_BITWISE	623
21.91.1.6	FDT_FORCE	623
21.91.1.7	INT_MAX	623
21.91.1.8	memchr	623
21.91.1.9	memcmp	623
21.91.1.10	memcpy	623
21.91.1.11	memmove	624
21.91.1.12	memset	624
21.91.1.13	strchr	624
21.91.1.14	strcmp	624
21.91.1.15	strcpy	624
21.91.1.16	strlen	624
21.91.1.17	strnlen	624
21.91.1.18	strrchr	624
21.91.1.19	UINT_MAX	625
21.91.2	Typedef Documentation	625
21.91.2.1	fdt16_t	625
21.91.2.2	fdt32_t	625
21.91.2.3	fdt64_t	625
21.91.3	Function Documentation	625

21.91.3.1	cpu_to_fdt16()	625
21.91.3.2	cpu_to_fdt32()	626
21.91.3.3	cpu_to_fdt64()	626
21.91.3.4	fdt16_to_cpu()	627
21.91.3.5	fdt32_to_cpu()	627
21.91.3.6	fdt64_to_cpu()	627
21.92	lib/utls/libfdt/libfdt_internal.h File Reference	628
21.92.1	Macro Definition Documentation	628
21.92.1.1	FDT_ALIGN	629
21.92.1.2	FDT_RO_PROBE	629
21.92.1.3	FDT_SW_MAGIC	629
21.92.1.4	FDT_TAGALIGN	629
21.92.2	Function Documentation	629
21.92.2.1	fdt_check_node_offset()	630
21.92.2.2	fdt_check_prop_offset()	630
21.92.2.3	fdt_find_string()	631
21.92.2.4	fdt_mem_rsv()	632
21.92.2.5	fdt_mem_rsv_w()	632
21.92.2.6	fdt_node_end_offset()	633
21.92.2.7	fdt_offset_ptr()	633
21.92.2.8	fdt_offset_ptr_w()	634
21.92.2.9	fdt_ro_probe()	634
21.93	lib/utls/serial/sifive-uart.c File Reference	635
21.93.1	Macro Definition Documentation	636
21.93.1.1	UART_REG_DIV	636
21.93.1.2	UART_REG_IE	636
21.93.1.3	UART_REG_IP	636
21.93.1.4	UART_REG_RXCTRL	637
21.93.1.5	UART_REG_RXFIFO	637
21.93.1.6	UART_REG_TXCTRL	637

21.93.1.7 UART_REG_TXFIFO	637
21.93.1.8 UART_RXCTRL_RXEN	637
21.93.1.9 UART_RXFIFO_DATA	637
21.93.1.10 UART_RXFIFO_EMPTY	637
21.93.1.11 UART_TXCTRL_TXEN	637
21.93.1.12 UART_TXFIFO_FULL	638
21.93.2 Function Documentation	638
21.93.2.1 get_reg()	638
21.93.2.2 set_reg()	638
21.93.2.3 sifive_uart_getc()	639
21.93.2.4 sifive_uart_init()	639
21.93.2.5 sifive_uart_putc()	639
21.93.2.6 uart_min_clk_divisor()	640
21.93.3 Variable Documentation	640
21.93.3.1 uart_base	640
21.93.3.2 uart_baudrate	641
21.93.3.3 uart_in_freq	641
21.94lib/utils/serial/uart8250.c File Reference	641
21.94.1 Macro Definition Documentation	642
21.94.1.1 UART_DLL_OFFSET	642
21.94.1.2 UART_DLM_OFFSET	642
21.94.1.3 UART_FCR_OFFSET	642
21.94.1.4 UART_IER_OFFSET	643
21.94.1.5 UART_IIR_OFFSET	643
21.94.1.6 UART_LCR_OFFSET	643
21.94.1.7 UART_LSR_BI	643
21.94.1.8 UART_LSR_BRK_ERROR_BITS	643
21.94.1.9 UART_LSR_DR	643
21.94.1.10 UART_LSR_FE	643
21.94.1.11 UART_LSR_FIFOE	643

21.94.1.12	UART_LSR_OE	644
21.94.1.13	UART_LSR_OFFSET	644
21.94.1.14	UART_LSR_PE	644
21.94.1.15	UART_LSR_TEMT	644
21.94.1.16	UART_LSR_THRE	644
21.94.1.17	UART_MCR_OFFSET	644
21.94.1.18	UART_MDR1_OFFSET	644
21.94.1.19	UART_MSR_OFFSET	644
21.94.1.20	UART_RBR_OFFSET	645
21.94.1.21	UART_SCR_OFFSET	645
21.94.1.22	UART_THR_OFFSET	645
21.94.2	Function Documentation	645
21.94.2.1	get_reg()	645
21.94.2.2	set_reg()	646
21.94.2.3	uart8250_getc()	646
21.94.2.4	uart8250_init()	646
21.94.2.5	uart8250_putc()	647
21.94.3	Variable Documentation	647
21.94.3.1	uart8250_base	647
21.94.3.2	uart8250_baudrate	647
21.94.3.3	uart8250_in_freq	648
21.94.3.4	uart8250_reg_shift	648
21.94.3.5	uart8250_reg_width	648
21.95	lib/utlis/sys/clint.c File Reference	648
21.95.1	Function Documentation	649
21.95.1.1	clint_cold_ipi_init()	649
21.95.1.2	clint_cold_timer_init()	649
21.95.1.3	clint_ipi_clear()	650
21.95.1.4	clint_ipi_send()	650
21.95.1.5	clint_time_rd32()	650

21.95.1.6 clint_time_rd64()	650
21.95.1.7 clint_time_wr32()	651
21.95.1.8 clint_time_wr64()	651
21.95.1.9 clint_timer_event_start()	651
21.95.1.10 clint_timer_event_stop()	652
21.95.1.11 clint_timer_value()	652
21.95.1.12 clint_warm_ipi_init()	652
21.95.1.13 clint_warm_timer_init()	653
21.95.2 Variable Documentation	653
21.95.2.1 clint_ipi	653
21.95.2.2 clint_ipi_base	653
21.95.2.3 clint_ipi_hart_count	653
21.95.2.4 clint_time_base	653
21.95.2.5 clint_time_cmp	654
21.95.2.6 clint_time_hart_count	654
21.95.2.7 clint_time_rd	654
21.95.2.8 clint_time_val	654
21.95.2.9 clint_time_wr	654
21.96 lib/utis/sys/htif.c File Reference	654
21.96.1 Macro Definition Documentation	655
21.96.1.1 FROMHOST_CMD	655
21.96.1.2 FROMHOST_DATA	656
21.96.1.3 FROMHOST_DEV	656
21.96.1.4 HTIF_CMD_BITS	656
21.96.1.5 HTIF_CMD_MASK	656
21.96.1.6 HTIF_CMD_SHIFT	656
21.96.1.7 HTIF_CONSOLE_CMD_GETC	656
21.96.1.8 HTIF_CONSOLE_CMD_PUTC	656
21.96.1.9 HTIF_DATA_BITS	657
21.96.1.10 HTIF_DATA_MASK	657

21.96.1.11	HTIF_DATA_SHIFT	657
21.96.1.12	HTIF_DEV_BITS	657
21.96.1.13	HTIF_DEV_CONSOLE	657
21.96.1.14	HTIF_DEV_MASK	657
21.96.1.15	HTIF_DEV_SHIFT	657
21.96.1.16	HTIF_DEV_SYSTEM	657
21.96.1.17	PK_SYS_write	658
21.96.1.18	TOHOST_CMD	658
21.96.2	Function Documentation	658
21.96.2.1	__attribute__()	658
21.96.2.2	__check_fromhost()	658
21.96.2.3	__set_tohost()	659
21.96.2.4	htif_getc()	659
21.96.2.5	htif_putc()	660
21.96.2.6	htif_system_down()	660
21.96.3	Variable Documentation	660
21.96.3.1	htif_console_buf	660
21.96.3.2	htif_lock	661
21.97	README.md File Reference	661
	Index	663

Chapter 1

README

Copyright (c) 2019 Western Digital Corporation or its affiliates and other contributors.

RISC-V Open Source Supervisor Binary Interface (OpenSBI)

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](#).

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 *qemu/virt*.

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 versa.

License

OpenSBI 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.

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 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 *libsbis.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 *libsbis.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 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 `-mcmmodel`.

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 OpenSBI data types based on external firmware data type binding. For example, *bool* is declared as *int* in [sbi_types.h](#). However in EDK2 build system, *bool* is declared as *BOOLEAN* which is defined as *unsigned char* data type.

External firmware can define **OPENSBI_EXTERNAL_SBI_TYPES** in CFLAGS and specify it to the header file maintained in its code tree. However, the external build system has to address the additional include directory for the external header file based on its own build system. For example, `-D***OPENSBI_EXTERNAL_SBI_TYPES***=OpensbiTypes.h` Above tells [sbi_types.h](#) to refer to *OpensbiTypes.h* instead of using original definitions of data types.

Chapter 5

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 6

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 7

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 8

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_PAYLOAD_ALIGN* are defined, *FW_PAYLOAD_OFFSET* is used and *FW_PAYLOAD_ALIGN* is ignored.
- **FW_PAYLOAD_PATH** - Path to the image file of the next booting stage binary. If this option is not provided then a simple test payload is automatically generated and used as a payload. This test payload executes an infinite `while (1)` loop after printing a message on the platform console.
- **FW_PAYLOAD_FDT_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 and no internal device tree file is specified by the platform (c.f. *FW_PAYLOAD_FDT*), then the firmware will expect the FDT to be passed as an argument by the prior booting stage.
- **FW_PAYLOAD_FDT** - Path to an internal flattened device tree binary file defined by the platform code. The file name must match the DTB file name specified in the platform *objects.mk* file with the *platform-dtb-y* entry. This option results in *FW_PAYLOAD_FDT_PATH* to be automatically set. Specifying *FW_PAYLOAD_FDT_PATH* on the `make` command line disables *FW_PAYLOAD_FDT* and the command line specified device tree binary file is used for building the final firmware.
- **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.

The *kendryte/k210* platform also enables a build of a *FW_PAYLOAD* using an internally defined device tree file (*FW_PAYLOAD_FDT*).

Chapter 9

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 10

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.

The current stable upstream code of U-Boot does not yet include all patches necessary to fully support OpenSBI. To use U-Boot as an OpenSBI payload, the following out-of-tree patch series must be applied to the upstream U-Boot source code:

HiFive Unleashed support for U-Boot

<https://lists.denx.de/pipermail/u-boot/2019-February/358058.html>

This patch series enables a single CPU to execute U-Boot. As a result, the next stage boot code such as a Linux kernel can also only execute on a single CPU. U-Boot SMP support for RISC-V can be enabled with the following additional patches:

<https://lists.denx.de/pipermail/u-boot/2019-February/358393.html>

Building and Generating U-Boot images

Please refer to the U-Boot build documentation for detailed instructions on how to build U-Boot images.

Once U-Boot images are built, the Linux kernel image needs to be converted into a format that U-Boot understands:

```
<uboot-dir>/tools/mkimage -A riscv -O linux -T kernel -C none -a 0x80200000 -e 0x80200000 -n Linux -d \  
    <linux_build_directory>arch/riscv/boot/Image \  
    <linux_build_directory>/arch/riscv/boot/uImage
```

Copy the ulmage to your tftpboot server path if network boot is required.

Chapter 11

Andes AE350 SoC Platform

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

To build platform specific library and firmwares, provide the *PLATFORM=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 12

Ariane FPGA SoC Platform

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

The FPGA SoC currently contains the following peripherals:

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

To build platform specific library and firmwares, provide the *PLATFORM=ariane-fpga* 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=ariane-fpga 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 13

OpenSBI Supported Platforms

OpenSBI currently supports the following virtual and hardware platforms:

- **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.
- **Andes AE350 SoC:** Platform support for the Andes's SoC (AE350).
- **T-HEAD C910:** Platform support for the T-HEAD C910 Processor.
- **Spike:** Platform support for the Spike emulator.

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 14

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=qemu/virt* 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=qemu/virt
```

Run:

```
qemu-system-riscv64 -M virt -m 256M -nographic \  
-kernel build/platform/qemu/virt/firmware/fw_payload.elf
```

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=qemu/virt FW_PAYLOAD_PATH=<uboot_build_directory>/u-boot.bin
```

Run:

```
qemu-system-riscv64 -M virt -m 256M -nographic \
  -kernel build/platform/qemu/virt/firmware/fw_payload.elf
```

or

```
qemu-system-riscv64 -M virt -m 256M -nographic \
  -kernel build/platform/qemu/virt/firmware/fw_jump.elf \
  -device loader,file=<uboot_build_directory>/u-boot.bin,addr=0x80200000
```

Linux Kernel Payload

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

Build:

```
make PLATFORM=qemu/virt FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```

Run:

```
qemu-system-riscv64 -M virt -m 256M -nographic \
  -kernel build/platform/qemu/virt/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 \
  -kernel build/platform/qemu/virt/firmware/fw_jump.elf \
  -device loader,file=<linux_build_directory>/arch/riscv/boot/Image,addr=0x80200000 \
  -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=qemu/virt
```

Run:

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

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=qemu/virt FW_PAYLOAD_PATH=<uboot_build_directory>/u-boot.bin
```

Run:

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

or

```
qemu-system-riscv32 -M virt -m 256M -nographic \  
-kernel build/platform/qemu/virt/firmware/fw_jump.elf \  
-device loader,file=<uboot_build_directory>/u-boot.bin,addr=0x80400000
```

Linux Kernel Payload

Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/rv32_defconfig* (kernel 5.1 and newer) respectively using *arch/riscv/configs/defconfig* plus setting `CONFIG_ARCH_RV32I=y` (kernel 5.0 and older).

Build:

```
make PLATFORM=qemu/virt FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```

Run:

```
qemu-system-riscv32 -M virt -m 256M -nographic \  
-kernel build/platform/qemu/virt/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 \  
-kernel build/platform/qemu/virt/firmware/fw_jump.elf \  
-device loader,file=<linux_build_directory>/arch/riscv/boot/Image,addr=0x80400000 \  
-drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \  
-device virtio-blk-device,drive=hd0 \  
-append "root=/dev/vda rw console=ttyS0"
```


Chapter 15

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

As hart0 in the FU540 doesn't have an MMU, only harts 1-4 boot by default. A hart mask i.e. *FU540_ENABLED_HART_MASK* compile time option is provided to select any other hart for booting. Please keep in mind that this is not a generic option and it can only be specified for FU540 platform in the following way:

```
make PLATFORM=sifive/fu540 FW_PAYLOAD_PATH=Image FU540_ENABLED_HART_MASK=0x02
```

This will let the board boot only hart1 instead of default 1-4.

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 available at [U-Boot](#).

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

U-Boot & Linux Kernel as a single payload

A single monolithic image containing both U-Boot & Linux can also be used if network boot setup is not available.

1. Create a temporary image with `u-boot-dtb.bin` as the first payload. The command-line example here assumes that U-Boot was compiled using `sifive_fu540_defconfig` configuration.

```
dd if=~/.workspace/u-boot-riscv/u-boot-dtb.bin of=/tmp/temp.bin bs=1M
```

2. Append the Linux Kernel image.

```
dd if=<linux_build_directory>/arch/riscv/boot/Image of=/tmp/temp.bin bs=1M seek=4
```

3. Compile OpenSBI with `temp.bin` (generated in step 2) as payload.

```
make PLATFORM=sifive/fu540 FW_PAYLOAD_PATH=/tmp/temp.bin
or
(For U-Boot which follows Linux v5.2 (or higher) DT bindings)
make PLATFORM=sifive/fu540 FW_PAYLOAD_PATH=/tmp/temp.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)

8. Now boot into Linux.

```
booti ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r}
or
(If ramdisk is not loaded from network)
booti ${kernel_addr_r} - ${fdt_addr_r}
```

U-Boot & Linux Kernel as a single payload

At U-Boot prompt execute the following boot command to boot Linux.

```
booti ${kernel_addr_r} - ${fdt_addr_r}
```

QEMU Specific Instructions

If you want to test OpenSBI with QEMU 'sifive_u' machine, please follow the same instructions above, with the exception of not passing FW_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 16

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=spike* 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=spike
```

Run:

```
spike build/platform/spike/firmware/fw_payload.elf
```

Linux Kernel Payload

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

Build:

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

Run:

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

Execution on QEMU RISC-V 64-bit

No Payload Case

Build:

```
make PLATFORM=spike
```

Run:

```
qemu-system-riscv64 -M spike -m 256M -nographic \  
-kernel build/platform/spike/firmware/fw_payload.elf
```

Linux Kernel Payload

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

Build:

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

Run:

```
qemu-system-riscv64 -M spike -m 256M -nographic \  
-kernel build/platform/spike/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/spike/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 17

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_PAYLOAD_OFFSET. The kernel image will be embedded in the OPENSBI firmware binary, T-head will directly boot into Linux after OpenSBI.

Chapter 18

Data Structure Index

18.1 Data Structures

Here are the data structures with brief descriptions:

atomic_t	45
fdt_header	45
fdt_node_header	47
fdt_property	48
fdt_reserve_entry	49
fw_dynamic_info	49
sbi_dlist	51
sbi_ecall_extension	52
sbi_fifo	53
sbi_ipi_event_ops	54
sbi_platform	55
sbi_platform_operations	57
sbi_scratch	61
sbi_tlb_info	63
sbi_trap_info	65
sbi_trap_regs	66
spinlock_t	71

Chapter 19

File Index

19.1 File List

Here is a list of all files with brief descriptions:

include/sbi/fw_dynamic.h	74
include/sbi/riscv_asm.h	77
include/sbi/riscv_atomic.h	86
include/sbi/riscv_barrier.h	90
include/sbi/riscv_encoding.h	93
include/sbi/riscv_fp.h	164
include/sbi/riscv_io.h	165
include/sbi/riscv_locks.h	172
include/sbi/sbi_bitops.h	177
include/sbi/sbi_bits.h	180
include/sbi/sbi_console.h	181
include/sbi/sbi_const.h	186
include/sbi/sbi_ecall.h	189
include/sbi/sbi_ecall_interface.h	194
include/sbi/sbi_emulate_csr.h	200
include/sbi/sbi_error.h	202
include/sbi/sbi_fifo.h	205
include/sbi/sbi_hart.h	211
include/sbi/sbi_hfence.h	220
include/sbi/sbi_illegal_insn.h	223
include/sbi/sbi_init.h	225
include/sbi/sbi_ipi.h	229
include/sbi/sbi_list.h	236
include/sbi/sbi_misaligned_ldst.h	243
include/sbi/sbi_platform.h	246
include/sbi/sbi_scratch.h	270
include/sbi/sbi_string.h	276
include/sbi/sbi_system.h	280
include/sbi/sbi_timer.h	284
include/sbi/sbi_tlb.h	289
include/sbi/sbi_trap.h	293
include/sbi/sbi_types.h	302
include/sbi/sbi_unpriv.h	309
include/sbi/sbi_version.h	311
include/sbi_utils/irqchip/plic.h	312

include/sbi_utils/serial/sifive-uart.h	315
include/sbi_utils/serial/uart8250.h	317
include/sbi_utils/sys/clint.h	320
include/sbi_utils/sys/htif.h	324
lib/sbi/riscv_asm.c	326
lib/sbi/riscv_atomic.c	330
lib/sbi/riscv_locks.c	337
lib/sbi/sbi_console.c	342
lib/sbi/sbi_ecall.c	353
lib/sbi/sbi_ecall_base.c	358
lib/sbi/sbi_ecall_legacy.c	360
lib/sbi/sbi_ecall_replace.c	362
lib/sbi/sbi_ecall_vendor.c	366
lib/sbi/sbi_emulate_csr.c	368
lib/sbi/sbi_fifo.c	370
lib/sbi/sbi_hart.c	378
lib/sbi/sbi_illegal_insn.c	390
lib/sbi/sbi_init.c	393
lib/sbi/sbi_ipi.c	400
lib/sbi/sbi_misaligned_ldst.c	410
lib/sbi/sbi_scratch.c	412
lib/sbi/sbi_string.c	414
lib/sbi/sbi_system.c	418
lib/sbi/sbi_timer.c	422
lib/sbi/sbi_tlb.c	428
lib/sbi/sbi_trap.c	440
lib/sbi/sbi_unpriv.c	443
lib/utis/irqchip/plic.c	447
lib/utis/libfdt/fdt.c	452
lib/utis/libfdt/fdt.h	463
lib/utis/libfdt/fdt_addresses.c	465
lib/utis/libfdt/fdt_empty_tree.c	467
lib/utis/libfdt/fdt_overlay.c	469
lib/utis/libfdt/fdt_ro.c	481
lib/utis/libfdt/fdt_rw.c	515
lib/utis/libfdt/fdt_strerror.c	530
lib/utis/libfdt/fdt_sw.c	532
lib/utis/libfdt/fdt_wip.c	542
lib/utis/libfdt/libfdt.h	546
lib/utis/libfdt/libfdt_env.h	620
lib/utis/libfdt/libfdt_internal.h	628
lib/utis/serial/sifive-uart.c	635
lib/utis/serial/uart8250.c	641
lib/utis/sys/clint.c	648
lib/utis/sys/htif.c	654

Chapter 20

Data Structure Documentation

20.1 `atomic_t` Struct Reference

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

Data Fields

- volatile long [counter](#)

20.1.1 Field Documentation

20.1.1.1 `counter`

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

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

- `include/sbi/riscv_atomic.h`

20.2 `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](#)

20.2.1 Field Documentation

20.2.1.1 boot_cpuid_phys

[fdt32_t](#) fdt_header::boot_cpuid_phys

20.2.1.2 last_comp_version

[fdt32_t](#) fdt_header::last_comp_version

20.2.1.3 magic

[fdt32_t](#) fdt_header::magic

20.2.1.4 off_dt_strings

[fdt32_t](#) fdt_header::off_dt_strings

20.2.1.5 off_dt_struct

[fdt32_t](#) fdt_header::off_dt_struct

20.2.1.6 off_mem_rsvmap

`fdt32_t fdt_header::off_mem_rsvmap`

20.2.1.7 size_dt_strings

`fdt32_t fdt_header::size_dt_strings`

20.2.1.8 size_dt_struct

`fdt32_t fdt_header::size_dt_struct`

20.2.1.9 totalsize

`fdt32_t fdt_header::totalsize`

20.2.1.10 version

`fdt32_t fdt_header::version`

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

- `lib/utils/libfdt/fdt.h`

20.3 fdt_node_header Struct Reference

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

Data Fields

- `fdt32_t tag`
- `char name [0]`

20.3.1 Field Documentation

20.3.1.1 name

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

20.3.1.2 tag

```
fdt32_t fdt_node_header::tag
```

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

- [lib/utils/libfdt/fdt.h](#)

20.4 fdt_property Struct Reference

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

Data Fields

- [fdt32_t tag](#)
- [fdt32_t len](#)
- [fdt32_t nameoff](#)
- char [data](#) [0]

20.4.1 Field Documentation

20.4.1.1 data

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

20.4.1.2 len

```
fdt32_t fdt_property::len
```


20.4.1.3 nameoff

```
fdt32_t fdt_property::nameoff
```

20.4.1.4 tag

```
fdt32_t fdt_property::tag
```

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

- [lib/utils/libfdt/fdt.h](#)

20.5 fdt_reserve_entry Struct Reference

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

Data Fields

- [fdt64_t address](#)
- [fdt64_t size](#)

20.5.1 Field Documentation

20.5.1.1 address

```
fdt64_t fdt_reserve_entry::address
```

20.5.1.2 size

```
fdt64_t fdt_reserve_entry::size
```

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

- [lib/utils/libfdt/fdt.h](#)

20.6 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](#)

20.6.1 Detailed Description

Representation dynamic info passed by previous booting stage

20.6.2 Field Documentation

20.6.2.1 [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.

20.6.2.2 [magic](#)

```
unsigned long fw_dynamic_info::magic
```

Info magic

20.6.2.3 [next_addr](#)

```
unsigned long fw_dynamic_info::next_addr
```

Next booting stage address

20.6.2.4 [next_mode](#)

```
unsigned long fw_dynamic_info::next_mode
```

Next booting stage mode

20.6.2.5 options

```
unsigned long fw_dynamic_info::options
```

Options for OpenSBI library

20.6.2.6 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](#)

20.7 sbi_dlist Struct Reference

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

Data Fields

- struct [sbi_dlist](#) * [next](#)
- struct [sbi_dlist](#) * [prev](#)

20.7.1 Field Documentation

20.7.1.1 next

```
struct sbi\_dlist* sbi_dlist::next
```

20.7.1.2 prev

```
struct sbi\_dlist * sbi_dlist::prev
```

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

- [include/sbi/sbi_list.h](#)

20.8 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](#))(struct [sbi_scratch](#) *scratch, unsigned long extid, unsigned long *out_val)
- int(* [handle](#))(struct [sbi_scratch](#) *scratch, unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct [sbi_trap_info](#) *out_trap)

20.8.1 Field Documentation

20.8.1.1 [extid_end](#)

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

20.8.1.2 [extid_start](#)

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

20.8.1.3 [handle](#)

```
int(* sbi_ecall_extension::handle) (struct sbi\_scratch *scratch, unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct sbi\_trap\_info *out_trap)
```

20.8.1.4 [head](#)

```
struct sbi\_dlist sbi_ecall_extension::head
```

20.8.1.5 probe

```
int (* sbi_ecall_extension::probe) (struct sbi\_scratch *scratch, unsigned long extid, unsigned long *out_val)
```

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

- [include/sbi/sbi_ecall.h](#)

20.9 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

20.9.1 Field Documentation

20.9.1.1 avail

```
u16 sbi_fifo::avail
```

20.9.1.2 entry_size

```
u16 sbi_fifo::entry_size
```

20.9.1.3 num_entries

```
u16 sbi_fifo::num_entries
```

20.9.1.4 qlock

```
spinlock_t sbi_fifo::qlock
```

20.9.1.5 queue

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

20.9.1.6 tail

```
u16 sbi_fifo::tail
```

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

- include/sbi/[sbi_fifo.h](#)

20.10 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)

20.10.1 Detailed Description

IPI event operations or callbacks

20.10.2 Field Documentation

20.10.2.1 name

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

Name of the IPI event operations

20.10.2.2 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.

20.10.2.3 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.

20.10.2.4 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](#)

20.11 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](#)
- [u64 disabled_hart_mask](#)
- [unsigned long platform_ops_addr](#)
- [unsigned long firmware_context](#)

20.11.1 Detailed Description

Representation of a platform

20.11.2 Field Documentation

20.11.2.1 disabled_hart_mask

```
u64 sbi_platform::disabled_hart_mask
```

Mask representing the set of disabled HARTs

20.11.2.2 features

```
u64 sbi_platform::features
```

Supported features

20.11.2.3 firmware_context

```
unsigned long sbi_platform::firmware_context
```

Pointer to system firmware specific context

20.11.2.4 hart_count

```
u32 sbi_platform::hart_count
```

Total number of HARTs

20.11.2.5 hart_stack_size

```
u32 sbi_platform::hart_stack_size
```

Per-HART stack size for exception/interrupt handling

20.11.2.6 name

```
char sbi_platform::name[64]
```

Name of the platform

20.11.2.7 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

20.11.2.8 platform_ops_addr

```
unsigned long sbi_platform::platform_ops_addr
```

Pointer to sbi platform operations

20.11.2.9 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`

20.12 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(* system_reboot)(u32 type)`
- `int(* system_shutdown)(u32 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)`

20.12.1 Detailed Description

Platform functions

20.12.2 Field Documentation

20.12.2.1 console_getc

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

Read a character from the platform console input

20.12.2.2 console_init

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

Initialize the platform console

20.12.2.3 console_putc

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

Write a character to the platform console output

20.12.2.4 early_exit

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

Platform early exit

20.12.2.5 early_init

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

Platform early initialization

20.12.2.6 final_exit

```
void (* sbi_platform_operations::final_exit) (void)
```

Platform final exit

20.12.2.7 final_init

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

Platform final initialization

20.12.2.8 get_tlbr_flush_limit

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

Get tlb flush limit value

20.12.2.9 ipi_clear

```
void (* sbi_platform_operations::ipi_clear) (u32 target_hart)
```

Clear IPI for a target HART

20.12.2.10 ipi_exit

```
void (* sbi_platform_operations::ipi_exit) (void)
```

Exit IPI for current HART

20.12.2.11 ipi_init

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

Initialize IPI for current HART

20.12.2.12 ipi_send

```
void (* sbi_platform_operations::ipi_send) (u32 target_hart)
```

Send IPI to a target HART

20.12.2.13 irqchip_exit

```
void (* sbi_platform_operations::irqchip_exit) (void)
```

Exit the platform interrupt controller for current HART

20.12.2.14 irqchip_init

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

Initialize the platform interrupt controller for current HART

20.12.2.15 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.

20.12.2.16 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.

20.12.2.17 pmp_region_count

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

Get number of PMP regions for given HART

20.12.2.18 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

20.12.2.19 system_reboot

```
int (* sbi_platform_operations::system_reboot) (u32 type)
```

Reboot the platform

20.12.2.20 system_shutdown

```
int (* sbi_platform_operations::system_shutdown) (u32 type)
```

Shutdown or poweroff the platform

20.12.2.21 timer_event_start

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

Start platform timer event for current HART

20.12.2.22 timer_event_stop

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

Stop platform timer event for current HART

20.12.2.23 timer_exit

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

Exit platform timer for current HART

20.12.2.24 timer_init

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

Initialize platform timer for current HART

20.12.2.25 timer_value

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

Get platform timer value

20.12.2.26 vendor_ext_check

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

platform specific SBI extension implementation probe function

20.12.2.27 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](#)

20.13 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](#)

20.13.1 Detailed Description

Representation of per-HART scratch space

20.13.2 Field Documentation

20.13.2.1 fw_size

```
unsigned long sbi_scratch::fw_size
```

Size (in bytes) of firmware linked to OpenSBI library

20.13.2.2 fw_start

```
unsigned long sbi_scratch::fw_start
```

Start (or base) address of firmware linked to OpenSBI library

20.13.2.3 hartid_to_scratch

```
unsigned long sbi_scratch::hartid_to_scratch
```

Address of HART ID to [sbi_scratch](#) conversion function

20.13.2.4 next_addr

```
unsigned long sbi_scratch::next_addr
```

Address of next booting stage for this HART

20.13.2.5 next_arg1

```
unsigned long sbi_scratch::next_arg1
```

Arg1 (or 'a1' register) of next booting stage for this HART

20.13.2.6 next_mode

```
unsigned long sbi_scratch::next_mode
```

Privileged mode of next booting stage for this HART

20.13.2.7 options

```
unsigned long sbi_scratch::options
```

Options for OpenSBI library

20.13.2.8 platform_addr

```
unsigned long sbi_scratch::platform_addr
```

Address of [sbi_platform](#)

20.13.2.9 tmp0

```
unsigned long sbi_scratch::tmp0
```

Temporary storage

20.13.2.10 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](#)

20.14 sbi_tlb_info Struct Reference

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

Data Fields

- unsigned long [start](#)
- unsigned long [size](#)
- unsigned long [asid](#)
- unsigned long [type](#)
- unsigned long [shart_mask](#)

20.14.1 Field Documentation

20.14.1.1 asid

`unsigned long sbi_tlb_info::asid`

20.14.1.2 shart_mask

`unsigned long sbi_tlb_info::shart_mask`

20.14.1.3 size

`unsigned long sbi_tlb_info::size`

20.14.1.4 start

`unsigned long sbi_tlb_info::start`

20.14.1.5 type

`unsigned long sbi_tlb_info::type`

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

- `include/sbi/sbi_tlb.h`

20.15 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](#)

20.15.1 Detailed Description

Representation of trap details

20.15.2 Field Documentation

20.15.2.1 cause

```
unsigned long sbi_trap_info::cause
```

cause Trap exception cause

20.15.2.2 epc

```
unsigned long sbi_trap_info::epc
```

epc Trap program counter

20.15.2.3 tinst

```
unsigned long sbi_trap_info::tinst
```

tinst Trap instruction

20.15.2.4 tval

```
unsigned long sbi_trap_info::tval
```

tval Trap value

20.15.2.5 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](#)

20.16 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](#)

20.16.1 Detailed Description

Representation of register state at time of trap/interrupt

20.16.2 Field Documentation

20.16.2.1 a0

```
unsigned long sbi_trap_regs::a0
```

a0 register state

20.16.2.2 a1

```
unsigned long sbi_trap_regs::a1
```

a1 register state

20.16.2.3 a2

```
unsigned long sbi_trap_regs::a2
```

a2 register state

20.16.2.4 a3

```
unsigned long sbi_trap_regs::a3
```

a3 register state

20.16.2.5 a4

```
unsigned long sbi_trap_regs::a4
```

a4 register state

20.16.2.6 a5

```
unsigned long sbi_trap_regs::a5
```

a5 register state

20.16.2.7 a6

```
unsigned long sbi_trap_regs::a6
```

a6 register state

20.16.2.8 a7

```
unsigned long sbi_trap_regs::a7
```

a7 register state

20.16.2.9 gp

```
unsigned long sbi_trap_regs::gp
```

gp register state

20.16.2.10 mepc

```
unsigned long sbi_trap_regs::mepc
```

mepc register state

20.16.2.11 mstatus

```
unsigned long sbi_trap_regs::mstatus
```

mstatus register state

20.16.2.12 mstatusH

```
unsigned long sbi_trap_regs::mstatusH
```

mstatusH register state (only for 32-bit)

20.16.2.13 ra

```
unsigned long sbi_trap_regs::ra
```

ra register state

20.16.2.14 s0

```
unsigned long sbi_trap_regs::s0
```

s0 register state

20.16.2.15 s1

```
unsigned long sbi_trap_regs::s1
```

s1 register state

20.16.2.16 s10

```
unsigned long sbi_trap_regs::s10
```

s10 register state

20.16.2.17 s11

```
unsigned long sbi_trap_regs::s11
```

s11 register state

20.16.2.18 s2

```
unsigned long sbi_trap_regs::s2
```

s2 register state

20.16.2.19 s3

```
unsigned long sbi_trap_regs::s3
```

s3 register state

20.16.2.20 s4

```
unsigned long sbi_trap_regs::s4
```

s4 register state

20.16.2.21 s5

```
unsigned long sbi_trap_regs::s5
```

s5 register state

20.16.2.22 s6

```
unsigned long sbi_trap_regs::s6
```

s6 register state

20.16.2.23 s7

```
unsigned long sbi_trap_regs::s7
```

s7 register state

20.16.2.24 s8

```
unsigned long sbi_trap_regs::s8
```

s8 register state

20.16.2.25 s9

```
unsigned long sbi_trap_regs::s9
```

s9 register state

20.16.2.26 sp

```
unsigned long sbi_trap_regs::sp
```

sp register state

20.16.2.27 t0

```
unsigned long sbi_trap_regs::t0
```

t0 register state

20.16.2.28 t1

```
unsigned long sbi_trap_regs::t1
```

t1 register state

20.16.2.29 t2

```
unsigned long sbi_trap_regs::t2
```

t2 register state

20.16.2.30 t3

```
unsigned long sbi_trap_regs::t3
```

t3 register state

20.16.2.31 t4

```
unsigned long sbi_trap_regs::t4
```

t4 register state

20.16.2.32 t5

```
unsigned long sbi_trap_regs::t5
```

t5 register state

20.16.2.33 t6

```
unsigned long sbi_trap_regs::t6
```

t6 register state

20.16.2.34 tp

```
unsigned long sbi_trap_regs::tp
```

tp register state

20.16.2.35 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](#)

20.17 spinlock_t Struct Reference

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

Data Fields

- volatile long [lock](#)

20.17.1 Field Documentation

20.17.1.1 lock

```
volatile long spinlock_t::lock
```

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

- [include/sbi/riscv_locks.h](#)

Chapter 21

File Documentation

21.1 docs/contributing.md File Reference

21.2 docs/firmware/fw.md File Reference

21.3 docs/firmware/fw_dynamic.md File Reference

21.4 docs/firmware/fw_jump.md File Reference

21.5 docs/firmware/fw_payload.md File Reference

21.6 docs/firmware/payload_linux.md File Reference

21.7 docs/firmware/payload_uboot.md File Reference

21.8 docs/library_usage.md File Reference

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

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

21.11 docs/platform/platform.md File Reference

21.12 docs/platform/qemu_virt.md File Reference

21.13 docs/platform/sifive_fu540.md File Reference

21.14 docs/platform/spike.md File Reference

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

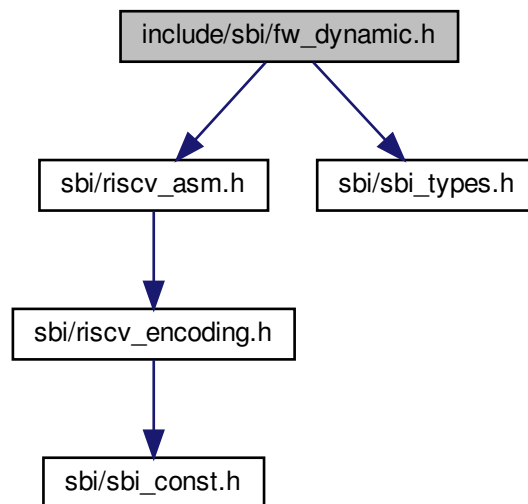
21.16 docs/platform_guide.md File Reference

21.17 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](#)

21.17.1 Macro Definition Documentation

21.17.1.1 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)

21.17.1.2 FW_DYNAMIC_INFO_MAGIC_OFFSET

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

Offset of magic member in [fw_dynamic_info](#)

21.17.1.3 FW_DYNAMIC_INFO_MAGIC_VALUE

```
#define FW_DYNAMIC_INFO_MAGIC_VALUE 0x4942534f
```

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

21.17.1.4 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)

21.17.1.5 FW_DYNAMIC_INFO_NEXT_MODE_M

```
#define FW_DYNAMIC_INFO_NEXT_MODE_M 0x3
```

21.17.1.6 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)

21.17.1.7 FW_DYNAMIC_INFO_NEXT_MODE_S

```
#define FW_DYNAMIC_INFO_NEXT_MODE_S 0x1
```

21.17.1.8 FW_DYNAMIC_INFO_NEXT_MODE_U

```
#define FW_DYNAMIC_INFO_NEXT_MODE_U 0x0
```

Possible next mode values

21.17.1.9 FW_DYNAMIC_INFO_OPTIONS_OFFSET

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

Offset of options member in [fw_dynamic_info](#) (version ≥ 1)

21.17.1.10 FW_DYNAMIC_INFO_VERSION_MAX

```
#define FW_DYNAMIC_INFO_VERSION_MAX 0x2
```

Maximum supported info version

21.17.1.11 FW_DYNAMIC_INFO_VERSION_OFFSET

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

Offset of version member in [fw_dynamic_info](#)

21.17.2 Variable Documentation

21.17.2.1 __packed

```
struct fw\_dynamic\_info __packed
```


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)
- static void [misa_string](#) (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 *log2len_out)

21.18.1 Macro Definition Documentation

21.18.1.1 [__ASM_STR](#)

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

21.18.1.2 [csr_clear](#)

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

Value:

```
((  
    unsigned long __v = (unsigned long)(val);  
    __asm__ __volatile__("csrr " \_\_ASM\_STR(csr) ", %0" \n  
        :  
        : "rK"(__v)  
        : "memory");  
))
```

21.18.1.3 [csr_read](#)

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

Value:

```
((  
    register unsigned long __v;  
    __asm__ __volatile__("csrr %0, " \_\_ASM\_STR(csr) \n  
        : "=r"(__v)  
        :  
        : "memory");  
    __v;  
))
```

21.18.1.4 csr_read_clear

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

Value:

```
((  
    unsigned long __v = (unsigned long)(val);  
    __asm__ __volatile__("csrrc %0, " __ASM_STR(csr) ", %1" \br/>        : "=r"(__v)  
        : "rK"(__v)  
        : "memory");  
    __v;  
}))
```

21.18.1.5 csr_read_set

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

Value:

```
((  
    unsigned long __v = (unsigned long)(val);  
    __asm__ __volatile__("csrrs %0, " __ASM_STR(csr) ", %1" \br/>        : "=r"(__v)  
        : "rK"(__v)  
        : "memory");  
    __v;  
}))
```

21.18.1.6 csr_set

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

Value:

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

21.18.1.7 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;
    ))
```

21.18.1.8 csr_write

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

Value:

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

21.18.1.9 LGREG

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

21.18.1.10 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);\
    ))
```


21.18.1.11 PAGE_MASK

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

21.18.1.12 PAGE_SHIFT

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

21.18.1.13 PAGE_SIZE

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

21.18.1.14 REG_L

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

21.18.1.15 REG_S

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

21.18.1.16 SZREG

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

21.18.1.17 wfi

```
#define wfi( )
```

Value:

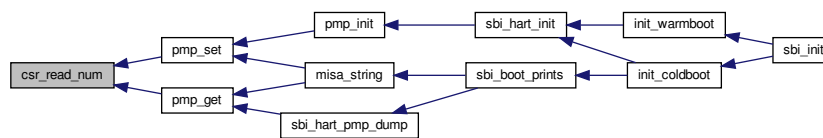
```
do {  
    __asm__ __volatile__("wfi" ::: "memory"); \  
} while (0)
```

21.18.2 Function Documentation

21.18.2.1 csr_read_num()

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

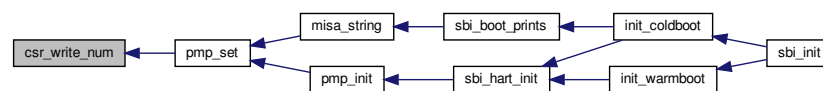
Here is the caller graph for this function:



21.18.2.2 csr_write_num()

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

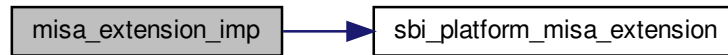
Here is the caller graph for this function:



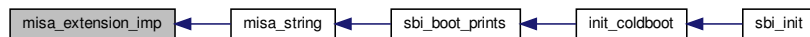
21.18.2.3 misa_extension_imp()

```
int misa_extension_imp (
    char ext )
```

Here is the call graph for this function:



Here is the caller graph for this function:

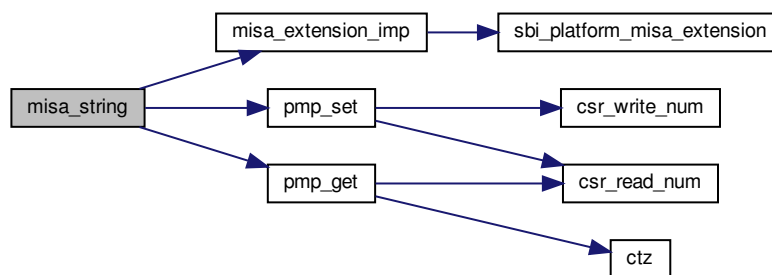


21.18.2.4 misa_string()

```

static void misa_string (
    char * out,
    unsigned int out_sz ) [inline], [static]
  
```

Here is the call graph for this function:



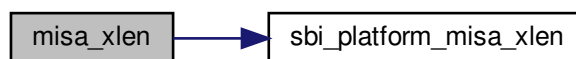
Here is the caller graph for this function:



21.18.2.5 misa_xlen()

```
int misa_xlen (
    void )
```

Here is the call graph for this function:



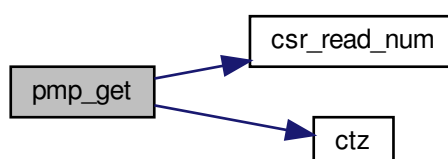
Here is the caller graph for this function:



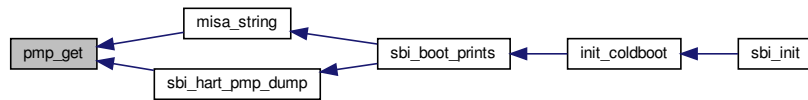
21.18.2.6 pmp_get()

```
int pmp_get (
    unsigned int n,
    unsigned long * prot_out,
    unsigned long * addr_out,
    unsigned long * log2len_out )
```

Here is the call graph for this function:



Here is the caller graph for this function:



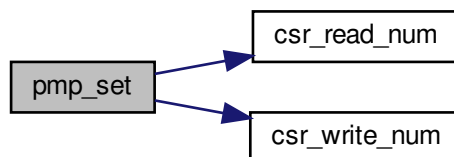
21.18.2.7 pmp_set()

```

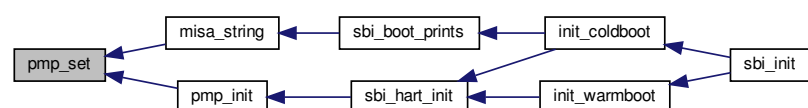
int pmp_set (
    unsigned int n,
    unsigned long prot,
    unsigned long addr,
    unsigned long log2len )

```

Here is the call graph for this function:

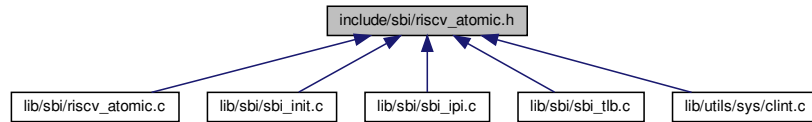


Here is the caller graph for this function:



21.19 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 [arch_atomic_cmpxchg](#) ([atomic_t](#) *atom, long oldval, long newval)
- long [arch_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)

21.19.1 Macro Definition Documentation

21.19.1.1 ATOMIC_INIT

```

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

21.19.1.2 ATOMIC_INITIALIZER

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

Value:

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

21.19.2 Function Documentation

21.19.2.1 arch_atomic_cmpxchg()

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

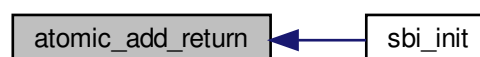
21.19.2.2 arch_atomic_xchg()

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

21.19.2.3 atomic_add_return()

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

Here is the caller graph for this function:



21.19.2.4 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. : Bit to set. : atomic variable to modify Here is the call graph for this function:



21.19.2.5 atomic_raw_clear_bit()

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

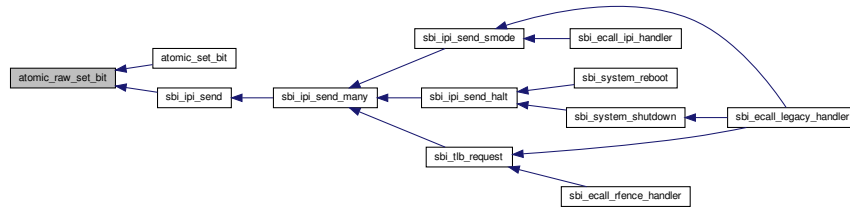
Clear a bit in any address and return the new value . : Bit to set. : Address to modify Here is the caller graph for this function:



21.19.2.6 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 . : Bit to set. : Address to modify Here is the caller graph for this function:



21.19.2.7 atomic_raw_xchg_uint()

```

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

```

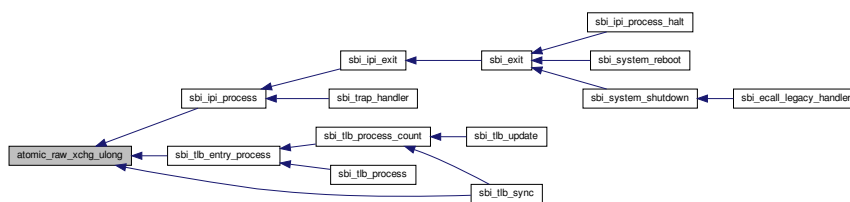
21.19.2.8 atomic_raw_xchg_ulong()

```

unsigned long atomic_raw_xchg_ulong (
    volatile unsigned long * ptr,
    unsigned long newval )

```

Here is the caller graph for this function:



21.19.2.9 atomic_read()

```

long atomic_read (
    atomic_t * atom )

```

21.19.2.10 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. : Bit to set. : atomic variable to modify Here is the call graph for this function:



21.19.2.11 atomic_sub_return()

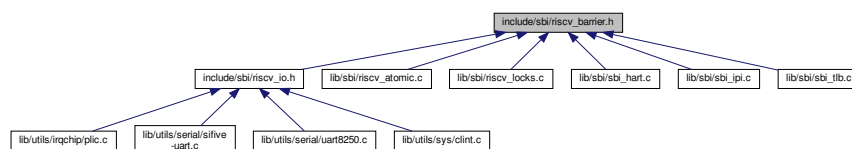
```
long atomic_sub_return (
    atomic_t * atom,
    long value )
```

21.19.2.12 atomic_write()

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

21.20 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)`

21.20.1 Macro Definition Documentation

21.20.1.1 __smp_load_acquire

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

Value:

```
((  
    typeof(*p) __p1 = *(p); \br/>    RISCV_FENCE(r, rw); \br/>    __p1; \br/>))
```

21.20.1.2 __smp_store_release

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

Value:

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

21.20.1.3 `cpu_relax`

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

21.20.1.4 `mb`

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

21.20.1.5 `RISCV_ACQUIRE_BARRIER`

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

21.20.1.6 `RISCV_FENCE`

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

21.20.1.7 `RISCV_RELEASE_BARRIER`

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

21.20.1.8 `rmb`

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

21.20.1.9 `smp_mb`

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

21.20.1.10 smp_rmb

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

21.20.1.11 smp_wmb

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

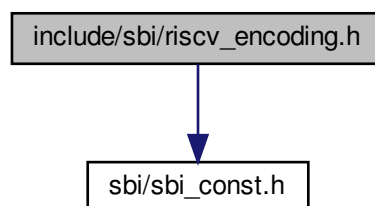
21.20.1.12 wmb

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

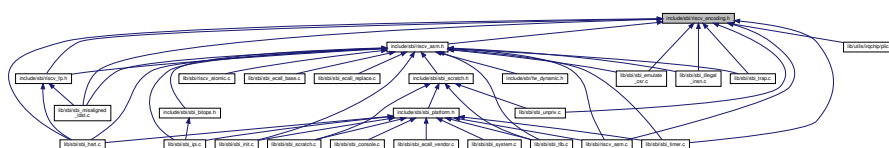
21.21 include/sbi/riscv_encoding.h File Reference

```
#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_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_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_VTVM_UL(0x00100000)
- #define HSTATUS_SP2V_UL(0x00000200)
- #define HSTATUS_SP2P_UL(0x00000100)
- #define HSTATUS_SPV_UL(0x00000080)
- #define HSTATUS_SPRV_UL(0x00000001)
- #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_SSSIP MIP_SSSIP
- #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(0x003FFFFFFF)
- #define SATP64_MODE _ULL(0xF000000000000000)
- #define SATP64_ASID _ULL(0x0FFFF00000000000)
- #define SATP64_PPN _ULL(0x00000FFFFFFFFFFFFF)
- #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 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 PTE_V _UL(0x001) /* Valid */
- #define PTE_R _UL(0x002) /* Read */
- #define PTE_W _UL(0x004) /* Write */
- #define PTE_X _UL(0x008) /* Execute */
- #define PTE_U _UL(0x010) /* User */
- #define PTE_G _UL(0x020) /* Global */
- #define PTE_A _UL(0x040) /* Accessed */
- #define PTE_D _UL(0x080) /* Dirty */
- #define PTE_SOFT _UL(0x300) /* Reserved for Software */
- #define PTE_PPN_SHIFT 10
- #define PTE_TABLE(PTE) (((PTE) & (PTE_V | PTE_R | PTE_W | PTE_X)) == PTE_V)
- #define MSTATUS_SD MSTATUS32_SD
- #define SSTATUS_SD SSTATUS32_SD
- #define RISCVP_GLEVEL_BITS 10
- #define SATP_MODE SATP32_MODE
- #define RISCVP_PGSHIFT 12
- #define RISCVP_PGSIZE (1 << RISCVP_PGSHIFT)
- #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_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](#) 0xfffff00
- #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

21.21.1 Macro Definition Documentation

21.21.1.1 CAUSE_BREAKPOINT

```
#define CAUSE_BREAKPOINT 0x3
```

21.21.1.2 CAUSE_FETCH_ACCESS

```
#define CAUSE_FETCH_ACCESS 0x1
```

21.21.1.3 CAUSE_FETCH_GUEST_PAGE_FAULT

```
#define CAUSE_FETCH_GUEST_PAGE_FAULT 0x14
```

21.21.1.4 CAUSE_FETCH_PAGE_FAULT

```
#define CAUSE_FETCH_PAGE_FAULT 0xc
```

21.21.1.5 CAUSE_HYPERVISOR_ECALL

```
#define CAUSE_HYPERVISOR_ECALL 0x9
```

21.21.1.6 CAUSE_ILLEGAL_INSTRUCTION

```
#define CAUSE_ILLEGAL_INSTRUCTION 0x2
```

21.21.1.7 CAUSE_LOAD_ACCESS

```
#define CAUSE_LOAD_ACCESS 0x5
```

21.21.1.8 CAUSE_LOAD_GUEST_PAGE_FAULT

```
#define CAUSE_LOAD_GUEST_PAGE_FAULT 0x15
```

21.21.1.9 CAUSE_LOAD_PAGE_FAULT

```
#define CAUSE_LOAD_PAGE_FAULT 0xd
```

21.21.1.10 CAUSE_MACHINE_ECALL

```
#define CAUSE_MACHINE_ECALL 0xb
```

21.21.1.11 CAUSE_MISALIGNED_FETCH

```
#define CAUSE_MISALIGNED_FETCH 0x0
```

21.21.1.12 CAUSE_MISALIGNED_LOAD

```
#define CAUSE_MISALIGNED_LOAD 0x4
```

21.21.1.13 CAUSE_MISALIGNED_STORE

```
#define CAUSE_MISALIGNED_STORE 0x6
```

21.21.1.14 CAUSE_STORE_ACCESS

```
#define CAUSE_STORE_ACCESS 0x7
```

21.21.1.15 CAUSE_STORE_GUEST_PAGE_FAULT

```
#define CAUSE_STORE_GUEST_PAGE_FAULT 0x17
```

21.21.1.16 CAUSE_STORE_PAGE_FAULT

```
#define CAUSE_STORE_PAGE_FAULT 0xf
```

21.21.1.17 CAUSE_SUPERVISOR_ECALL

```
#define CAUSE_SUPERVISOR_ECALL 0xa
```

21.21.1.18 CAUSE_USER_ECALL

```
#define CAUSE_USER_ECALL 0x8
```


21.21.1.19 CSR_CYCLE

```
#define CSR_CYCLE 0xc00
```

21.21.1.20 CSR_CYCLEH

```
#define CSR_CYCLEH 0xc80
```

21.21.1.21 CSR_DCSR

```
#define CSR_DCSR 0x7b0
```

21.21.1.22 CSR_DPC

```
#define CSR_DPC 0x7b1
```

21.21.1.23 CSR_DSCRATCH

```
#define CSR_DSCRATCH 0x7b2
```

21.21.1.24 CSR_FCSR

```
#define CSR_FCSR 0x3
```

21.21.1.25 CSR_FFLAGS

```
#define CSR_FFLAGS 0x1
```

21.21.1.26 CSR_FRM

```
#define CSR_FRM 0x2
```

21.21.1.27 CSR_HCOUNTERNEN

```
#define CSR_HCOUNTERNEN 0x606
```

21.21.1.28 CSR_HEDELEG

```
#define CSR_HEDELEG 0x602
```

21.21.1.29 CSR_HGATP

```
#define CSR_HGATP 0x680
```

21.21.1.30 CSR_HGEIE

```
#define CSR_HGEIE 0x607
```

21.21.1.31 CSR_HGEIP

```
#define CSR_HGEIP 0xe07
```

21.21.1.32 CSR_HIDELEG

```
#define CSR_HIDELEG 0x603
```

21.21.1.33 CSR_HIE

```
#define CSR_HIE 0x604
```

21.21.1.34 CSR_HIP

```
#define CSR_HIP 0x644
```

21.21.1.35 CSR_HPMCounter10

```
#define CSR_HPMCounter10 0xc0a
```

21.21.1.36 CSR_HPMCounter10H

```
#define CSR_HPMCounter10H 0xc8a
```

21.21.1.37 CSR_HPMCounter11

```
#define CSR_HPMCounter11 0xc0b
```

21.21.1.38 CSR_HPMCounter11H

```
#define CSR_HPMCounter11H 0xc8b
```

21.21.1.39 CSR_HPMCounter12

```
#define CSR_HPMCounter12 0xc0c
```

21.21.1.40 CSR_HPMCounter12H

```
#define CSR_HPMCounter12H 0xc8c
```

21.21.1.41 CSR_HPMCounter13

```
#define CSR_HPMCounter13 0xc0d
```

21.21.1.42 CSR_HPMCounter13H

```
#define CSR_HPMCounter13H 0xc8d
```

21.21.1.43 CSR_HPMCounter14

```
#define CSR_HPMCounter14 0xc0e
```

21.21.1.44 CSR_HPMCounter14H

```
#define CSR_HPMCounter14H 0xc8e
```

21.21.1.45 CSR_HPMCounter15

```
#define CSR_HPMCounter15 0xc0f
```

21.21.1.46 CSR_HPMCounter15H

```
#define CSR_HPMCounter15H 0xc8f
```

21.21.1.47 CSR_HPMCounter16

```
#define CSR_HPMCounter16 0xc10
```

21.21.1.48 CSR_HPMCounter16H

```
#define CSR_HPMCounter16H 0xc90
```

21.21.1.49 CSR_HPMCounter17

```
#define CSR_HPMCounter17 0xc11
```

21.21.1.50 CSR_HPMCounter17H

```
#define CSR_HPMCounter17H 0xc91
```

21.21.1.51 CSR_HPM_COUNTER18

```
#define CSR_HPM_COUNTER18 0xc12
```

21.21.1.52 CSR_HPM_COUNTER18H

```
#define CSR_HPM_COUNTER18H 0xc92
```

21.21.1.53 CSR_HPM_COUNTER19

```
#define CSR_HPM_COUNTER19 0xc13
```

21.21.1.54 CSR_HPM_COUNTER19H

```
#define CSR_HPM_COUNTER19H 0xc93
```

21.21.1.55 CSR_HPM_COUNTER20

```
#define CSR_HPM_COUNTER20 0xc14
```

21.21.1.56 CSR_HPM_COUNTER20H

```
#define CSR_HPM_COUNTER20H 0xc94
```

21.21.1.57 CSR_HPM_COUNTER21

```
#define CSR_HPM_COUNTER21 0xc15
```

21.21.1.58 CSR_HPM_COUNTER21H

```
#define CSR_HPM_COUNTER21H 0xc95
```

21.21.1.59 CSR_HPMCounter22

```
#define CSR_HPMCounter22 0xc16
```

21.21.1.60 CSR_HPMCounter22H

```
#define CSR_HPMCounter22H 0xc96
```

21.21.1.61 CSR_HPMCounter23

```
#define CSR_HPMCounter23 0xc17
```

21.21.1.62 CSR_HPMCounter23H

```
#define CSR_HPMCounter23H 0xc97
```

21.21.1.63 CSR_HPMCounter24

```
#define CSR_HPMCounter24 0xc18
```

21.21.1.64 CSR_HPMCounter24H

```
#define CSR_HPMCounter24H 0xc98
```

21.21.1.65 CSR_HPMCounter25

```
#define CSR_HPMCounter25 0xc19
```

21.21.1.66 CSR_HPMCounter25H

```
#define CSR_HPMCounter25H 0xc99
```

21.21.1.67 CSR_HPM_COUNTER26

```
#define CSR_HPM_COUNTER26 0xc1a
```

21.21.1.68 CSR_HPM_COUNTER26H

```
#define CSR_HPM_COUNTER26H 0xc9a
```

21.21.1.69 CSR_HPM_COUNTER27

```
#define CSR_HPM_COUNTER27 0xc1b
```

21.21.1.70 CSR_HPM_COUNTER27H

```
#define CSR_HPM_COUNTER27H 0xc9b
```

21.21.1.71 CSR_HPM_COUNTER28

```
#define CSR_HPM_COUNTER28 0xc1c
```

21.21.1.72 CSR_HPM_COUNTER28H

```
#define CSR_HPM_COUNTER28H 0xc9c
```

21.21.1.73 CSR_HPM_COUNTER29

```
#define CSR_HPM_COUNTER29 0xc1d
```

21.21.1.74 CSR_HPM_COUNTER29H

```
#define CSR_HPM_COUNTER29H 0xc9d
```

21.21.1.75 CSR_HPMCounter3

```
#define CSR_HPMCounter3 0xc03
```

21.21.1.76 CSR_HPMCounter30

```
#define CSR_HPMCounter30 0xc1e
```

21.21.1.77 CSR_HPMCounter30H

```
#define CSR_HPMCounter30H 0xc9e
```

21.21.1.78 CSR_HPMCounter31

```
#define CSR_HPMCounter31 0xc1f
```

21.21.1.79 CSR_HPMCounter31H

```
#define CSR_HPMCounter31H 0xc9f
```

21.21.1.80 CSR_HPMCounter3H

```
#define CSR_HPMCounter3H 0xc83
```

21.21.1.81 CSR_HPMCounter4

```
#define CSR_HPMCounter4 0xc04
```

21.21.1.82 CSR_HPMCounter4H

```
#define CSR_HPMCounter4H 0xc84
```


21.21.1.83 CSR_HPM_COUNTER5

```
#define CSR_HPM_COUNTER5 0xc05
```

21.21.1.84 CSR_HPM_COUNTER5H

```
#define CSR_HPM_COUNTER5H 0xc85
```

21.21.1.85 CSR_HPM_COUNTER6

```
#define CSR_HPM_COUNTER6 0xc06
```

21.21.1.86 CSR_HPM_COUNTER6H

```
#define CSR_HPM_COUNTER6H 0xc86
```

21.21.1.87 CSR_HPM_COUNTER7

```
#define CSR_HPM_COUNTER7 0xc07
```

21.21.1.88 CSR_HPM_COUNTER7H

```
#define CSR_HPM_COUNTER7H 0xc87
```

21.21.1.89 CSR_HPM_COUNTER8

```
#define CSR_HPM_COUNTER8 0xc08
```

21.21.1.90 CSR_HPM_COUNTER8H

```
#define CSR_HPM_COUNTER8H 0xc88
```

21.21.1.91 CSR_HPM_COUNTER9

```
#define CSR_HPM_COUNTER9 0xc09
```

21.21.1.92 CSR_HPM_COUNTER9H

```
#define CSR_HPM_COUNTER9H 0xc89
```

21.21.1.93 CSR_HSTATUS

```
#define CSR_HSTATUS 0x600
```

21.21.1.94 CSR_HTIMEDELTA

```
#define CSR_HTIMEDELTA 0x605
```

21.21.1.95 CSR_HTIMEDELTAH

```
#define CSR_HTIMEDELTAH 0x615
```

21.21.1.96 CSR_HTINST

```
#define CSR_HTINST 0x64a
```

21.21.1.97 CSR_HTVAL

```
#define CSR_HTVAL 0x643
```

21.21.1.98 CSR_INSTRET

```
#define CSR_INSTRET 0xc02
```

21.21.1.99 CSR_INSTRETH

```
#define CSR_INSTRETH 0xc82
```

21.21.1.100 CSR_MARCHID

```
#define CSR_MARCHID 0xf12
```

21.21.1.101 CSR_MCAUSE

```
#define CSR_MCAUSE 0x342
```

21.21.1.102 CSR_MCOUNTEREN

```
#define CSR_MCOUNTEREN 0x306
```

21.21.1.103 CSR_MCYCLE

```
#define CSR_MCYCLE 0xb00
```

21.21.1.104 CSR_MCYCLEH

```
#define CSR_MCYCLEH 0xb80
```

21.21.1.105 CSR_MEDELEG

```
#define CSR_MEDELEG 0x302
```

21.21.1.106 CSR_MEPC

```
#define CSR_MEPC 0x341
```

21.21.1.107 CSR_MHARTID

```
#define CSR_MHARTID 0xf14
```

21.21.1.108 CSR_MHPMCOUNTER10

```
#define CSR_MHPMCOUNTER10 0xb0a
```

21.21.1.109 CSR_MHPMCOUNTER10H

```
#define CSR_MHPMCOUNTER10H 0xb8a
```

21.21.1.110 CSR_MHPMCOUNTER11

```
#define CSR_MHPMCOUNTER11 0xb0b
```

21.21.1.111 CSR_MHPMCOUNTER11H

```
#define CSR_MHPMCOUNTER11H 0xb8b
```

21.21.1.112 CSR_MHPMCOUNTER12

```
#define CSR_MHPMCOUNTER12 0xb0c
```

21.21.1.113 CSR_MHPMCOUNTER12H

```
#define CSR_MHPMCOUNTER12H 0xb8c
```

21.21.1.114 CSR_MHPMCOUNTER13

```
#define CSR_MHPMCOUNTER13 0xb0d
```

21.21.1.115 CSR_MHPMCOUNTER13H

```
#define CSR_MHPMCOUNTER13H 0xb8d
```

21.21.1.116 CSR_MHPMCOUNTER14

```
#define CSR_MHPMCOUNTER14 0xb0e
```

21.21.1.117 CSR_MHPMCOUNTER14H

```
#define CSR_MHPMCOUNTER14H 0xb8e
```

21.21.1.118 CSR_MHPMCOUNTER15

```
#define CSR_MHPMCOUNTER15 0xb0f
```

21.21.1.119 CSR_MHPMCOUNTER15H

```
#define CSR_MHPMCOUNTER15H 0xb8f
```

21.21.1.120 CSR_MHPMCOUNTER16

```
#define CSR_MHPMCOUNTER16 0xb10
```

21.21.1.121 CSR_MHPMCOUNTER16H

```
#define CSR_MHPMCOUNTER16H 0xb90
```

21.21.1.122 CSR_MHPMCOUNTER17

```
#define CSR_MHPMCOUNTER17 0xb11
```

21.21.1.123 CSR_MHPMCOUNTER17H

```
#define CSR_MHPMCOUNTER17H 0xb91
```

21.21.1.124 CSR_MHPMCOUNTER18

```
#define CSR_MHPMCOUNTER18 0xb12
```

21.21.1.125 CSR_MHPMCOUNTER18H

```
#define CSR_MHPMCOUNTER18H 0xb92
```

21.21.1.126 CSR_MHPMCOUNTER19

```
#define CSR_MHPMCOUNTER19 0xb13
```

21.21.1.127 CSR_MHPMCOUNTER19H

```
#define CSR_MHPMCOUNTER19H 0xb93
```

21.21.1.128 CSR_MHPMCOUNTER20

```
#define CSR_MHPMCOUNTER20 0xb14
```

21.21.1.129 CSR_MHPMCOUNTER20H

```
#define CSR_MHPMCOUNTER20H 0xb94
```

21.21.1.130 CSR_MHPMCOUNTER21

```
#define CSR_MHPMCOUNTER21 0xb15
```

21.21.1.131 CSR_MHPMCOUNTER21H

```
#define CSR_MHPMCOUNTER21H 0xb95
```

21.21.1.132 CSR_MHPMCOUNTER22

```
#define CSR_MHPMCOUNTER22 0xb16
```

21.21.1.133 CSR_MHPMCOUNTER22H

```
#define CSR_MHPMCOUNTER22H 0xb96
```

21.21.1.134 CSR_MHPMCOUNTER23

```
#define CSR_MHPMCOUNTER23 0xb17
```

21.21.1.135 CSR_MHPMCOUNTER23H

```
#define CSR_MHPMCOUNTER23H 0xb97
```

21.21.1.136 CSR_MHPMCOUNTER24

```
#define CSR_MHPMCOUNTER24 0xb18
```

21.21.1.137 CSR_MHPMCOUNTER24H

```
#define CSR_MHPMCOUNTER24H 0xb98
```

21.21.1.138 CSR_MHPMCOUNTER25

```
#define CSR_MHPMCOUNTER25 0xb19
```

21.21.1.139 CSR_MHPMCOUNTER25H

```
#define CSR_MHPMCOUNTER25H 0xb99
```

21.21.1.140 CSR_MHPMCOUNTER26

```
#define CSR_MHPMCOUNTER26 0xb1a
```

21.21.1.141 CSR_MHPMCOUNTER26H

```
#define CSR_MHPMCOUNTER26H 0xb9a
```

21.21.1.142 CSR_MHPMCOUNTER27

```
#define CSR_MHPMCOUNTER27 0xb1b
```

21.21.1.143 CSR_MHPMCOUNTER27H

```
#define CSR_MHPMCOUNTER27H 0xb9b
```

21.21.1.144 CSR_MHPMCOUNTER28

```
#define CSR_MHPMCOUNTER28 0xb1c
```

21.21.1.145 CSR_MHPMCOUNTER28H

```
#define CSR_MHPMCOUNTER28H 0xb9c
```

21.21.1.146 CSR_MHPMCOUNTER29

```
#define CSR_MHPMCOUNTER29 0xb1d
```


21.21.1.147 CSR_MHPMCOUNTER29H

```
#define CSR_MHPMCOUNTER29H 0xb9d
```

21.21.1.148 CSR_MHPMCOUNTER3

```
#define CSR_MHPMCOUNTER3 0xb03
```

21.21.1.149 CSR_MHPMCOUNTER30

```
#define CSR_MHPMCOUNTER30 0xb1e
```

21.21.1.150 CSR_MHPMCOUNTER30H

```
#define CSR_MHPMCOUNTER30H 0xb9e
```

21.21.1.151 CSR_MHPMCOUNTER31

```
#define CSR_MHPMCOUNTER31 0xb1f
```

21.21.1.152 CSR_MHPMCOUNTER31H

```
#define CSR_MHPMCOUNTER31H 0xb9f
```

21.21.1.153 CSR_MHPMCOUNTER3H

```
#define CSR_MHPMCOUNTER3H 0xb83
```

21.21.1.154 CSR_MHPMCOUNTER4

```
#define CSR_MHPMCOUNTER4 0xb04
```

21.21.1.155 CSR_MHPMCOUNTER4H

```
#define CSR_MHPMCOUNTER4H 0xb84
```

21.21.1.156 CSR_MHPMCOUNTER5

```
#define CSR_MHPMCOUNTER5 0xb05
```

21.21.1.157 CSR_MHPMCOUNTER5H

```
#define CSR_MHPMCOUNTER5H 0xb85
```

21.21.1.158 CSR_MHPMCOUNTER6

```
#define CSR_MHPMCOUNTER6 0xb06
```

21.21.1.159 CSR_MHPMCOUNTER6H

```
#define CSR_MHPMCOUNTER6H 0xb86
```

21.21.1.160 CSR_MHPMCOUNTER7

```
#define CSR_MHPMCOUNTER7 0xb07
```

21.21.1.161 CSR_MHPMCOUNTER7H

```
#define CSR_MHPMCOUNTER7H 0xb87
```

21.21.1.162 CSR_MHPMCOUNTER8

```
#define CSR_MHPMCOUNTER8 0xb08
```

21.21.1.163 CSR_MHPMCOUNTER8H

```
#define CSR_MHPMCOUNTER8H 0xb88
```

21.21.1.164 CSR_MHPMCOUNTER9

```
#define CSR_MHPMCOUNTER9 0xb09
```

21.21.1.165 CSR_MHPMCOUNTER9H

```
#define CSR_MHPMCOUNTER9H 0xb89
```

21.21.1.166 CSR_MHPMEVENT10

```
#define CSR_MHPMEVENT10 0x32a
```

21.21.1.167 CSR_MHPMEVENT11

```
#define CSR_MHPMEVENT11 0x32b
```

21.21.1.168 CSR_MHPMEVENT12

```
#define CSR_MHPMEVENT12 0x32c
```

21.21.1.169 CSR_MHPMEVENT13

```
#define CSR_MHPMEVENT13 0x32d
```

21.21.1.170 CSR_MHPMEVENT14

```
#define CSR_MHPMEVENT14 0x32e
```

21.21.1.171 CSR_MHPMEVENT15

```
#define CSR_MHPMEVENT15 0x32f
```

21.21.1.172 CSR_MHPMEVENT16

```
#define CSR_MHPMEVENT16 0x330
```

21.21.1.173 CSR_MHPMEVENT17

```
#define CSR_MHPMEVENT17 0x331
```

21.21.1.174 CSR_MHPMEVENT18

```
#define CSR_MHPMEVENT18 0x332
```

21.21.1.175 CSR_MHPMEVENT19

```
#define CSR_MHPMEVENT19 0x333
```

21.21.1.176 CSR_MHPMEVENT20

```
#define CSR_MHPMEVENT20 0x334
```

21.21.1.177 CSR_MHPMEVENT21

```
#define CSR_MHPMEVENT21 0x335
```

21.21.1.178 CSR_MHPMEVENT22

```
#define CSR_MHPMEVENT22 0x336
```

21.21.1.179 CSR_MHPMEVENT23

```
#define CSR_MHPMEVENT23 0x337
```

21.21.1.180 CSR_MHPMEVENT24

```
#define CSR_MHPMEVENT24 0x338
```

21.21.1.181 CSR_MHPMEVENT25

```
#define CSR_MHPMEVENT25 0x339
```

21.21.1.182 CSR_MHPMEVENT26

```
#define CSR_MHPMEVENT26 0x33a
```

21.21.1.183 CSR_MHPMEVENT27

```
#define CSR_MHPMEVENT27 0x33b
```

21.21.1.184 CSR_MHPMEVENT28

```
#define CSR_MHPMEVENT28 0x33c
```

21.21.1.185 CSR_MHPMEVENT29

```
#define CSR_MHPMEVENT29 0x33d
```

21.21.1.186 CSR_MHPMEVENT3

```
#define CSR_MHPMEVENT3 0x323
```

21.21.1.187 CSR_MHPMEVENT30

```
#define CSR_MHPMEVENT30 0x33e
```

21.21.1.188 CSR_MHPMEVENT31

```
#define CSR_MHPMEVENT31 0x33f
```

21.21.1.189 CSR_MHPMEVENT4

```
#define CSR_MHPMEVENT4 0x324
```

21.21.1.190 CSR_MHPMEVENT5

```
#define CSR_MHPMEVENT5 0x325
```

21.21.1.191 CSR_MHPMEVENT6

```
#define CSR_MHPMEVENT6 0x326
```

21.21.1.192 CSR_MHPMEVENT7

```
#define CSR_MHPMEVENT7 0x327
```

21.21.1.193 CSR_MHPMEVENT8

```
#define CSR_MHPMEVENT8 0x328
```

21.21.1.194 CSR_MHPMEVENT9

```
#define CSR_MHPMEVENT9 0x329
```

21.21.1.195 CSR_MIDELEG

```
#define CSR_MIDELEG 0x303
```

21.21.1.196 CSR_MIE

```
#define CSR_MIE 0x304
```

21.21.1.197 CSR_MIMPID

```
#define CSR_MIMPID 0xf13
```

21.21.1.198 CSR_MINSTRET

```
#define CSR_MINSTRET 0xb02
```

21.21.1.199 CSR_MINSTRETH

```
#define CSR_MINSTRETH 0xb82
```

21.21.1.200 CSR_MIP

```
#define CSR_MIP 0x344
```

21.21.1.201 CSR_MISA

```
#define CSR_MISA 0x301
```

21.21.1.202 CSR_MSCRATCH

```
#define CSR_MSCRATCH 0x340
```

21.21.1.203 CSR_MSTATUS

```
#define CSR_MSTATUS 0x300
```

21.21.1.204 CSR_MSTATUSH

```
#define CSR_MSTATUSH 0x310
```

21.21.1.205 CSR_MTINST

```
#define CSR_MTINST 0x34a
```

21.21.1.206 CSR_MTVAL

```
#define CSR_MTVAL 0x343
```

21.21.1.207 CSR_MTVAL2

```
#define CSR_MTVAL2 0x34b
```

21.21.1.208 CSR_MTVEC

```
#define CSR_MTVEC 0x305
```

21.21.1.209 CSR_MVENDORID

```
#define CSR_MVENDORID 0xf11
```

21.21.1.210 CSR_PMPADDR0

```
#define CSR_PMPADDR0 0x3b0
```


21.21.1.211 CSR_PMPADDR1

```
#define CSR_PMPADDR1 0x3b1
```

21.21.1.212 CSR_PMPADDR10

```
#define CSR_PMPADDR10 0x3ba
```

21.21.1.213 CSR_PMPADDR11

```
#define CSR_PMPADDR11 0x3bb
```

21.21.1.214 CSR_PMPADDR12

```
#define CSR_PMPADDR12 0x3bc
```

21.21.1.215 CSR_PMPADDR13

```
#define CSR_PMPADDR13 0x3bd
```

21.21.1.216 CSR_PMPADDR14

```
#define CSR_PMPADDR14 0x3be
```

21.21.1.217 CSR_PMPADDR15

```
#define CSR_PMPADDR15 0x3bf
```

21.21.1.218 CSR_PMPADDR2

```
#define CSR_PMPADDR2 0x3b2
```

21.21.1.219 CSR_PMPADDR3

```
#define CSR_PMPADDR3 0x3b3
```

21.21.1.220 CSR_PMPADDR4

```
#define CSR_PMPADDR4 0x3b4
```

21.21.1.221 CSR_PMPADDR5

```
#define CSR_PMPADDR5 0x3b5
```

21.21.1.222 CSR_PMPADDR6

```
#define CSR_PMPADDR6 0x3b6
```

21.21.1.223 CSR_PMPADDR7

```
#define CSR_PMPADDR7 0x3b7
```

21.21.1.224 CSR_PMPADDR8

```
#define CSR_PMPADDR8 0x3b8
```

21.21.1.225 CSR_PMPADDR9

```
#define CSR_PMPADDR9 0x3b9
```

21.21.1.226 CSR_PMPCFG0

```
#define CSR_PMPCFG0 0x3a0
```

21.21.1.227 CSR_PMPCFG1

```
#define CSR_PMPCFG1 0x3a1
```

21.21.1.228 CSR_PMPCFG2

```
#define CSR_PMPCFG2 0x3a2
```

21.21.1.229 CSR_PMPCFG3

```
#define CSR_PMPCFG3 0x3a3
```

21.21.1.230 CSR_SATP

```
#define CSR_SATP 0x180
```

21.21.1.231 CSR_SCAUSE

```
#define CSR_SCAUSE 0x142
```

21.21.1.232 CSR_SCOUNTEREN

```
#define CSR_SCOUNTEREN 0x106
```

21.21.1.233 CSR_SEPC

```
#define CSR_SEPC 0x141
```

21.21.1.234 CSR_SIE

```
#define CSR_SIE 0x104
```

21.21.1.235 CSR_SIP

```
#define CSR_SIP 0x144
```

21.21.1.236 CSR_SSCRATCH

```
#define CSR_SSCRATCH 0x140
```

21.21.1.237 CSR_SSTATUS

```
#define CSR_SSTATUS 0x100
```

21.21.1.238 CSR_STVAL

```
#define CSR_STVAL 0x143
```

21.21.1.239 CSR_STVEC

```
#define CSR_STVEC 0x105
```

21.21.1.240 CSR_TDATA1

```
#define CSR_TDATA1 0x7a1
```

21.21.1.241 CSR_TDATA2

```
#define CSR_TDATA2 0x7a2
```

21.21.1.242 CSR_TDATA3

```
#define CSR_TDATA3 0x7a3
```

21.21.1.243 CSR_TIME

```
#define CSR_TIME 0xc01
```

21.21.1.244 CSR_TIMEH

```
#define CSR_TIMEH 0xc81
```

21.21.1.245 CSR_TSELECT

```
#define CSR_TSELECT 0x7a0
```

21.21.1.246 CSR_UCAUSE

```
#define CSR_UCAUSE 0x42
```

21.21.1.247 CSR_UEPC

```
#define CSR_UEPC 0x41
```

21.21.1.248 CSR_UIE

```
#define CSR_UIE 0x4
```

21.21.1.249 CSR_UIP

```
#define CSR_UIP 0x44
```

21.21.1.250 CSR_USCRATCH

```
#define CSR_USCRATCH 0x40
```

21.21.1.251 CSR_USTATUS

```
#define CSR_USTATUS 0x0
```

21.21.1.252 CSR_UTVAL

```
#define CSR_UTVAL 0x43
```

21.21.1.253 CSR_UTVEC

```
#define CSR_UTVEC 0x5
```

21.21.1.254 CSR_VSATP

```
#define CSR_VSATP 0x280
```

21.21.1.255 CSR_VSCAUSE

```
#define CSR_VSCAUSE 0x242
```

21.21.1.256 CSR_VSEPC

```
#define CSR_VSEPC 0x241
```

21.21.1.257 CSR_VSIE

```
#define CSR_VSIE 0x204
```

21.21.1.258 CSR_VSIP

```
#define CSR_VSIP 0x244
```

21.21.1.259 CSR_VSSCRATCH

```
#define CSR_VSSCRATCH 0x240
```

21.21.1.260 CSR_VSSTATUS

```
#define CSR_VSSTATUS 0x200
```

21.21.1.261 CSR_VSTVAL

```
#define CSR_VSTVAL 0x243
```

21.21.1.262 CSR_VSTVEC

```
#define CSR_VSTVEC 0x205
```

21.21.1.263 GET_RM

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

21.21.1.264 GET_RS1

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

21.21.1.265 GET_RS1S

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

21.21.1.266 GET_RS2

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

21.21.1.267 GET_RS2C

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

21.21.1.268 GET_RS2S

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

21.21.1.269 GET_SP

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

21.21.1.270 HSTATUS_SP2P

```
#define HSTATUS_SP2P _UL(0x00000100)
```

21.21.1.271 HSTATUS_SP2V

```
#define HSTATUS_SP2V _UL(0x00000200)
```

21.21.1.272 HSTATUS_SPRV

```
#define HSTATUS_SPRV _UL(0x00000001)
```


21.21.1.273 HSTATUS_SPV

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

21.21.1.274 HSTATUS_VTSR

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

21.21.1.275 HSTATUS_VTVM

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

21.21.1.276 IMM_I

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

21.21.1.277 IMM_S

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

Value:

```
((s32)(insn) >> 25 << 5) | \  
    (s32)((insn) >> 7) & 0x1f))
```

21.21.1.278 INSN_16BIT_MASK

```
#define INSN_16BIT_MASK 0x3
```

21.21.1.279 INSN_32BIT_MASK

```
#define INSN_32BIT_MASK 0x1c
```

21.21.1.280 INSN_IS_16BIT

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

21.21.1.281 INSN_IS_32BIT

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

Value:

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

21.21.1.282 INSN_LEN

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

21.21.1.283 INSN_MASK_C_FLD

```
#define INSN_MASK_C_FLD 0xe003
```

21.21.1.284 INSN_MASK_C_FLDSP

```
#define INSN_MASK_C_FLDSP 0xe003
```

21.21.1.285 INSN_MASK_C_FLW

```
#define INSN_MASK_C_FLW 0xe003
```

21.21.1.286 INSN_MASK_C_FLWSP

```
#define INSN_MASK_C_FLWSP 0xe003
```

21.21.1.287 INSN_MASK_C_FSD

```
#define INSN_MASK_C_FSD 0xe003
```

21.21.1.288 INSN_MASK_C_FSDSP

```
#define INSN_MASK_C_FSDSP 0xe003
```

21.21.1.289 INSN_MASK_C_FSW

```
#define INSN_MASK_C_FSW 0xe003
```

21.21.1.290 INSN_MASK_C_FSWSP

```
#define INSN_MASK_C_FSWSP 0xe003
```

21.21.1.291 INSN_MASK_C_LD

```
#define INSN_MASK_C_LD 0xe003
```

21.21.1.292 INSN_MASK_C_LDSP

```
#define INSN_MASK_C_LDSP 0xe003
```

21.21.1.293 INSN_MASK_C_LW

```
#define INSN_MASK_C_LW 0xe003
```

21.21.1.294 INSN_MASK_C_LWSP

```
#define INSN_MASK_C_LWSP 0xe003
```

21.21.1.295 INSN_MASK_C_SD

```
#define INSN_MASK_C_SD 0xe003
```

21.21.1.296 INSN_MASK_C_SDSP

```
#define INSN_MASK_C_SDSP 0xe003
```

21.21.1.297 INSN_MASK_C_SW

```
#define INSN_MASK_C_SW 0xe003
```

21.21.1.298 INSN_MASK_C_SWSP

```
#define INSN_MASK_C_SWSP 0xe003
```

21.21.1.299 INSN_MASK_FLD

```
#define INSN_MASK_FLD 0x707f
```

21.21.1.300 INSN_MASK_FLQ

```
#define INSN_MASK_FLQ 0x707f
```

21.21.1.301 INSN_MASK_FLW

```
#define INSN_MASK_FLW 0x707f
```

21.21.1.302 INSN_MASK_FSD

```
#define INSN_MASK_FSD 0x707f
```

21.21.1.303 INSN_MASK_FSQ

```
#define INSN_MASK_FSQ 0x707f
```

21.21.1.304 INSN_MASK_FSW

```
#define INSN_MASK_FSW 0x707f
```

21.21.1.305 INSN_MASK_LB

```
#define INSN_MASK_LB 0x707f
```

21.21.1.306 INSN_MASK_LBU

```
#define INSN_MASK_LBU 0x707f
```

21.21.1.307 INSN_MASK_LD

```
#define INSN_MASK_LD 0x707f
```

21.21.1.308 INSN_MASK_LH

```
#define INSN_MASK_LH 0x707f
```

21.21.1.309 INSN_MASK_LHU

```
#define INSN_MASK_LHU 0x707f
```

21.21.1.310 INSN_MASK_LW

```
#define INSN_MASK_LW 0x707f
```

21.21.1.311 INSN_MASK_LWU

```
#define INSN_MASK_LWU 0x707f
```

21.21.1.312 INSN_MASK_SB

```
#define INSN_MASK_SB 0x707f
```

21.21.1.313 INSN_MASK_SD

```
#define INSN_MASK_SD 0x707f
```

21.21.1.314 INSN_MASK_SH

```
#define INSN_MASK_SH 0x707f
```

21.21.1.315 INSN_MASK_SW

```
#define INSN_MASK_SW 0x707f
```

21.21.1.316 INSN_MASK_WFI

```
#define INSN_MASK_WFI 0xffffffff00
```

21.21.1.317 INSN_MATCH_C_FLD

```
#define INSN_MATCH_C_FLD 0x2000
```

21.21.1.318 INSN_MATCH_C_FLDSP

```
#define INSN_MATCH_C_FLDSP 0x2002
```

21.21.1.319 INSN_MATCH_C_FLW

```
#define INSN_MATCH_C_FLW 0x6000
```

21.21.1.320 INSN_MATCH_C_FLWSP

```
#define INSN_MATCH_C_FLWSP 0x6002
```

21.21.1.321 INSN_MATCH_C_FSD

```
#define INSN_MATCH_C_FSD 0xa000
```

21.21.1.322 INSN_MATCH_C_FSDSP

```
#define INSN_MATCH_C_FSDSP 0xa002
```

21.21.1.323 INSN_MATCH_C_FSW

```
#define INSN_MATCH_C_FSW 0xe000
```

21.21.1.324 INSN_MATCH_C_FSWSP

```
#define INSN_MATCH_C_FSWSP 0xe002
```

21.21.1.325 INSN_MATCH_C_LD

```
#define INSN_MATCH_C_LD 0x6000
```

21.21.1.326 INSN_MATCH_C_LDSP

```
#define INSN_MATCH_C_LDSP 0x6002
```

21.21.1.327 INSN_MATCH_C_LW

```
#define INSN_MATCH_C_LW 0x4000
```

21.21.1.328 INSN_MATCH_C_LWSP

```
#define INSN_MATCH_C_LWSP 0x4002
```

21.21.1.329 INSN_MATCH_C_SD

```
#define INSN_MATCH_C_SD 0xe000
```

21.21.1.330 INSN_MATCH_C_SDSP

```
#define INSN_MATCH_C_SDSP 0xe002
```

21.21.1.331 INSN_MATCH_C_SW

```
#define INSN_MATCH_C_SW 0xc000
```

21.21.1.332 INSN_MATCH_C_SWSP

```
#define INSN_MATCH_C_SWSP 0xc002
```

21.21.1.333 INSN_MATCH_FLD

```
#define INSN_MATCH_FLD 0x3007
```


21.21.1.334 INSN_MATCH_FLQ

```
#define INSN_MATCH_FLQ 0x4007
```

21.21.1.335 INSN_MATCH_FLW

```
#define INSN_MATCH_FLW 0x2007
```

21.21.1.336 INSN_MATCH_FSD

```
#define INSN_MATCH_FSD 0x3027
```

21.21.1.337 INSN_MATCH_FSQ

```
#define INSN_MATCH_FSQ 0x4027
```

21.21.1.338 INSN_MATCH_FSW

```
#define INSN_MATCH_FSW 0x2027
```

21.21.1.339 INSN_MATCH_LB

```
#define INSN_MATCH_LB 0x3
```

21.21.1.340 INSN_MATCH_LBU

```
#define INSN_MATCH_LBU 0x4003
```

21.21.1.341 INSN_MATCH_LD

```
#define INSN_MATCH_LD 0x3003
```

21.21.1.342 INSN_MATCH_LH

```
#define INSN_MATCH_LH 0x1003
```

21.21.1.343 INSN_MATCH_LHU

```
#define INSN_MATCH_LHU 0x5003
```

21.21.1.344 INSN_MATCH_LW

```
#define INSN_MATCH_LW 0x2003
```

21.21.1.345 INSN_MATCH_LWU

```
#define INSN_MATCH_LWU 0x6003
```

21.21.1.346 INSN_MATCH_SB

```
#define INSN_MATCH_SB 0x23
```

21.21.1.347 INSN_MATCH_SD

```
#define INSN_MATCH_SD 0x3023
```

21.21.1.348 INSN_MATCH_SH

```
#define INSN_MATCH_SH 0x1023
```

21.21.1.349 INSN_MATCH_SW

```
#define INSN_MATCH_SW 0x2023
```

21.21.1.350 INSN_MATCH_WFI

```
#define INSN_MATCH_WFI 0x10500000
```

21.21.1.351 IRQ_M_EXT

```
#define IRQ_M_EXT 11
```

21.21.1.352 IRQ_M_SOFT

```
#define IRQ_M_SOFT 3
```

21.21.1.353 IRQ_M_TIMER

```
#define IRQ_M_TIMER 7
```

21.21.1.354 IRQ_S_EXT

```
#define IRQ_S_EXT 9
```

21.21.1.355 IRQ_S_GEXT

```
#define IRQ_S_GEXT 12
```

21.21.1.356 IRQ_S_SOFT

```
#define IRQ_S_SOFT 1
```

21.21.1.357 IRQ_S_TIMER

```
#define IRQ_S_TIMER 5
```

21.21.1.358 IRQ_VS_EXT

```
#define IRQ_VS_EXT 10
```

21.21.1.359 IRQ_VS_SOFT

```
#define IRQ_VS_SOFT 2
```

21.21.1.360 IRQ_VS_TIMER

```
#define IRQ_VS_TIMER 6
```

21.21.1.361 LOG_REGBYTES

```
#define LOG_REGBYTES 2
```

21.21.1.362 MASK_FUNC3

```
#define MASK_FUNC3 0x7000
```

21.21.1.363 MIP_MEIP

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

21.21.1.364 MIP_MSIP

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

21.21.1.365 MIP_MTIP

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

21.21.1.366 MIP_SEIP

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

21.21.1.367 MIP_SGEIP

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

21.21.1.368 MIP_SSIP

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

21.21.1.369 MIP_STIP

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

21.21.1.370 MIP_VSEIP

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

21.21.1.371 MIP_VSSIP

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

21.21.1.372 MIP_VSTIP

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

21.21.1.373 MSTATUS32_SD [1/2]

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

21.21.1.374 MSTATUS32_SD [2/2]

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

21.21.1.375 MSTATUS64_SD

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

21.21.1.376 MSTATUS_FS

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

21.21.1.377 MSTATUS_MIE

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

21.21.1.378 MSTATUS_MPIE

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

21.21.1.379 MSTATUS_MPP

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

21.21.1.380 MSTATUS_MPP_SHIFT

```
#define MSTATUS_MPP_SHIFT 11
```

21.21.1.381 MSTATUS_MPRV

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

21.21.1.382 MSTATUS_MXR

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

21.21.1.383 MSTATUS_SD

```
#define MSTATUS_SD MSTATUS32_SD
```

21.21.1.384 MSTATUS_SIE

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

21.21.1.385 MSTATUS_SPIE

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

21.21.1.386 MSTATUS_SPIE_SHIFT

```
#define MSTATUS_SPIE_SHIFT 5
```

21.21.1.387 MSTATUS_SPP

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

21.21.1.388 MSTATUS_SPP_SHIFT

```
#define MSTATUS_SPP_SHIFT 8
```

21.21.1.389 MSTATUS_SUM

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

21.21.1.390 MSTATUS_TSR

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

21.21.1.391 MSTATUS_TVM

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

21.21.1.392 MSTATUS_TW

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

21.21.1.393 MSTATUS_UBE

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

21.21.1.394 MSTATUS_XS

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

21.21.1.395 MSTATUSH_MBE

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

21.21.1.396 MSTATUSH_MPV

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

21.21.1.397 MSTATUSH_SBE

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


21.21.1.398 PMP_A

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

21.21.1.399 PMP_A_NA4

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

21.21.1.400 PMP_A_NAPOT

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

21.21.1.401 PMP_A_TOR

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

21.21.1.402 PMP_COUNT

```
#define PMP_COUNT 16
```

21.21.1.403 PMP_L

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

21.21.1.404 PMP_R

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

21.21.1.405 PMP_SHIFT

```
#define PMP_SHIFT 2
```

21.21.1.406 PMP_W

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

21.21.1.407 PMP_X

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

21.21.1.408 PRV_M

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

21.21.1.409 PRV_S

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

21.21.1.410 PRV_U

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

21.21.1.411 PTE_A

```
#define PTE_A _UL(0x040) /* Accessed */
```

21.21.1.412 PTE_D

```
#define PTE_D _UL(0x080) /* Dirty */
```

21.21.1.413 PTE_G

```
#define PTE_G _UL(0x020) /* Global */
```

21.21.1.414 PTE_PPN_SHIFT

```
#define PTE_PPN_SHIFT 10
```

21.21.1.415 PTE_R

```
#define PTE_R _UL(0x002) /* Read */
```

21.21.1.416 PTE_SOFT

```
#define PTE_SOFT _UL(0x300) /* Reserved for Software */
```

21.21.1.417 PTE_TABLE

```
#define PTE_TABLE(  
    PTE ) (((PTE) & (PTE_V | PTE_R | PTE_W | PTE_X)) == PTE_V)
```

21.21.1.418 PTE_U

```
#define PTE_U _UL(0x010) /* User */
```

21.21.1.419 PTE_V

```
#define PTE_V _UL(0x001) /* Valid */
```

21.21.1.420 PTE_W

```
#define PTE_W _UL(0x004) /* Write */
```

21.21.1.421 PTE_X

```
#define PTE_X _UL(0x008) /* Execute */
```

21.21.1.422 REG_MASK

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

21.21.1.423 REG_OFFSET

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

21.21.1.424 REG_PTR

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

21.21.1.425 REGBYTES

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

21.21.1.426 RISC_V_PGLEVEL_BITS

```
#define RISC_V_PGLEVEL_BITS 10
```

21.21.1.427 RISC_V_PGSHIFT

```
#define RISC_V_PGSHIFT 12
```

21.21.1.428 RISCV_PGSIZE

```
#define RISCV_PGSIZE (1 << RISCV_PGSHIFT)
```

21.21.1.429 RV_X

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

21.21.1.430 RVC_LD_IMM

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

Value:

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

21.21.1.431 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))
```

21.21.1.432 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))
```

21.21.1.433 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))
```

21.21.1.434 RVC_RS1S

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

21.21.1.435 RVC_RS2

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

21.21.1.436 RVC_RS2S

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

21.21.1.437 RVC_SDSP_IMM

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

Value:

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

21.21.1.438 RVC_SWSP_IMM

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

Value:

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

21.21.1.439 SATP32_ASID

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

21.21.1.440 SATP32_MODE

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

21.21.1.441 SATP32_PPN

```
#define SATP32_PPN _UL(0x003FFFFFF)
```

21.21.1.442 SATP64_ASID

```
#define SATP64_ASID _ULL(0x0FFFFF000000000000)
```

21.21.1.443 SATP64_MODE

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

21.21.1.444 SATP64_PPN

```
#define SATP64_PPN _ULL(0x0000FFFFFFFFFFFFFF)
```

21.21.1.445 SATP_MODE

```
#define SATP_MODE SATP32_MODE
```

21.21.1.446 SATP_MODE_OFF

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

21.21.1.447 SATP_MODE_SV32

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

21.21.1.448 SATP_MODE_SV39

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

21.21.1.449 SATP_MODE_SV48

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

21.21.1.450 SATP_MODE_SV57

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

21.21.1.451 SATP_MODE_SV64

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


21.21.1.452 SET_RD

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

21.21.1.453 SH_RD

```
#define SH_RD 7
```

21.21.1.454 SH_RS1

```
#define SH_RS1 15
```

21.21.1.455 SH_RS2

```
#define SH_RS2 20
```

21.21.1.456 SH_RS2C

```
#define SH_RS2C 2
```

21.21.1.457 SHIFT_RIGHT

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

21.21.1.458 SIP_SSIP

```
#define SIP_SSIP MIP_SSIP
```

21.21.1.459 SIP_STIP

```
#define SIP_STIP MIP_STIP
```

21.21.1.460 SSTATUS32_SD

```
#define SSTATUS32_SD MSTATUS32_SD
```

21.21.1.461 SSTATUS64_SD

```
#define SSTATUS64_SD MSTATUS64_SD
```

21.21.1.462 SSTATUS64_UXL

```
#define SSTATUS64_UXL MSTATUS_UXL
```

21.21.1.463 SSTATUS_FS

```
#define SSTATUS_FS MSTATUS_FS
```

21.21.1.464 SSTATUS_MXR

```
#define SSTATUS_MXR MSTATUS_MXR
```

21.21.1.465 SSTATUS_SD

```
#define SSTATUS_SD SSTATUS32_SD
```

21.21.1.466 SSTATUS_SIE

```
#define SSTATUS_SIE MSTATUS_SIE
```

21.21.1.467 SSTATUS_SPIE

```
#define SSTATUS_SPIE MSTATUS_SPIE
```

21.21.1.468 SSTATUS_SPIE_SHIFT

```
#define SSTATUS_SPIE_SHIFT MSTATUS_SPIE_SHIFT
```

21.21.1.469 SSTATUS_SPP

```
#define SSTATUS_SPP MSTATUS_SPP
```

21.21.1.470 SSTATUS_SPP_SHIFT

```
#define SSTATUS_SPP_SHIFT MSTATUS_SPP_SHIFT
```

21.21.1.471 SSTATUS_SUM

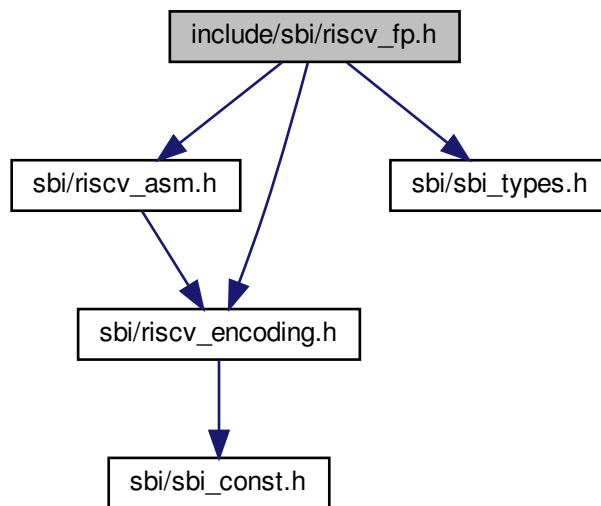
```
#define SSTATUS_SUM MSTATUS_SUM
```

21.21.1.472 SSTATUS_XS

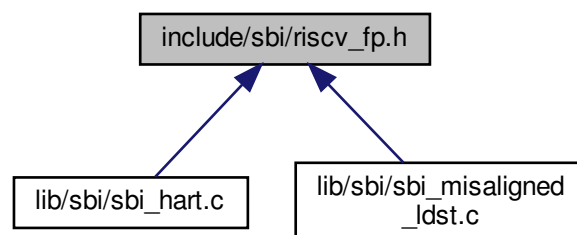
```
#define SSTATUS_XS MSTATUS_XS
```

21.22 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`

21.22.1 Macro Definition Documentation

21.22.1.1 GET_PRECISION

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

21.22.1.2 GET_RM

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

21.22.1.3 PRECISION_D

```
#define PRECISION_D 1
```

21.22.1.4 PRECISION_S

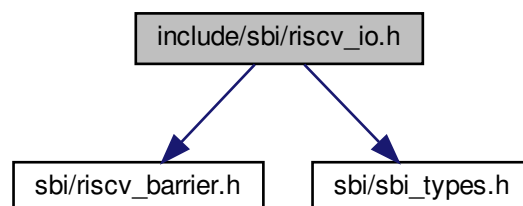
```
#define PRECISION_S 0
```

21.23 include/sbi/riscv_io.h File Reference

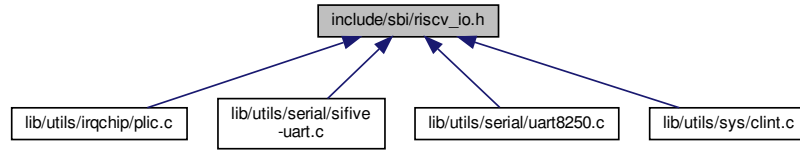
```
#include <sbi/riscv_barrier.h>
```

```
#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)

21.23.1 Macro Definition Documentation

21.23.1.1 __io_ar

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

21.23.1.2 __io_aw

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

21.23.1.3 __io_br

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

21.23.1.4 __io_bw

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

21.23.1.5 __io_rar

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

21.23.1.6 __io_raw

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

21.23.1.7 __io_rbr

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

21.23.1.8 __io_rbw

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

21.23.1.9 readb

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

21.23.1.10 readb_relaxed

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

21.23.1.11 readl

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

21.23.1.12 readl_relaxed

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

21.23.1.13 readq

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

21.23.1.14 readq_relaxed

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


21.23.1.15 readw

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

21.23.1.16 readw_relaxed

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

21.23.1.17 writeb

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

21.23.1.18 writeb_relaxed

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

21.23.1.19 writel

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

21.23.1.20 writel_relaxed

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

21.23.1.21 writeq

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

21.23.1.22 writeq_relaxed

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

21.23.1.23 writew

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

21.23.1.24 writew_relaxed

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

21.23.2 Function Documentation

21.23.2.1 __raw_readb()

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

21.23.2.2 __raw_readl()

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

21.23.2.3 __raw_readq()

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

21.23.2.4 __raw_readw()

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

21.23.2.5 __raw_writeb()

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

21.23.2.6 __raw_writel()

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

21.23.2.7 __raw_writeq()

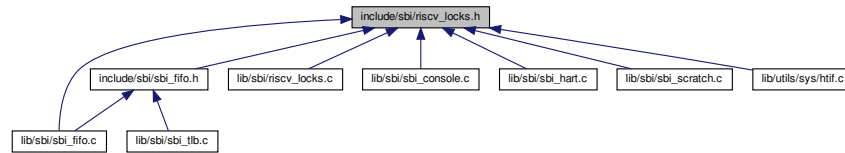
```
static void __raw_writeq (  
    u64 val,  
    volatile void * addr ) [inline], [static]
```

21.23.2.8 __raw_writew()

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

21.24 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)

21.24.1 Macro Definition Documentation

21.24.1.1 __RISCV_SPIN_UNLOCKED

```
#define __RISCV_SPIN_UNLOCKED 0
```

21.24.1.2 SPIN_LOCK_INIT

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

21.24.1.3 SPIN_LOCK_INITIALIZER

```
#define SPIN_LOCK_INITIALIZER
```

Value:

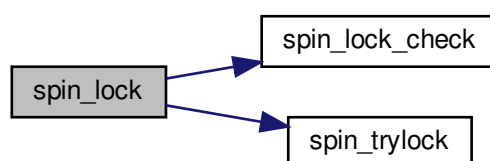
```
{  
    .lock = __RISCV_SPIN_UNLOCKED,  
}
```

21.24.2 Function Documentation

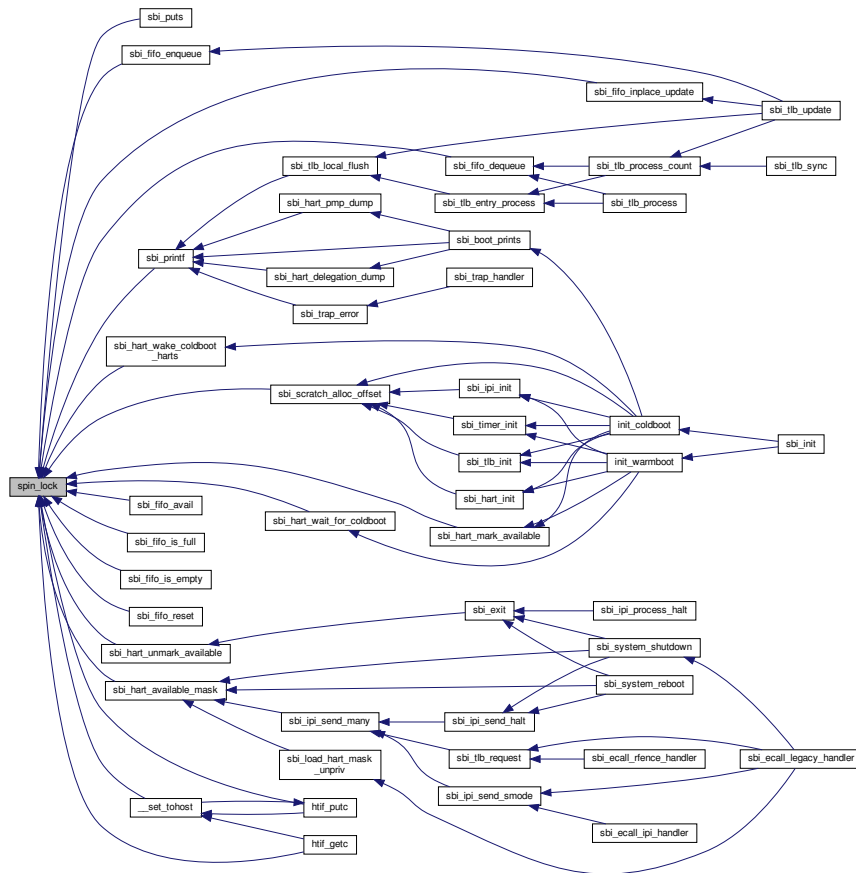
21.24.2.1 spin_lock()

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

Here is the call graph for this function:



Here is the caller graph for this function:

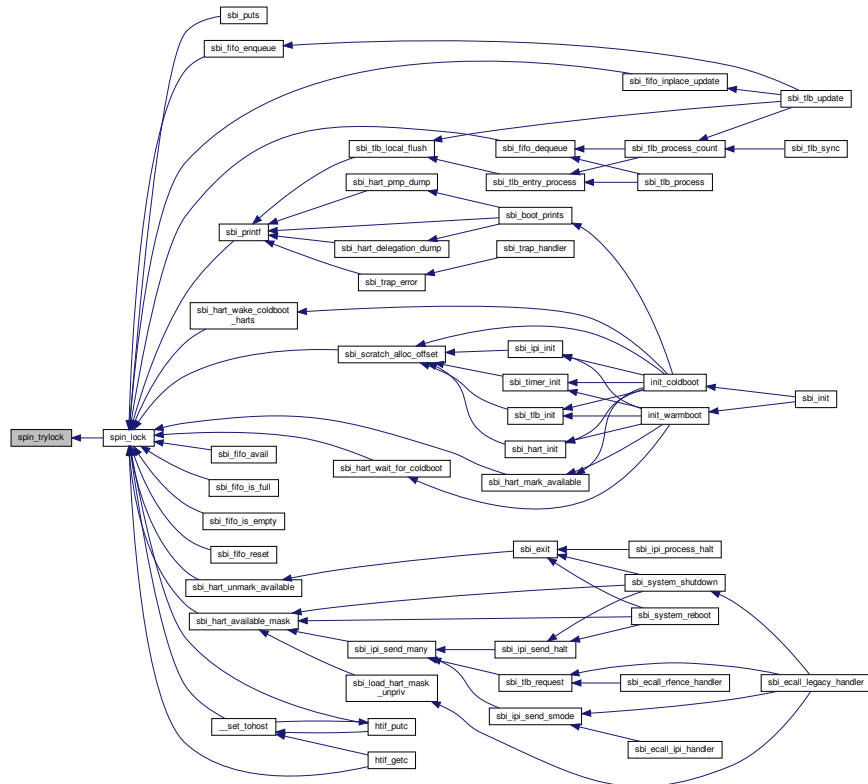


21.24.2.2 spin_lock_check()

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

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

Here is the caller graph for this function:

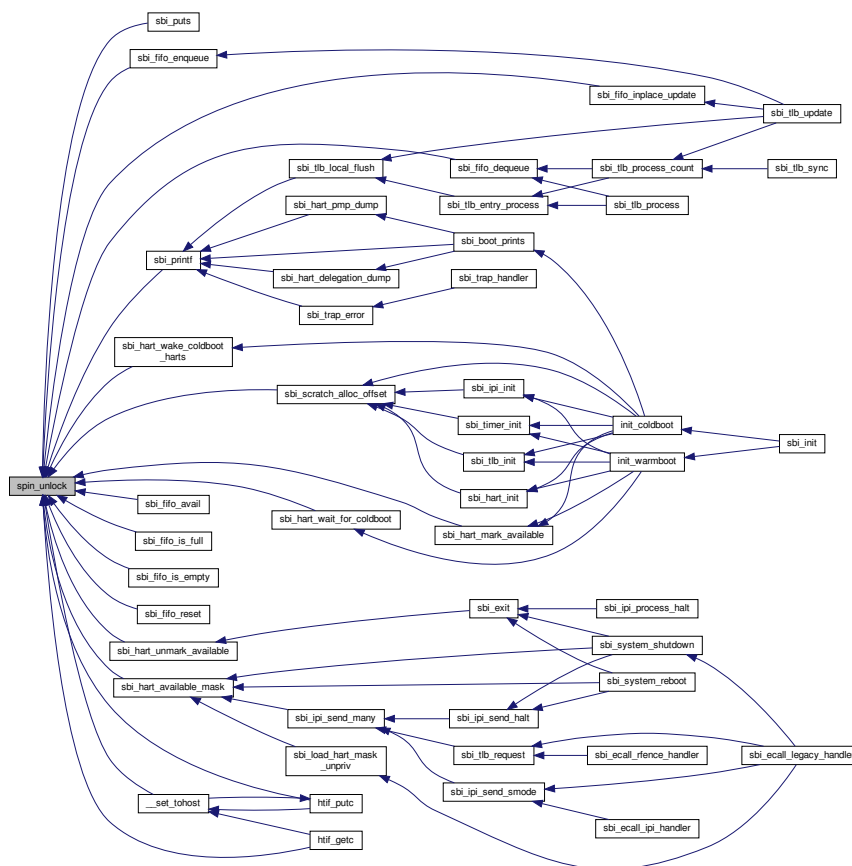


21.24.2.4 spin_unlock()

```

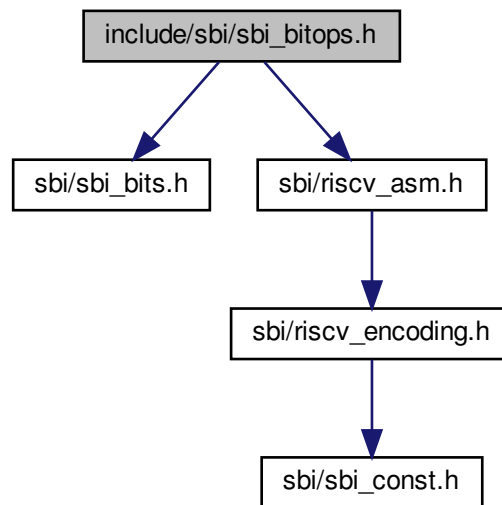
void spin_unlock (
    spinlock_t * lock )

```

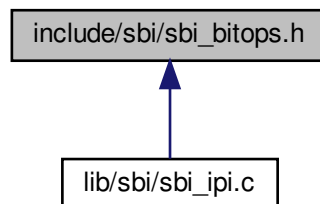



```
#include <sbi/sbi_bits.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_bitops.h:



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



Macros

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

Functions

- static int `ffs` (int x)
- static int `__ffs` (unsigned long word)
- static int `fls` (int x)
- static unsigned long `__fls` (unsigned long word)

21.25.1 Macro Definition Documentation

21.25.1.1 ffz

```
#define ffz(  
    x ) \_\_ffs(~(x))
```

21.25.2 Function Documentation

21.25.2.1 __ffs()

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

`__ffs` - find first bit in word. : The word to search

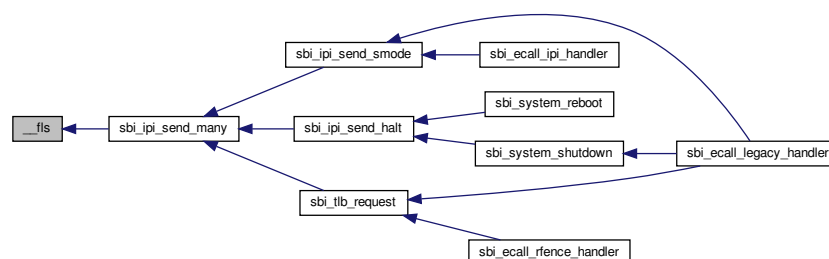
Undefined if no bit exists, so code should check against 0 first.

21.25.2.2 __fls()

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

`__fls` - find last (most-significant) set bit in a long 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:



21.25.2.3 ffs()

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

ffs - Find first bit set : 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).

21.25.2.4 fls()

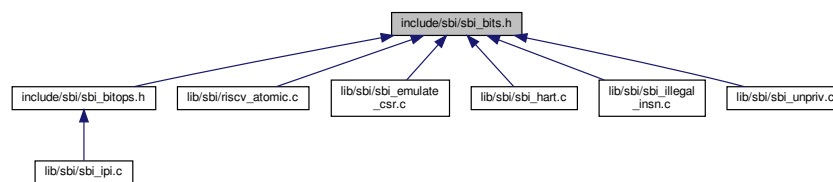
```
static int fls (
    int x )    [inline], [static]
```

fls - find last (most-significant) bit set : the word to search

This is defined the same way as ffs. Note fls(0) = 0, fls(1) = 1, fls(0x80000000) = 32.

21.26 include/sbi/sbi_bits.h File Reference

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 [BIT_MASK](#)(nr) (1UL << ((nr) % BITS_PER_LONG))
- #define [BIT_WORD](#)(nr) ((nr) / BITS_PER_LONG)

21.26.1 Macro Definition Documentation

21.26.1.1 BIT_MASK

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

21.26.1.2 BIT_WORD

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

21.26.1.3 EXTRACT_FIELD

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

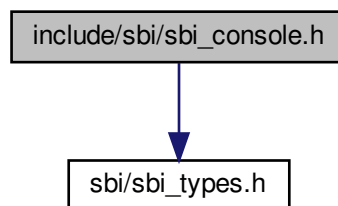
21.26.1.4 INSERT_FIELD

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

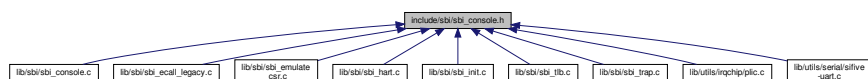
21.27 include/sbi/sbi_console.h File Reference

```
#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 const char int sbi_console_init` (struct `sbi_scratch` *scratch)

Variables

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

21.27.1 Macro Definition Documentation

21.27.1.1 __printf

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

21.27.2 Function Documentation

21.27.2.1 __printf() [1/3]

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

21.27.2.2 __printf() [2/3]

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

21.27.2.3 __printf() [3/3]

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

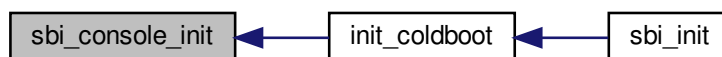
21.27.2.4 sbi_console_init()

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

Here is the call graph for this function:



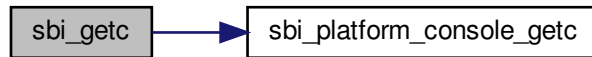
Here is the caller graph for this function:



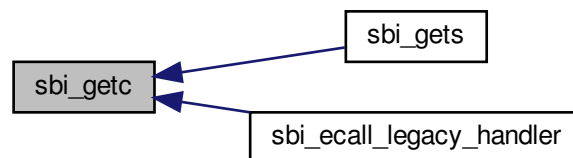
21.27.2.5 sbi_getc()

```
int sbi_getc (  
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.27.2.6 sbi_gets()

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

Here is the call graph for this function:




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

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

```
graph LR; sbi_putc --> sbi_platform_console_putc
```

[illegible]

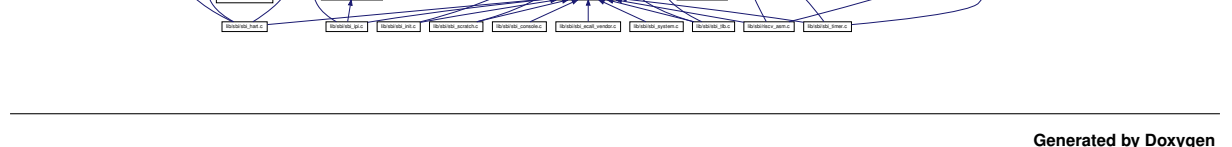
```
void sbi_puts (
    const char * str )
```

```
int const char * format
```

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

21.28 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)`

21.28.1 Macro Definition Documentation

21.28.1.1 __AC

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

21.28.1.2 __STR

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

21.28.1.3 _AC

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

21.28.1.4 _AT

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

21.28.1.5 _BITUL

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

21.28.1.6 _BITULL

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

21.28.1.7 _UL

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

21.28.1.8 _ULL

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

21.28.1.9 STRINGIFY

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

21.28.1.10 UL

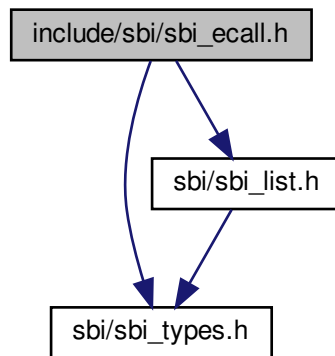
```
#define UL(  
    x ) ( _UL(x) )
```

21.28.1.11 ULL

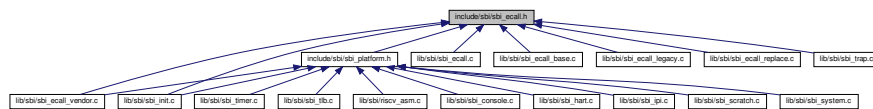
```
#define ULL(  
    x ) ( _ULL(x) )
```

21.29 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)
- 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](#) (u32 hartid, ulong mcause, struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch)
- 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`

21.29.1 Macro Definition Documentation

21.29.1.1 SBI_ECALL_VERSION_MAJOR

```
#define SBI_ECALL_VERSION_MAJOR 0
```

21.29.1.2 SBI_ECALL_VERSION_MINOR

```
#define SBI_ECALL_VERSION_MINOR 2
```

21.29.1.3 SBI_OPENSBI_IMPID

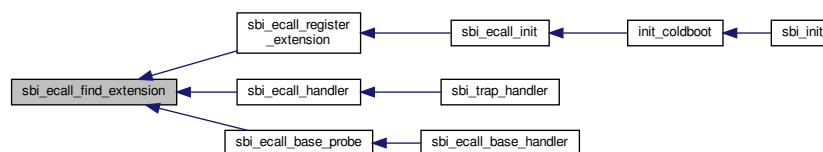
```
#define SBI_OPENSBI_IMPID 1
```

21.29.2 Function Documentation

21.29.2.1 sbi_ecall_find_extension()

```
struct sbi\_ecall\_extension* sbi_ecall_find_extension (
    unsigned long extid )
```

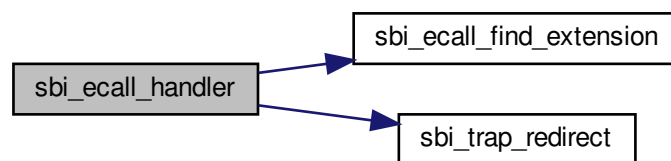
Here is the caller graph for this function:



21.29.2.2 sbi_ecall_handler()

```
int sbi_ecall_handler (
    u32 hartid,
    ulong mcause,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:



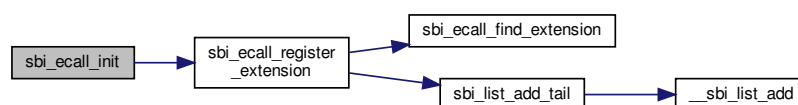
Here is the caller graph for this function:



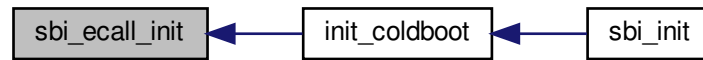
21.29.2.3 sbi_ecall_init()

```
int sbi_ecall_init (
    void )
```

Here is the call graph for this function:



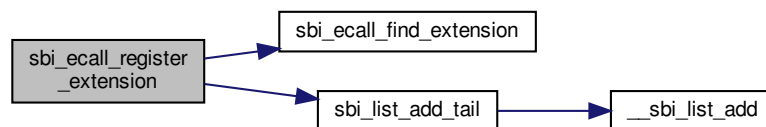
Here is the caller graph for this function:



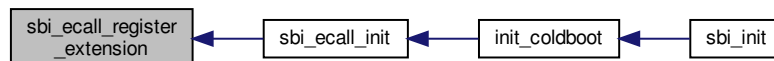
21.29.2.4 sbi_ecall_register_extension()

```
int sbi_ecall_register_extension (
    struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:



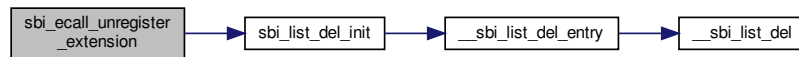
Here is the caller graph for this function:



21.29.2.5 sbi_ecall_unregister_extension()

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

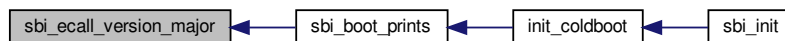

Here is the call graph for this function:



21.29.2.6 sbi_ecall_version_major()

```
u16 sbi_ecall_version_major (
    void )
```

Here is the caller graph for this function:



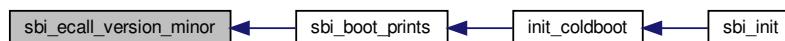
21.29.2.7 sbi_ecall_version_minor()

```
u16 sbi_ecall_version_minor (
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.29.3 Variable Documentation

21.29.3.1 ecall_base

struct `sbi_ecall_extension` `ecall_base`

21.29.3.2 ecall_ipi

struct `sbi_ecall_extension` `ecall_ipi`

21.29.3.3 ecall_legacy

struct `sbi_ecall_extension` `ecall_legacy`

21.29.3.4 ecall_rfence

struct `sbi_ecall_extension` `ecall_rfence`

21.29.3.5 ecall_time

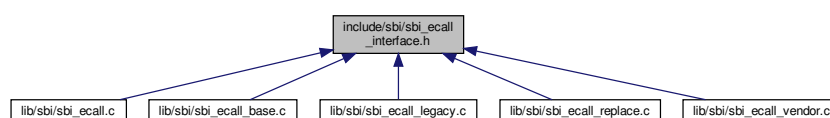
struct `sbi_ecall_extension` `ecall_time`

21.29.3.6 ecall_vendor

struct `sbi_ecall_extension` `ecall_vendor`

21.30 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_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_SPEC_VERSION_MAJOR_OFFSET 24`
- `#define SBI_SPEC_VERSION_MAJOR_MASK 0x7f`
- `#define SBI_SPEC_VERSION_MINOR_MASK 0xffff`
- `#define SBI_EXT_VENDOR_START 0x09000000`
- `#define SBI_EXT_VENDOR_END 0x09FFFFFF`

21.30.1 Macro Definition Documentation

21.30.1.1 SBI_EXT_0_1_CLEAR_IPI

```
#define SBI_EXT_0_1_CLEAR_IPI 0x3
```

21.30.1.2 SBI_EXT_0_1_CONSOLE_GETCHAR

```
#define SBI_EXT_0_1_CONSOLE_GETCHAR 0x2
```

21.30.1.3 SBI_EXT_0_1_CONSOLE_PUTCHAR

```
#define SBI_EXT_0_1_CONSOLE_PUTCHAR 0x1
```

21.30.1.4 SBI_EXT_0_1_REMOTE_FENCE_I

```
#define SBI_EXT_0_1_REMOTE_FENCE_I 0x5
```

21.30.1.5 SBI_EXT_0_1_REMOTE_SFENCE_VMA

```
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA 0x6
```

21.30.1.6 SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID

```
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID 0x7
```

21.30.1.7 SBI_EXT_0_1_SEND_IPI

```
#define SBI_EXT_0_1_SEND_IPI 0x4
```

21.30.1.8 SBI_EXT_0_1_SET_TIMER

```
#define SBI_EXT_0_1_SET_TIMER 0x0
```

21.30.1.9 SBI_EXT_0_1_SHUTDOWN

```
#define SBI_EXT_0_1_SHUTDOWN 0x8
```

21.30.1.10 SBI_EXT_BASE

```
#define SBI_EXT_BASE 0x10
```

21.30.1.11 SBI_EXT_BASE_GET_IMP_ID

```
#define SBI_EXT_BASE_GET_IMP_ID 0x1
```

21.30.1.12 SBI_EXT_BASE_GET_IMP_VERSION

```
#define SBI_EXT_BASE_GET_IMP_VERSION 0x2
```

21.30.1.13 SBI_EXT_BASE_GET_MARCHID

```
#define SBI_EXT_BASE_GET_MARCHID 0x5
```

21.30.1.14 SBI_EXT_BASE_GET_MIMPID

```
#define SBI_EXT_BASE_GET_MIMPID 0x6
```

21.30.1.15 SBI_EXT_BASE_GET_MVENDORID

```
#define SBI_EXT_BASE_GET_MVENDORID 0x4
```

21.30.1.16 SBI_EXT_BASE_GET_SPEC_VERSION

```
#define SBI_EXT_BASE_GET_SPEC_VERSION 0x0
```

21.30.1.17 SBI_EXT_BASE_PROBE_EXT

```
#define SBI_EXT_BASE_PROBE_EXT 0x3
```

21.30.1.18 SBI_EXT_IPI

```
#define SBI_EXT_IPI 0x735049
```

21.30.1.19 SBI_EXT_IPI_SEND_IPI

```
#define SBI_EXT_IPI_SEND_IPI 0x0
```

21.30.1.20 SBI_EXT_RFENCE

```
#define SBI_EXT_RFENCE 0x52464E43
```

21.30.1.21 SBI_EXT_RFENCE_REMOTE_FENCE_I

```
#define SBI_EXT_RFENCE_REMOTE_FENCE_I 0x0
```

21.30.1.22 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA 0x3
```

21.30.1.23 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID 0x4
```

21.30.1.24 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA 0x5
```

21.30.1.25 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID 0x6
```

21.30.1.26 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA

```
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA 0x1
```

21.30.1.27 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID

```
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID 0x2
```

21.30.1.28 SBI_EXT_TIME

```
#define SBI_EXT_TIME 0x54494D45
```

21.30.1.29 SBI_EXT_TIME_SET_TIMER

```
#define SBI_EXT_TIME_SET_TIMER 0x0
```

21.30.1.30 SBI_EXT_VENDOR_END

```
#define SBI_EXT_VENDOR_END 0x09FFFFFF
```

21.30.1.31 SBI_EXT_VENDOR_START

```
#define SBI_EXT_VENDOR_START 0x09000000
```

21.30.1.32 SBI_SPEC_VERSION_MAJOR_MASK

```
#define SBI_SPEC_VERSION_MAJOR_MASK 0x7f
```

21.30.1.33 SBI_SPEC_VERSION_MAJOR_OFFSET

```
#define SBI_SPEC_VERSION_MAJOR_OFFSET 24
```

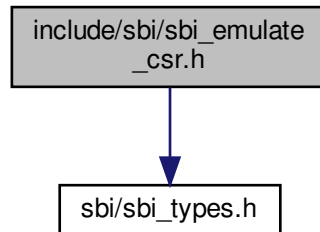
21.30.1.34 SBI_SPEC_VERSION_MINOR_MASK

```
#define SBI_SPEC_VERSION_MINOR_MASK 0xffffffff
```

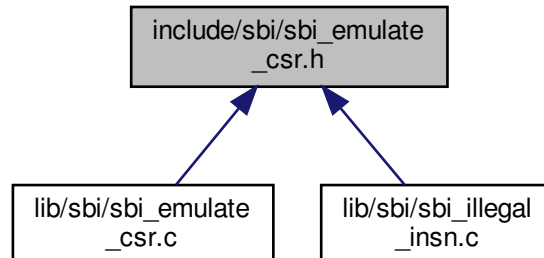
21.31 include/sbi/sbi_emulate_csr.h File Reference

```
#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

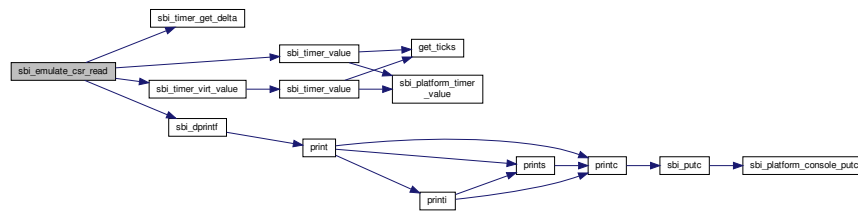
- `int sbi_emulate_csr_read` (int csr_num, [u32](#) hartid, struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch, [ulong](#) *csr_val)
- `int sbi_emulate_csr_write` (int csr_num, [u32](#) hartid, struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch, [ulong](#) csr_val)

21.31.1 Function Documentation

21.31.1.1 sbi_emulate_csr_read()

```
int sbi_emulate_csr_read (
    int csr_num,
    u32 hartid,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch,
    ulong * csr_val )
```

Here is the call graph for this function:



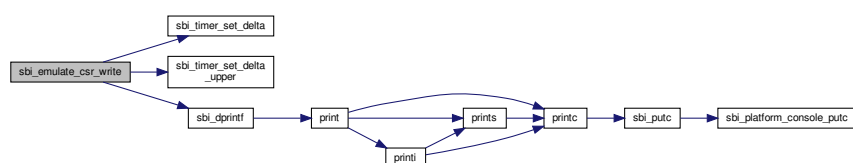
Here is the caller graph for this function:



21.31.1.2 sbi_emulate_csr_write()

```
int sbi_emulate_csr_write (
    int csr_num,
    u32 hartid,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch,
    ulong csr_val )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.32 include/sbi/sbi_error.h File Reference

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



Macros

- `#define SBI_OK 0`
- `#define SBI_EFAIL -1`
- `#define SBI_ENOTSUPP -2`
- `#define SBI_EINVAL -3`
- `#define SBI_DENIED -4`
- `#define SBI_INVALID_ADDR -5`
- `#define SBI_ENODEV -6`
- `#define SBI_ENOSYS -7`
- `#define SBI_ETIMEDOUT -8`
- `#define SBI_EIO -9`
- `#define SBI_EILL -10`
- `#define SBI_ENOSPC -11`
- `#define SBI_ENOMEM -12`
- `#define SBI_ETRAP -13`
- `#define SBI_EUNKNOWN -14`
- `#define SBI_ENOENT -15`

21.32.1 Macro Definition Documentation

21.32.1.1 SBI_DENIED

```
#define SBI_DENIED -4
```

21.32.1.2 SBI_EFAIL

```
#define SBI_EFAIL -1
```

21.32.1.3 SBI_EILL

```
#define SBI_EILL -10
```

21.32.1.4 SBI_EINVAL

```
#define SBI_EINVAL -3
```

21.32.1.5 SBI_EIO

```
#define SBI_EIO -9
```

21.32.1.6 SBI_ENODEV

```
#define SBI_ENODEV -6
```

21.32.1.7 SBI_ENOENT

```
#define SBI_ENOENT -15
```

21.32.1.8 SBI_ENOMEM

```
#define SBI_ENOMEM -12
```

21.32.1.9 SBI_ENOSPC

```
#define SBI_ENOSPC -11
```

21.32.1.10 SBI_ENOSYS

```
#define SBI_ENOSYS -7
```

21.32.1.11 SBI_ENOTSUPP

```
#define SBI_ENOTSUPP -2
```

21.32.1.12 SBI_ETIMEDOUT

```
#define SBI_ETIMEDOUT -8
```

21.32.1.13 SBI_ETRAP

```
#define SBI_ETRAP -13
```

21.32.1.14 SBI_EUNKNOWN

```
#define SBI_EUNKNOWN -14
```

21.32.1.15 SBI_INVALID_ADDR

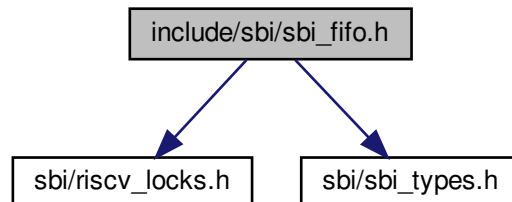
```
#define SBI_INVALID_ADDR -5
```

21.32.1.16 SBI_OK

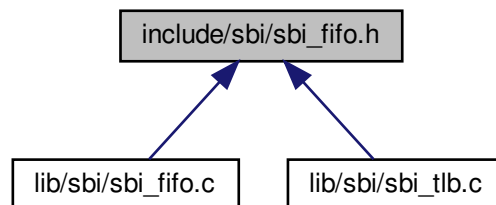
```
#define SBI_OK 0
```

21.33 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)

21.33.1 Enumeration Type Documentation

21.33.1.1 `sbi_fifo_inplace_update_types`

enum `sbi_fifo_inplace_update_types`

Enumerator

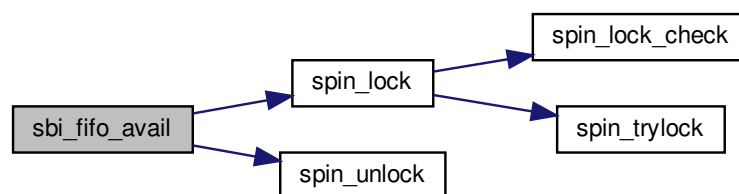
<code>SBI_FIFO_SKIP</code>	
<code>SBI_FIFO_UPDATED</code>	
<code>SBI_FIFO_UNCHANGED</code>	

21.33.2 Function Documentation

21.33.2.1 `sbi_fifo_avail()`

```
ul6 sbi_fifo_avail (
    struct sbi_fifo * fifo )
```

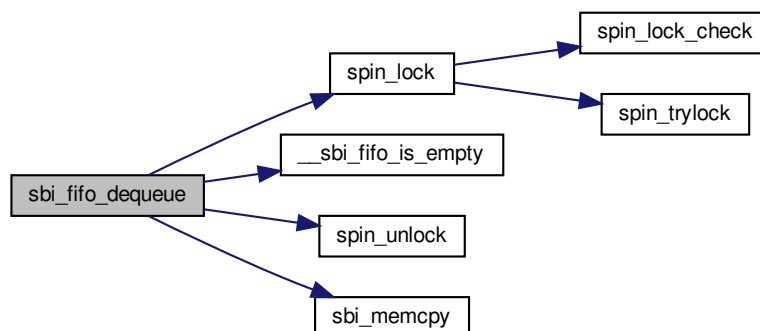
Here is the call graph for this function:



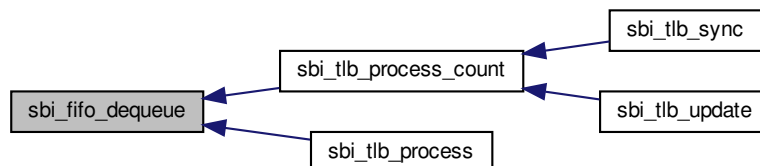
21.33.2.2 `sbi_fifo_dequeue()`

```
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:



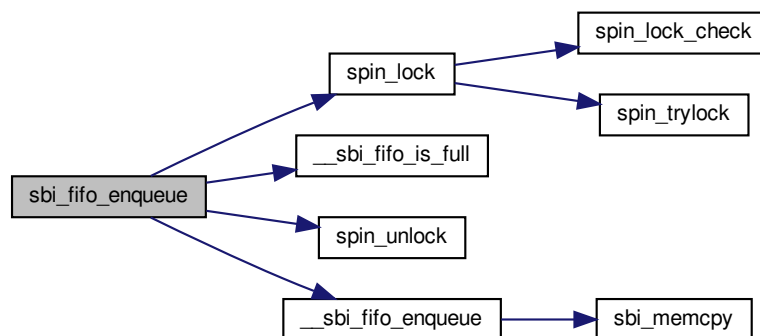
21.33.2.3 sbi_fifo_enqueue()

```

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:



21.33.2.4 sbi_fifo_init()

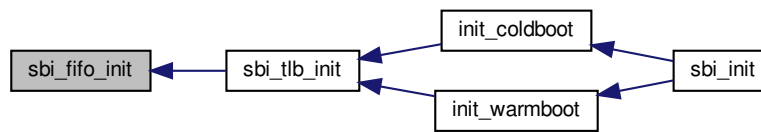
```

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:



21.33.2.5 sbi_fifo_inplace_update()

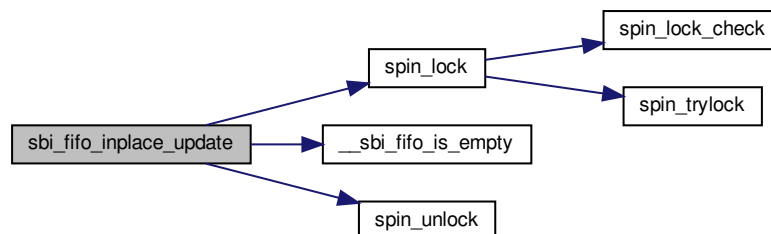
```

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:



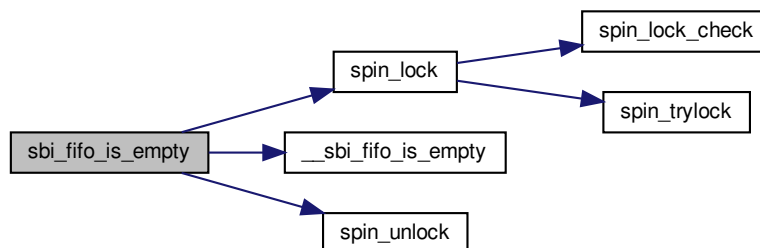
Here is the caller graph for this function:



21.33.2.6 sbi_fifo_is_empty()

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

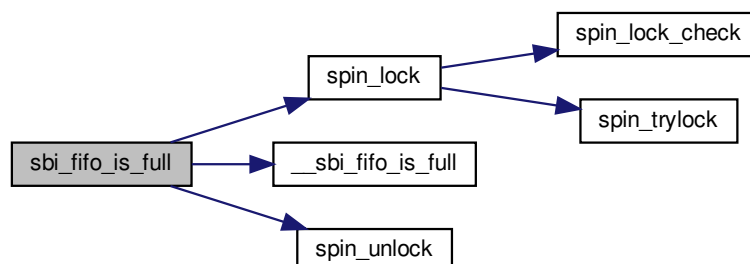
Here is the call graph for this function:



21.33.2.7 sbi_fifo_is_full()

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

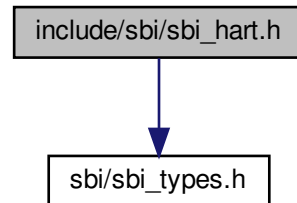
Here is the call graph for this function:



21.34 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:



Functions

- int [sbi_hart_init](#) (struct [sbi_scratch](#) *scratch, [u32](#) hartid, [bool](#) cold_boot)
- void * [sbi_hart_get_trap_info](#) (struct [sbi_scratch](#) *scratch)
- void [sbi_hart_set_trap_info](#) (struct [sbi_scratch](#) *scratch, void *data)
- void [sbi_hart_delegation_dump](#) (struct [sbi_scratch](#) *scratch)
- void [sbi_hart_pmp_dump](#) (struct [sbi_scratch](#) *scratch)
- void [__attribute__\(\(noreturn\)\) sbi_hart_hang](#)(void)
- void [sbi_hart_mark_available](#) ([u32](#) hartid)
- [ulong](#) [sbi_hart_available_mask](#) (void)
- void [sbi_hart_unmark_available](#) ([u32](#) hartid)
- struct [sbi_scratch](#) * [sbi_hart_id_to_scratch](#) (struct [sbi_scratch](#) *scratch, [u32](#) hartid)
- void [sbi_hart_wait_for_coldboot](#) (struct [sbi_scratch](#) *scratch, [u32](#) hartid)
- void [sbi_hart_wake_coldboot_harts](#) (struct [sbi_scratch](#) *scratch, [u32](#) hartid)
- [u32](#) [sbi_current_hartid](#) (void)

Variables

- 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](#)

21.34.1 Function Documentation

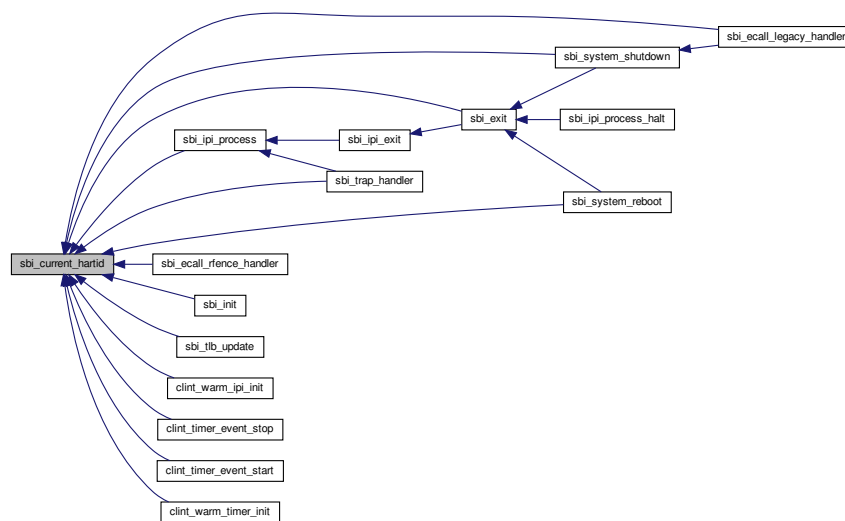
21.34.1.1 __attribute__((noreturn))

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

21.34.1.2 sbi_current_hartid()

```
u32 sbi_current_hartid (
    void )
```

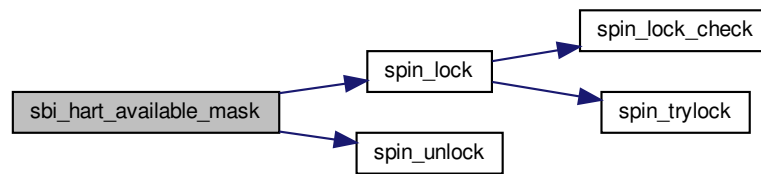
Return HART ID of the caller. Here is the caller graph for this function:



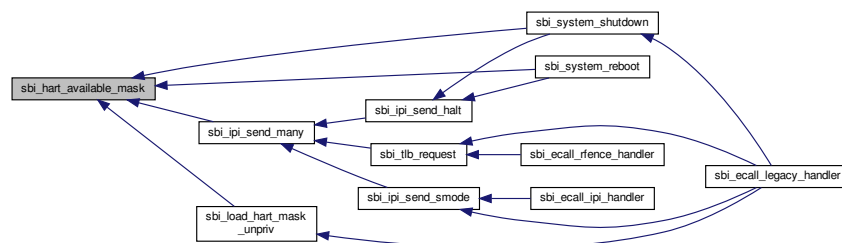
21.34.1.3 sbi_hart_available_mask()

```
ulong sbi_hart_available_mask (
    void )
```

Here is the call graph for this function:



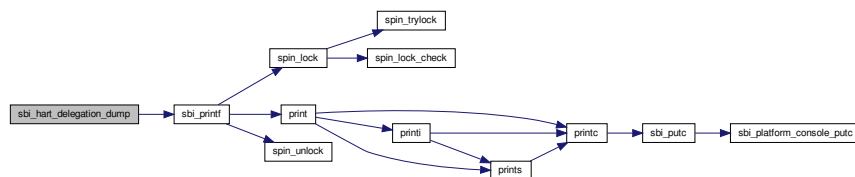
Here is the caller graph for this function:



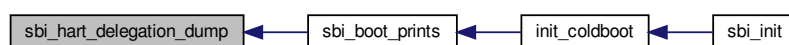
21.34.1.4 sbi_hart_delegation_dump()

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

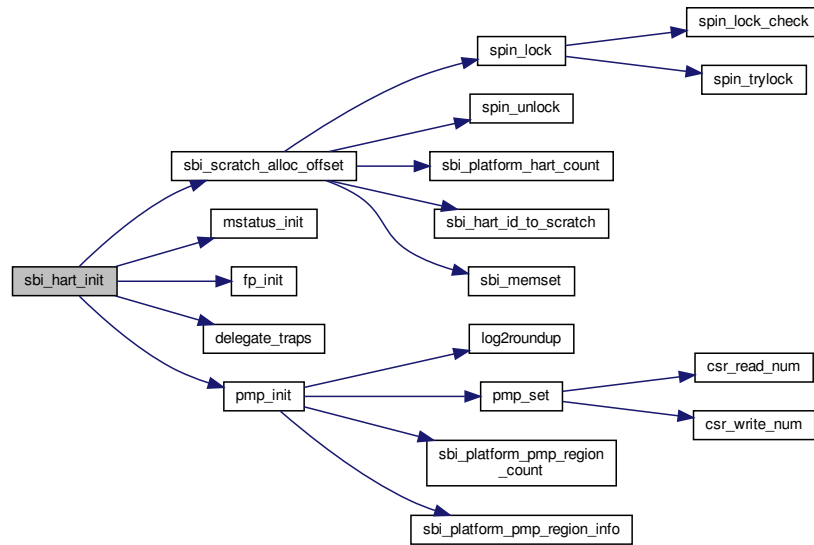
Here is the call graph for this function:



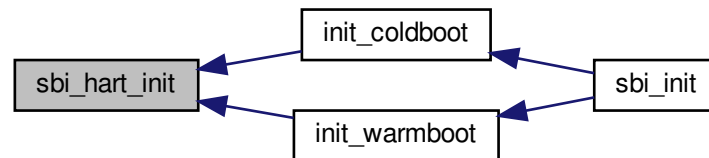
Here is the caller graph for this function:



Here is the call graph for this function:



Here is the caller graph for this function:

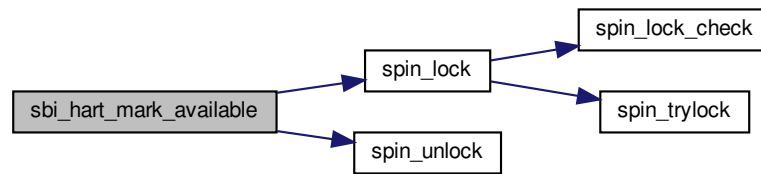


21.34.1.8 sbi_hart_mark_available()

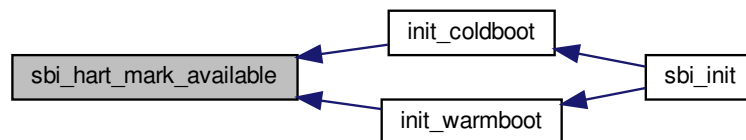
```

void sbi_hart_mark_available (
    u32 hartid )
  
```

Here is the call graph for this function:



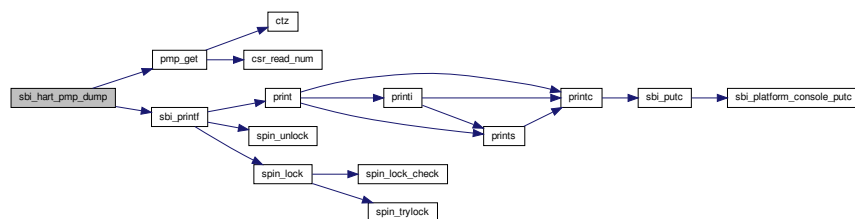
Here is the caller graph for this function:



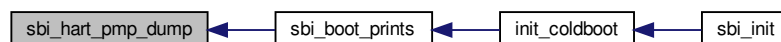
21.34.1.9 sbi_hart_pmp_dump()

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

Here is the call graph for this function:



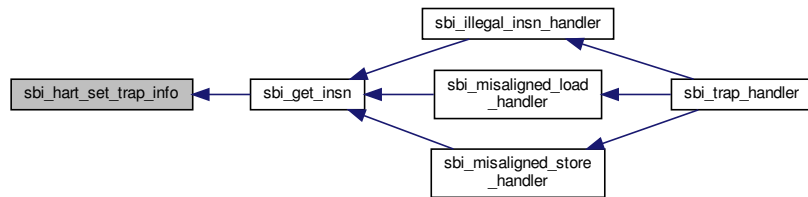
Here is the caller graph for this function:



21.34.1.10 sbi_hart_set_trap_info()

```
void sbi_hart_set_trap_info (
    struct sbi_scratch * scratch,
    void * data )
```

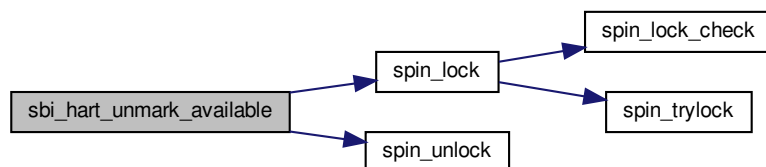
Here is the caller graph for this function:



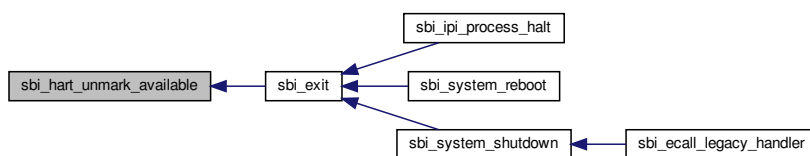
21.34.1.11 sbi_hart_unmark_available()

```
void sbi_hart_unmark_available (
    u32 hartid )
```

Here is the call graph for this function:



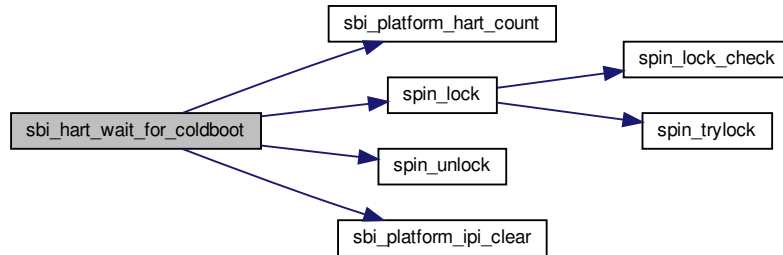
Here is the caller graph for this function:



21.34.1.12 sbi_hart_wait_for_coldboot()

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

Here is the call graph for this function:



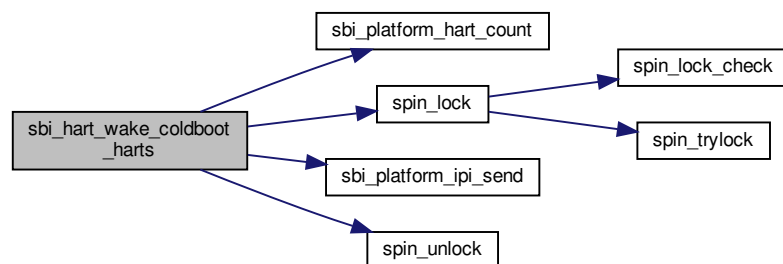
Here is the caller graph for this function:



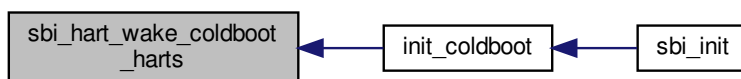
21.34.1.13 sbi_hart_wake_coldboot_harts()

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

Here is the call graph for this function:



Here is the caller graph for this function:



21.34.2 Variable Documentation

21.34.2.1 `arg1`

`void unsigned long arg1`

21.34.2.2 `next_addr`

`void unsigned long unsigned long next_addr`

21.34.2.3 `next_mode`

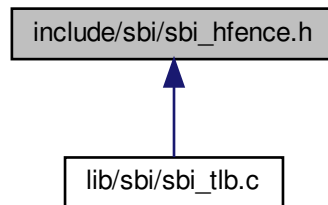
`void unsigned long unsigned long unsigned long next_mode`

21.34.2.4 `next_virt`

`void unsigned long unsigned long unsigned long bool next_virt`

21.35 include/sbi/sbi_hfence.h File Reference

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



Functions

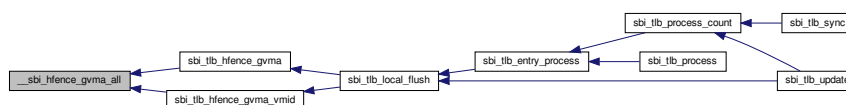
- void [__sbi_hfence_gvma_vmid_gpa](#) (unsigned long vmid, unsigned long gpa)
- void [__sbi_hfence_gvma_vmid](#) (unsigned long vmid)
- void [__sbi_hfence_gvma_gpa](#) (unsigned long gpa)
- void [__sbi_hfence_gvma_all](#) (void)
- void [__sbi_hfence_vvma_asid_va](#) (unsigned long asid, unsigned long va)
- void [__sbi_hfence_vvma_asid](#) (unsigned long asid)
- void [__sbi_hfence_vvma_va](#) (unsigned long va)
- void [__sbi_hfence_vvma_all](#) (void)

21.35.1 Function Documentation

21.35.1.1 __sbi_hfence_gvma_all()

```
void __sbi_hfence_gvma_all (
    void )
```

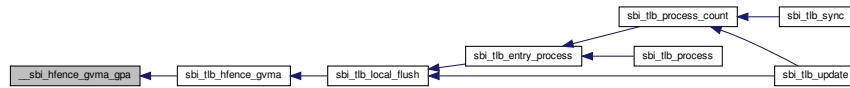
Invalidate all possible Stage2 TLBs Here is the caller graph for this function:



21.35.1.2 __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:



21.35.1.3 __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:



21.35.1.4 __sbi_hfence_gvma_vmid_gpa()

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

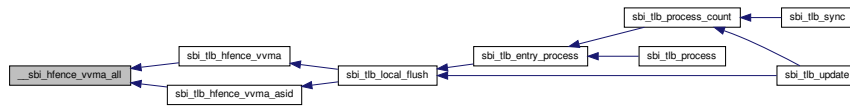
Invalidate Stage2 TLBs for given VMID and guest physical address Here is the caller graph for this function:



21.35.1.5 __sbi_hfence_vvma_all()

```
void __sbi_hfence_vvma_all (
    void )
```

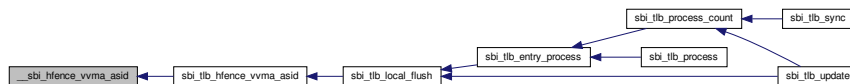
Invalidate all possible Stage2 TLBs Here is the caller graph for this function:



21.35.1.6 __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:



21.35.1.7 __sbi_hfence_vvma_asid_va()

```
void __sbi_hfence_vvma_asid_va (
    unsigned long asid,
    unsigned long va )
```

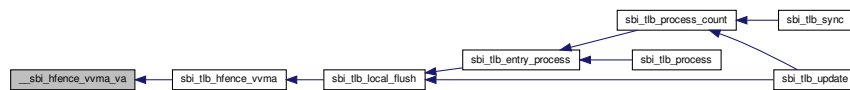
Invalidate unified TLB entries for given asid and guest virtual address Here is the caller graph for this function:



21.35.1.8 __sbi_hfence_vvma_va()

```
void __sbi_hfence_vvma_va (
    unsigned long va )
```

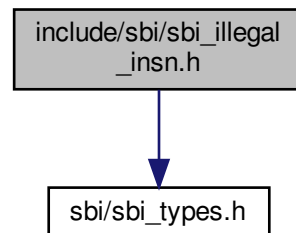
Invalidate unified TLB entries for a given guest virtual address Here is the caller graph for this function:



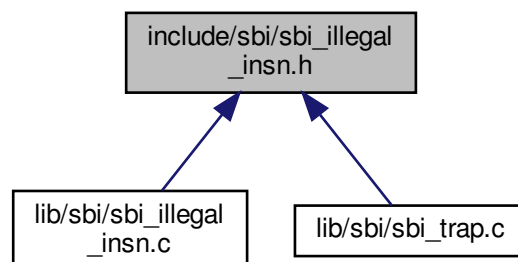
21.36 include/sbi/sbi_illegal_insn.h File Reference

```
#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

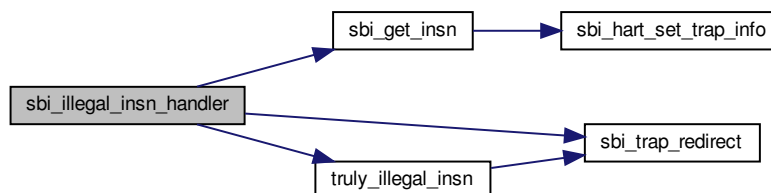
- int `sbi_illegal_insn_handler` (u32 hartid, ulong mcause, ulong insn, struct `sbi_trap_regs` *regs, struct `sbi_scratch` *scratch)

21.36.1 Function Documentation

21.36.1.1 `sbi_illegal_insn_handler()`

```
int sbi_illegal_insn_handler (
    u32 hartid,
    ulong mcause,
    ulong insn,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:



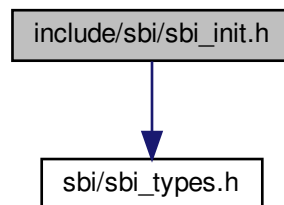
Here is the caller graph for this function:



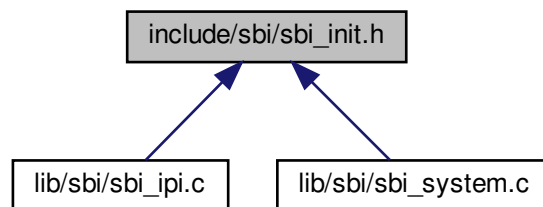
21.37 include/sbi/sbi_init.h File Reference

```
#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)

21.37.1 Function Documentation

21.37.1.1 sbi_exit()

```
void __noreturn sbi_exit (
    struct sbi_scratch * scratch )
```

Exit OpenSBI library for current HART and stop HART

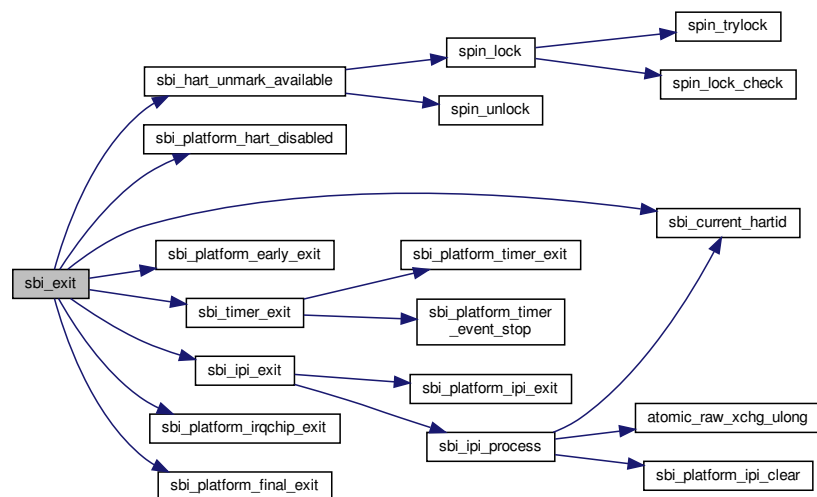
The function expects following:

1. The 'mscratch' CSR is pointing to `sbi_scratch` of current HART
2. Stack pointer (SP) is setup for current HART

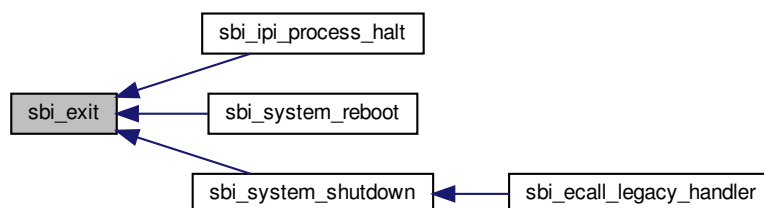
Parameters

<code>scratch</code>	pointer to <code>sbi_scratch</code> of current HART
----------------------	---

Here is the call graph for this function:



Here is the caller graph for this function:



21.37.1.2 sbi_init()

```
void __noreturn sbi_init (
    struct sbi_scratch * scratch )
```

Initialize OpenSBI library for current HART and jump to next booting stage.

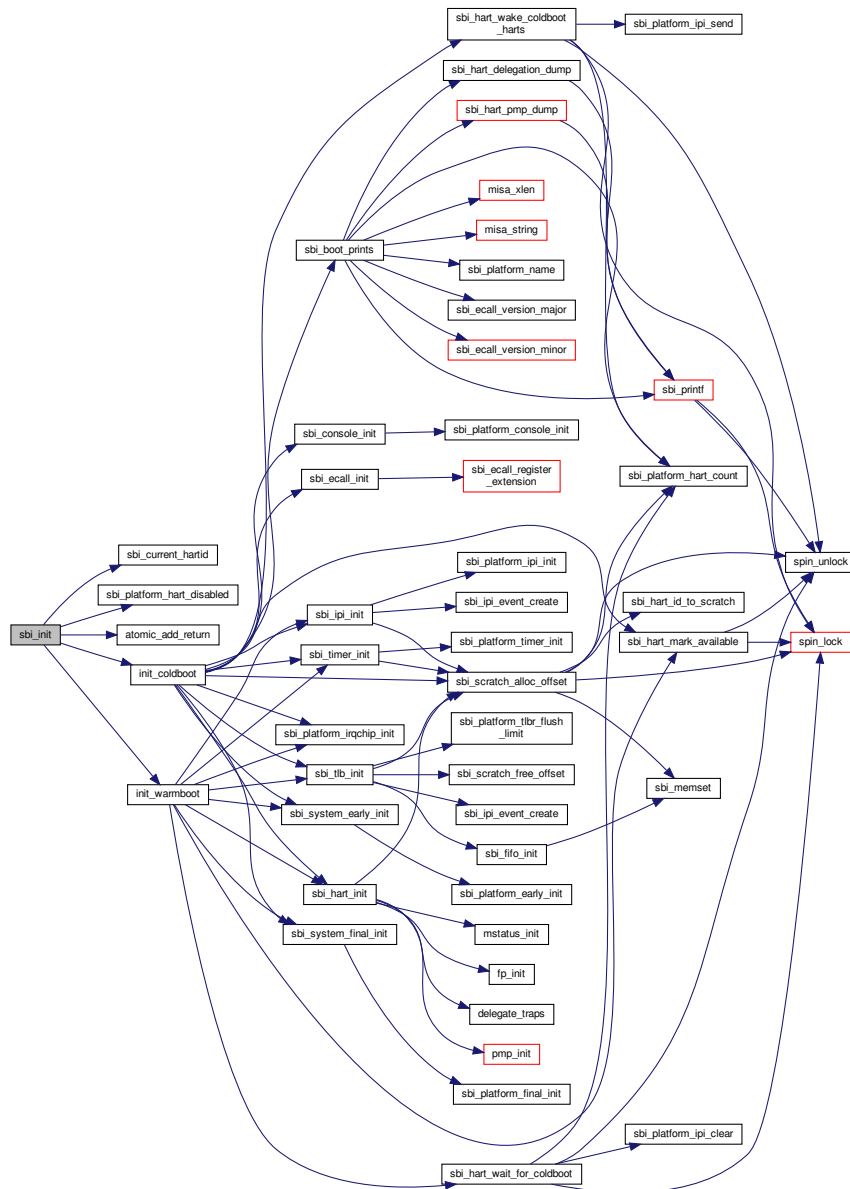
The function expects following:

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

<i>scratch</i>	pointer to sbi_scratch of current HART
----------------	--

Here is the call graph for this function:

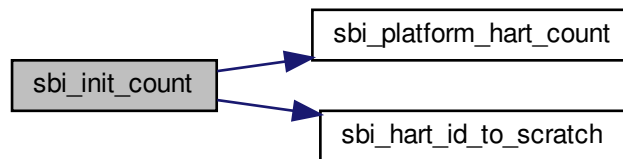


21.37.1.3 sbi_init_count()

```

unsigned long sbi_init_count (
    u32 hartid )
  
```

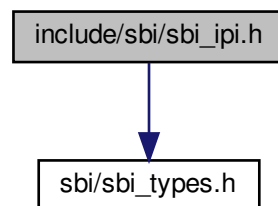
Here is the call graph for this function:



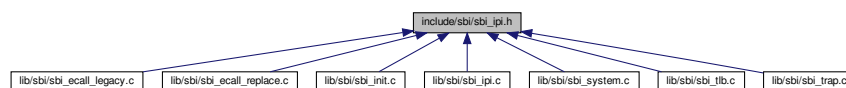
21.38 include/sbi/sbi_ipi.h File Reference

```
#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` (struct `sbi_scratch` *scratch, `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` (struct `sbi_scratch` *scratch, `ulong` hmask, `ulong` hbase)
- void `sbi_ipi_clear_smode` (struct `sbi_scratch` *scratch)
- int `sbi_ipi_send_halt` (struct `sbi_scratch` *scratch, `ulong` hmask, `ulong` hbase)
- void `sbi_ipi_process` (struct `sbi_scratch` *scratch)
- int `sbi_ipi_init` (struct `sbi_scratch` *scratch, `bool` cold_boot)
- void `sbi_ipi_exit` (struct `sbi_scratch` *scratch)

21.38.1 Macro Definition Documentation

21.38.1.1 SBI_IPI_EVENT_MAX

```
#define SBI_IPI_EVENT_MAX __riscv_xlen
```

21.38.2 Function Documentation

21.38.2.1 `sbi_ipi_clear_smode()`

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

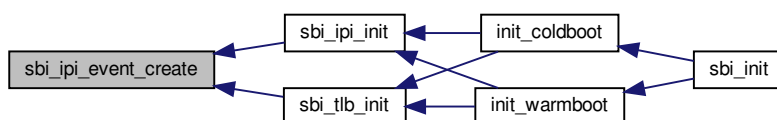
Here is the caller graph for this function:



21.38.2.2 sbi_ipi_event_create()

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

Here is the caller graph for this function:



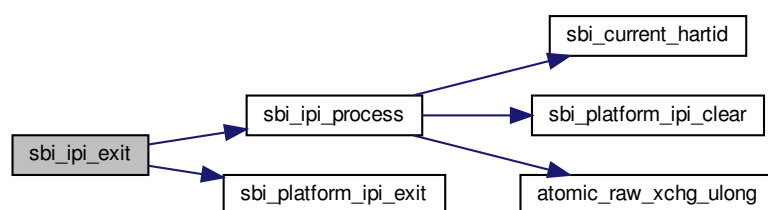
21.38.2.3 sbi_ipi_event_destroy()

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

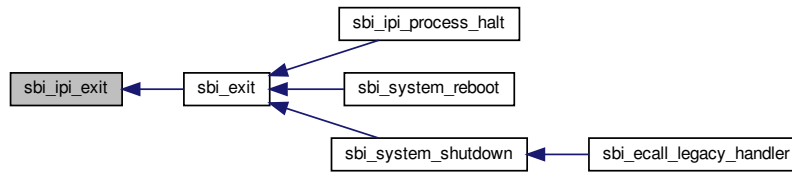
21.38.2.4 sbi_ipi_exit()

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

Here is the call graph for this function:



Here is the caller graph for this function:



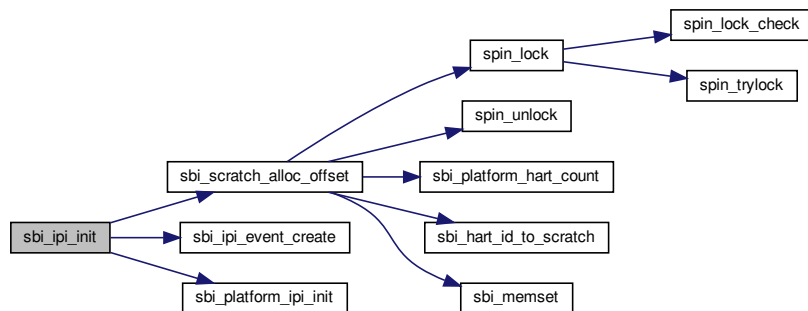
21.38.2.5 sbi_ipi_init()

```

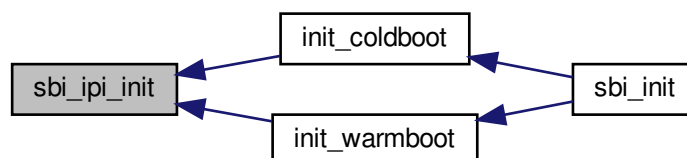
int sbi_ipi_init (
    struct sbi_scratch * scratch,
    bool cold_boot )

```

Here is the call graph for this function:



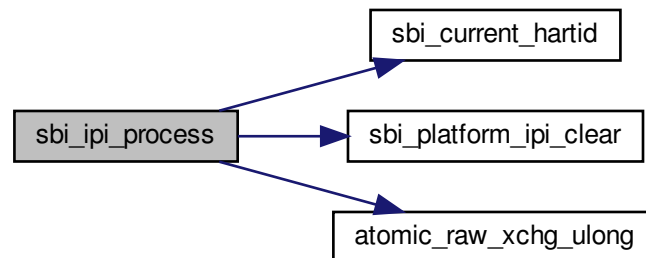
Here is the caller graph for this function:



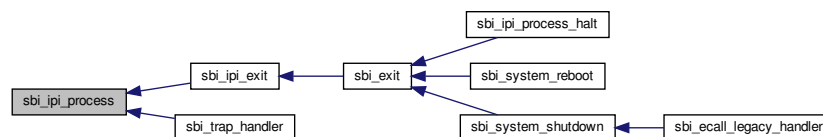
21.38.2.6 sbi_ipi_process()

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

Here is the call graph for this function:



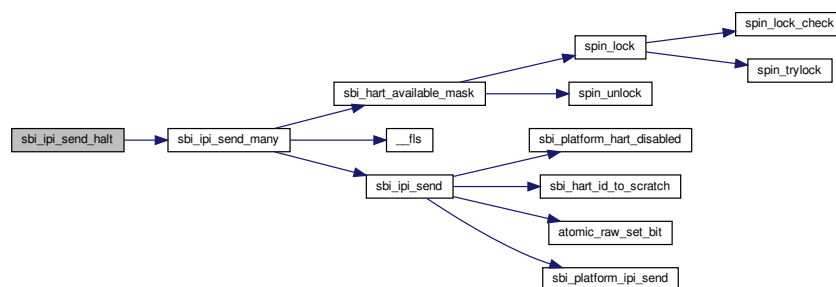
Here is the caller graph for this function:



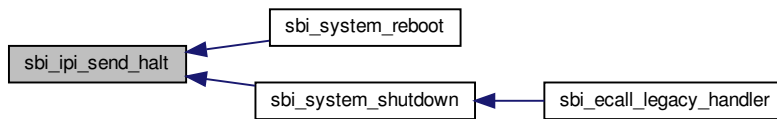
21.38.2.7 sbi_ipi_send_halt()

```
int sbi_ipi_send_halt (
    struct sbi_scratch * scratch,
    ulong hmask,
    ulong hbase )
```

Here is the call graph for this function:



Here is the caller graph for this function:



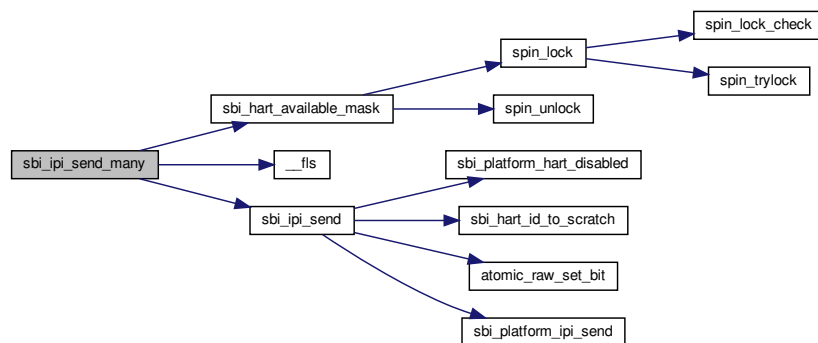
21.38.2.8 sbi_ipi_send_many()

```

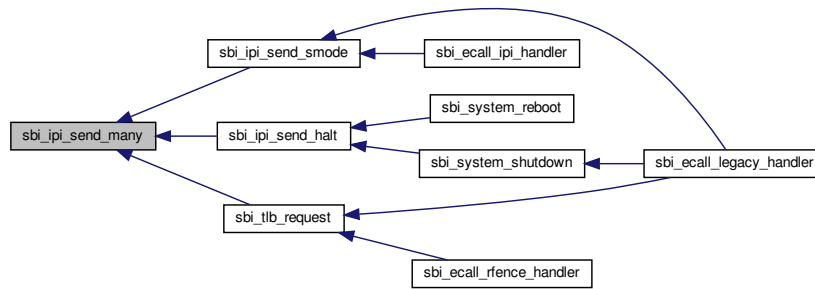
int sbi_ipi_send_many (
    struct sbi_scratch * scratch,
    ulong hmask,
    ulong hbase,
    u32 event,
    void * data )

```

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. FIXME: This check is valid only ULONG size. This is okay for now as available hart mask can support upto ULONG size only. Here is the call graph for this function:



Here is the caller graph for this function:



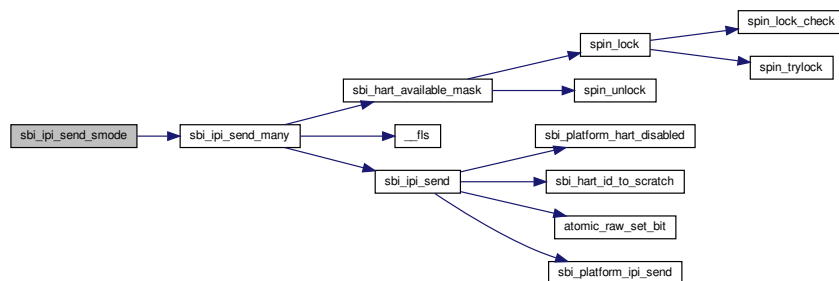
21.38.2.9 sbi_ipi_send_smode()

```

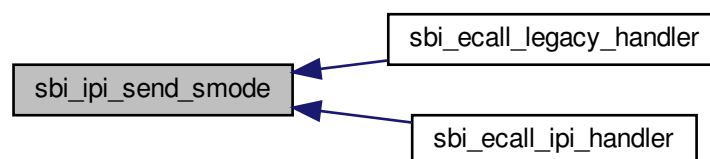
int sbi_ipi_send_smode (
    struct sbi_scratch * scratch,
    ulong hmask,
    ulong hbase )

```

Here is the call graph for this function:

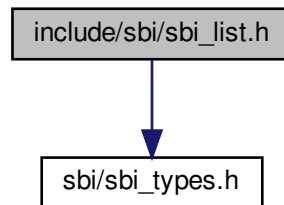


Here is the caller graph for this function:

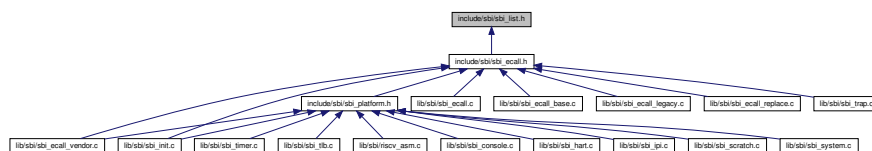


21.39 include/sbi/sbi_list.h File Reference

#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](#) 0xFADEBABA
- #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)

21.39.1 Macro Definition Documentation

21.39.1.1 SBI_INIT_LIST_HEAD

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

Value:

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

21.39.1.2 sbi_list_entry

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

Get the struct for this entry

Parameters

<i>ptr</i>	the &struct list_head pointer.
<i>type</i>	the type of the struct this is embedded in.
<i>member</i>	the name of the list_struct within the struct.

21.39.1.3 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

<i>ptr</i>	the list head to take the element from.
<i>type</i>	the type of the struct this is embedded in.
<i>member</i>	the name of the list_struct within the struct.

Note: that list is expected to be not empty.

21.39.1.4 `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

<i>pos</i>	the &struct list_head to use as a loop cursor.
<i>head</i>	the head for your list.

21.39.1.5 `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

<i>pos</i>	the type * to use as a loop cursor.
<i>head</i>	the head for your list.
<i>member</i>	the name of the list_struct within the struct.

21.39.1.6 `SBI_LIST_HEAD`

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

21.39.1.7 SBI_LIST_HEAD_INIT

```
#define SBI_LIST_HEAD_INIT(  
    __lname ) { &(__lname), &(__lname) }
```

21.39.1.8 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

<i>ptr</i>	the list head to take the element from.
<i>type</i>	the type of the struct this is embedded in.
<i>member</i>	the name of the list_head within the struct.

Note: that list is expected to be not empty.

21.39.1.9 SBI_LIST_POISON_NEXT

```
#define SBI_LIST_POISON_NEXT 0xFADEBABE
```

21.39.1.10 SBI_LIST_POISON_PREV

```
#define SBI_LIST_POISON_PREV 0xDEADBEEF
```

21.39.2 Function Documentation

21.39.2.1 __sbi_list_add()

```
static void __sbi_list_add (  
    struct sbi_dlist * new,  
    struct sbi_dlist * prev,  
    struct sbi_dlist * next ) [inline], [static]
```

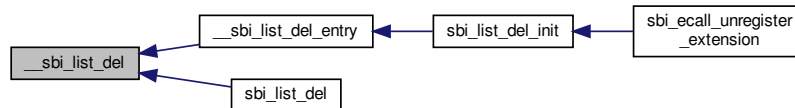
Here is the caller graph for this function:



21.39.2.2 __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:



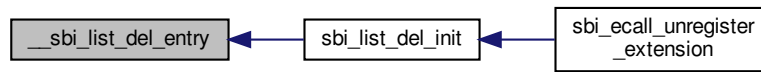
21.39.2.3 __sbi_list_del_entry()

```
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:



21.39.2.4 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

<i>new</i>	New node that needs to be added to list.
<i>head</i>	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:



21.39.2.5 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.

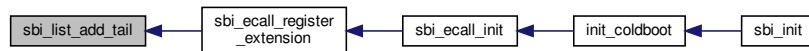
Parameters

<i>new</i>	The new node to be added before tail.
<i>tnode</i>	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:



21.39.2.6 sbi_list_del()

```
static void sbi_list_del (
    struct sbi_dlist * entry ) [inline], [static]
```

Deletes a given entry from list.

Parameters

<i>node</i>	Node to be deleted.
-------------	---------------------

Here is the call graph for this function:



21.39.2.7 sbi_list_del_init()

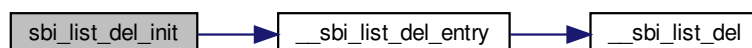
```
static void sbi_list_del_init (
    struct sbi_dlist * entry ) [inline], [static]
```

Deletes entry from list and reinitialize it.

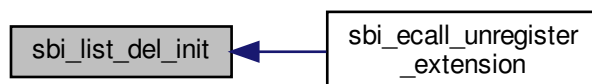
Parameters

<i>entry</i>	the element to delete from the list.
--------------	--------------------------------------

Here is the call graph for this function:



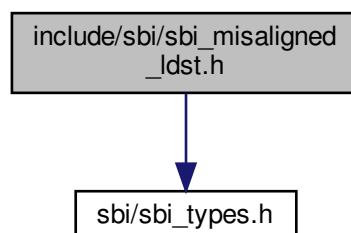
Here is the caller graph for this function:



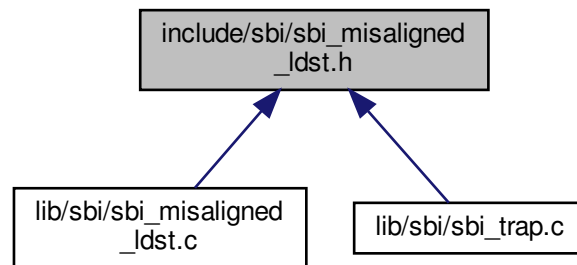
21.40 include/sbi/sbi_misaligned_ldst.h File Reference

```
#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` (u32 hartid, [ulong](#) mcause, [ulong](#) addr, [ulong](#) tval2, [ulong](#) tinst, struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch)
- int `sbi_misaligned_store_handler` (u32 hartid, [ulong](#) mcause, [ulong](#) addr, [ulong](#) tval2, [ulong](#) tinst, struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch)

21.40.1 Function Documentation

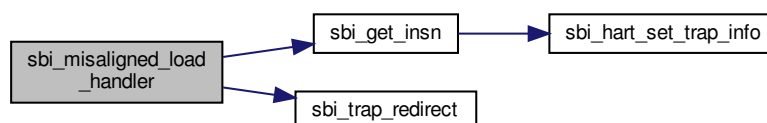
21.40.1.1 sbi_misaligned_load_handler()

```

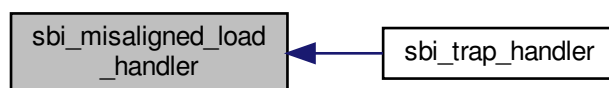
int sbi_misaligned_load_handler (
    u32 hartid,
    ulong mcause,
    ulong addr,
    ulong tval2,
    ulong tinst,
    struct sbi\_trap\_regs * regs,
    struct sbi\_scratch * scratch )

```

Here is the call graph for this function:



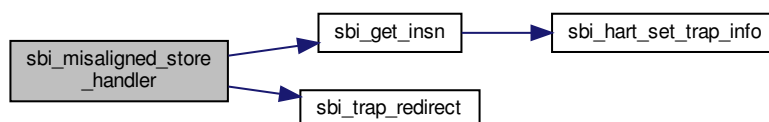
Here is the caller graph for this function:



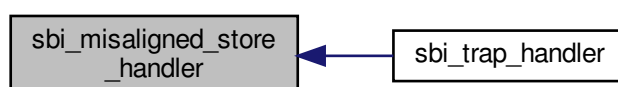
21.40.1.2 `sbi_misaligned_store_handler()`

```
int sbi_misaligned_store_handler (  
    u32 hartid,  
    ulong mcause,  
    ulong addr,  
    ulong tval2,  
    ulong tinst,  
    struct sbi_trap_regs * regs,  
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:

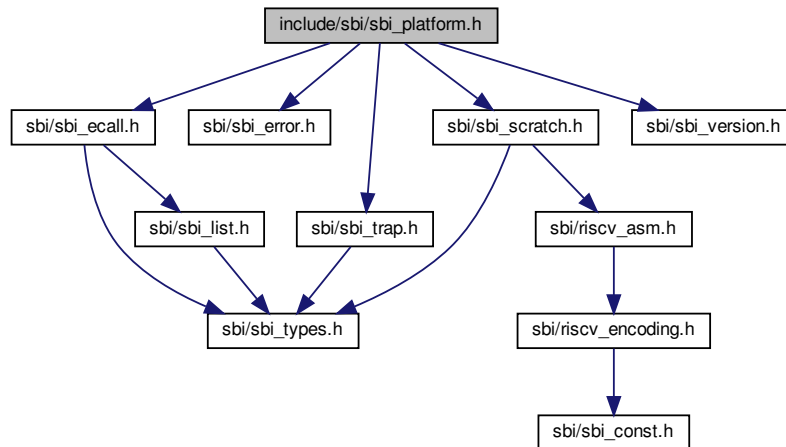


Here is the caller graph for this function:



21.41 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_DISABLED_HART_OFFSET](#) (0x58)
- #define [SBI_PLATFORM_OPS_OFFSET](#) (0x60)

- #define `SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET` (0x60 + __SIZEOF_POINTER__)
- #define `SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT` (1UL << 12)
- #define `SBI_PLATFORM_DEFAULT_FEATURES`
- #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_pmp(__p)` ((__p)->features & `SBI_PLATFORM_HAS_PMP`)
- #define `sbi_platform_has_scounteren(__p)` ((__p)->features & `SBI_PLATFORM_HAS_SCOUNTEREN`)
- #define `sbi_platform_has_mcounteren(__p)` ((__p)->features & `SBI_PLATFORM_HAS_MCOUNTEREN`)
- #define `sbi_platform_has_mfaults_delegation(__p)` ((__p)->features & `SBI_PLATFORM_HAS_MFAULTS_DELEGATION`)

Enumerations

- enum `sbi_platform_features` {
`SBI_PLATFORM_HAS_TIMER_VALUE` = (1 << 0), `SBI_PLATFORM_HAS_HART_HOTPLUG` = (1 << 1),
`SBI_PLATFORM_HAS_PMP` = (1 << 2), `SBI_PLATFORM_HAS_SCOUNTEREN` = (1 << 3),
`SBI_PLATFORM_HAS_MCOUNTEREN` = (1 << 4), `SBI_PLATFORM_HAS_MFAULTS_DELEGATION` =
(1 << 5) }

Functions

- static const char * `sbi_platform_name` (const struct `sbi_platform` *plat)
- static bool `sbi_platform_hart_disabled` (const struct `sbi_platform` *plat, u32 hartid)
- static u64 `sbi_platform_tlb_rflush_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 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_reboot` (const struct `sbi_platform` *plat, u32 type)
- static int `sbi_platform_system_shutdown` (const struct `sbi_platform` *plat, u32 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](#)

21.41.1 Macro Definition Documentation

21.41.1.1 SBI_PLATFORM_DEFAULT_FEATURES

```
#define SBI_PLATFORM_DEFAULT_FEATURES
```

Value:

```
(SBI_PLATFORM_HAS_TIMER_VALUE | SBI_PLATFORM_HAS_PMP |
 \
 SBI_PLATFORM_HAS_SCOUNTEREN | SBI_PLATFORM_HAS_MCOUNTEREN | \
 SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
```

Default feature set for a platform

21.41.1.2 SBI_PLATFORM_DISABLED_HART_OFFSET

```
#define SBI_PLATFORM_DISABLED_HART_OFFSET (0x58)
```

Offset of disabled_hart_mask in struct [sbi_platform](#)

21.41.1.3 SBI_PLATFORM_FEATURES_OFFSET

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

Offset of features in struct [sbi_platform](#)

21.41.1.4 SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET

```
#define SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET (0x60 + __SIZEOF_POINTER__)
```

Offset of firmware_context in struct [sbi_platform](#)

21.41.1.5 SBI_PLATFORM_HART_COUNT_OFFSET

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

Offset of hart_count in struct [sbi_platform](#)

21.41.1.6 SBI_PLATFORM_HART_STACK_SIZE_OFFSET

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

Offset of hart_stack_size in struct [sbi_platform](#)

21.41.1.7 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

21.41.1.8 sbi_platform_has_mcounteren

```
#define sbi_platform_has_mcounteren(  
    __p ) ((__p)->features & SBI_PLATFORM_HAS_MCOUNTEREN)
```

Check whether the platform supports mcounteren CSR

21.41.1.9 sbi_platform_has_mfaults_delegation

```
#define sbi_platform_has_mfaults_delegation(  
    __p ) ((__p)->features & SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
```

Check whether the platform supports fault delegation

21.41.1.10 sbi_platform_has_pmp

```
#define sbi_platform_has_pmp(  
    __p ) ((__p)->features & SBI_PLATFORM_HAS_PMP)
```

Check whether the platform has PMP support

21.41.1.11 sbi_platform_has_scounteren

```
#define sbi_platform_has_scounteren(  
    __p ) ((__p)->features & SBI_PLATFORM_HAS_SCOUNTEREN)
```

Check whether the platform supports scounteren CSR

21.41.1.12 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

21.41.1.13 SBI_PLATFORM_NAME_OFFSET

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

Offset of name in struct [sbi_platform](#)

21.41.1.14 SBI_PLATFORM_OPENSBI_VERSION_OFFSET

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

Offset of opensbi_version in struct [sbi_platform](#)

21.41.1.15 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

21.41.1.16 SBI_PLATFORM_OPS_OFFSET

```
#define SBI_PLATFORM_OPS_OFFSET (0x60)
```

Offset of platform_ops_addr in struct [sbi_platform](#)

21.41.1.17 sbi_platform_ptr

```
#define sbi_platform_ptr(  
    __s ) ((const struct sbi\_platform *) (__s)->platform_addr)
```

Get pointer to [sbi_platform](#) for [sbi_scratch](#) pointer

21.41.1.18 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

21.41.1.19 SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT

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

21.41.1.20 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

21.41.1.21 SBI_PLATFORM_VERSION_OFFSET

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

Offset of platform_version in struct [sbi_platform](#)

21.41.2 Enumeration Type Documentation

21.41.2.1 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_PMP	Platform has PMP support
SBI_PLATFORM_HAS_SCOUNTEREN	Platform has S-mode counter enable
SBI_PLATFORM_HAS_MCOUNTEREN	Platform has M-mode counter enable
SBI_PLATFORM_HAS_MFAULTS_DELEGATION	Platform has fault delegation support

21.41.3 Function Documentation

21.41.3.1 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

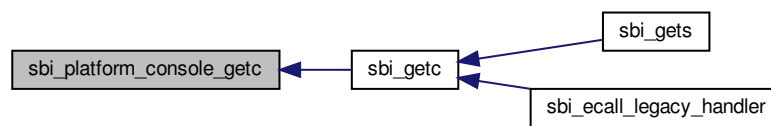
Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Returns

character read from console input

Here is the caller graph for this function:



21.41.3.2 sbi_platform_console_init()

```
static int sbi_platform_console_init (
    const struct sbi\_platform * plat ) [inline], [static]
```

Initialize the platform console

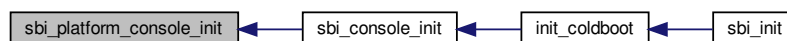
Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



21.41.3.3 sbi_platform_console_putc()

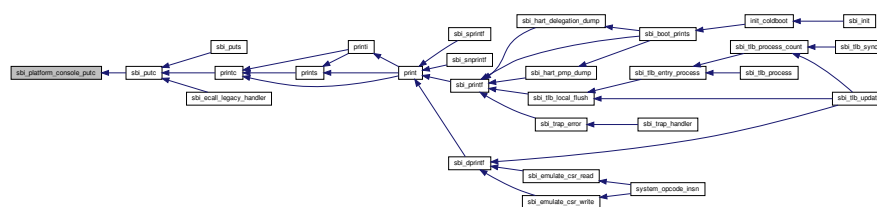
```
static void sbi_platform_console_putc (
    const struct sbi_platform * plat,
    char ch ) [inline], [static]
```

Write a character to the platform console output

Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>ch</i>	character to write

Here is the caller graph for this function:



21.41.3.4 sbi_platform_early_exit()

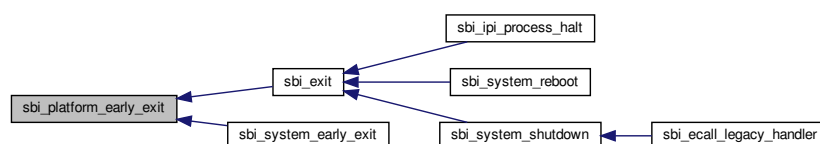
```
static void sbi_platform_early_exit (
    const struct sbi_platform * plat ) [inline], [static]
```

Early exit for current HART

Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Here is the caller graph for this function:



21.41.3.5 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

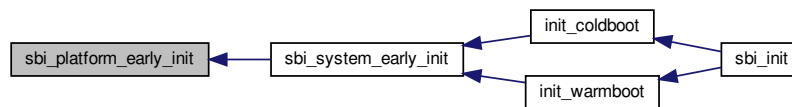
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>cold_boot</i>	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:



21.41.3.6 sbi_platform_final_exit()

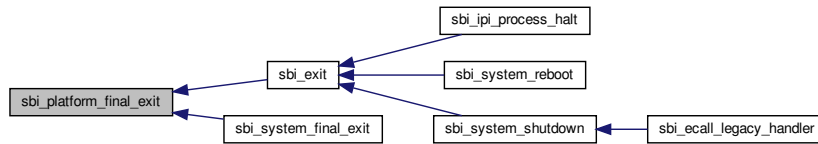
```
static void sbi_platform_final_exit (
    const struct sbi_platform * plat ) [inline], [static]
```

Final exit for current HART

Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Here is the caller graph for this function:



21.41.3.7 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

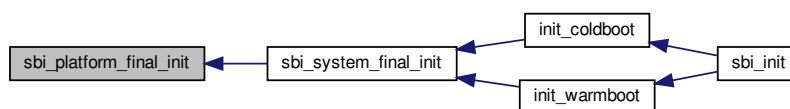
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>cold_boot</i>	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:



21.41.3.8 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

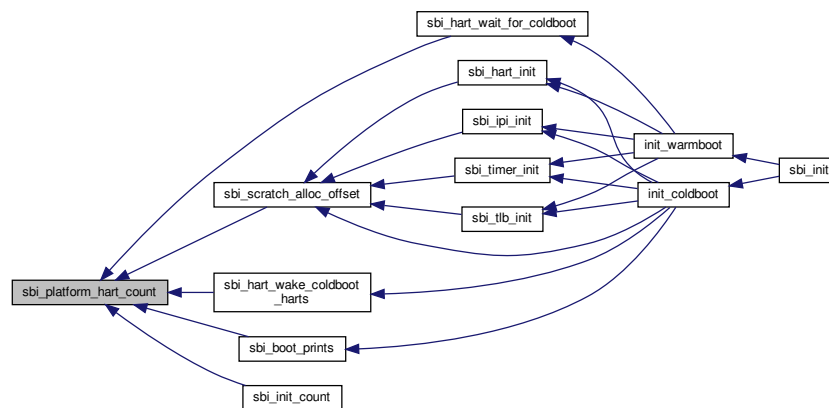
Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Returns

total number of HARTs

Here is the caller graph for this function:



21.41.3.9 sbi_platform_hart_disabled()

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

Check whether the given HART is disabled

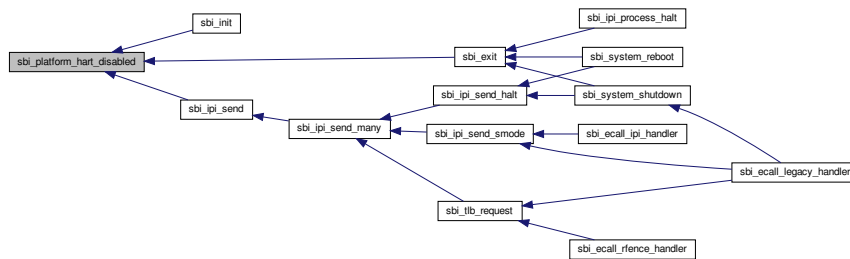
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>hartid</i>	HART ID

Returns

TRUE if HART is disabled and FALSE otherwise

Here is the caller graph for this function:

**21.41.3.10 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

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Returns

stack size in bytes

21.41.3.11 sbi_platform_ipi_clear()

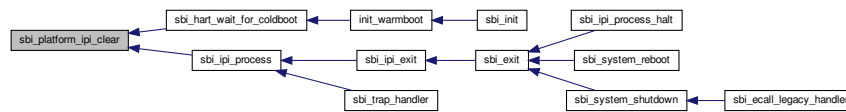
```
static void sbi_platform_ipi_clear (
    const struct sbi_platform * plat,
    u32 target_hart ) [inline], [static]
```

Clear IPI for a target HART

Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>target_hart</i>	HART ID of IPI target

Here is the caller graph for this function:



21.41.3.12 sbi_platform_ipi_exit()

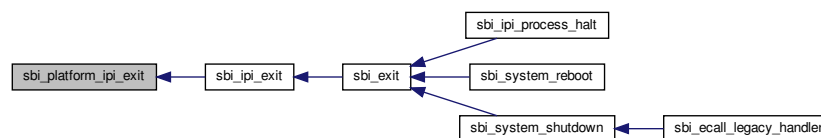
```
static void sbi_platform_ipi_exit (
    const struct sbi\_platform * plat ) [inline], [static]
```

Exit the platform IPI support for current HART

Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Here is the caller graph for this function:



21.41.3.13 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

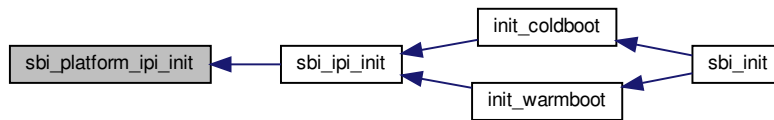
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>cold_boot</i>	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:

**21.41.3.14 sbi_platform_ipi_send()**

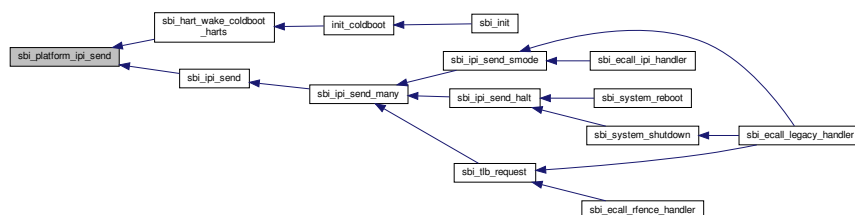
```
static void sbi_platform_ipi_send (
    const struct sbi_platform * plat,
    u32 target_hart ) [inline], [static]
```

Send IPI to a target HART

Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>target_hart</i>	HART ID of IPI target

Here is the caller graph for this function:

**21.41.3.15 sbi_platform_irqchip_exit()**

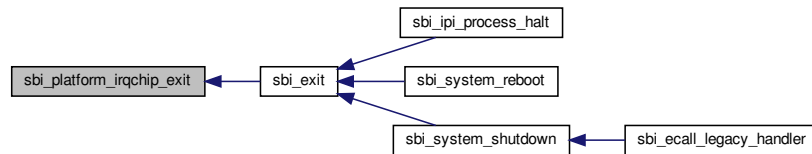
```
static void sbi_platform_irqchip_exit (
    const struct sbi_platform * plat ) [inline], [static]
```

Exit the platform interrupt controller for current HART

Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Here is the caller graph for this function:

21.41.3.16 `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

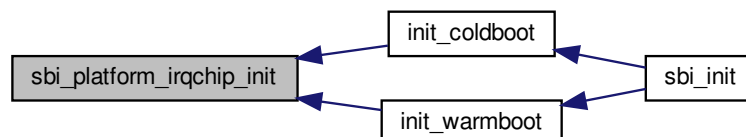
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>cold_boot</i>	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:



21.41.3.17 sbi_platform_misa_extension()

```
static int sbi_platform_misa_extension (
    const struct sbi_platform * plat,
    char ext ) [inline], [static]
```

Check CPU extension in MISA

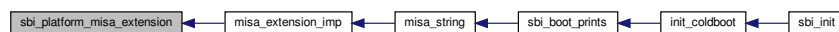
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>ext</i>	shorthand letter for CPU extensions

Returns

zero for not-supported and non-zero for supported

Here is the caller graph for this function:



21.41.3.18 sbi_platform_misa_xlen()

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

Get MXL field of MISA

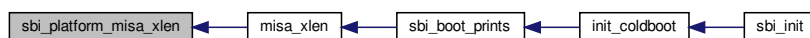
Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Returns

1/2/3 on success and error code on failure

Here is the caller graph for this function:



21.41.3.19 sbi_platform_name()

```
static const char* sbi_platform_name (
    const struct sbi_platform * plat ) [inline], [static]
```

Get name of the platform

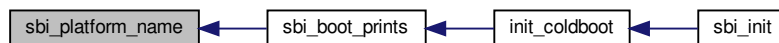
Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Returns

pointer to platform name on success and "Unknown" on failure

Here is the caller graph for this function:



21.41.3.20 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

<i>plat</i>	pointer to struct sbi_platform
<i>hartid</i>	HART ID

Returns

number of PMP regions

Here is the caller graph for this function:

**21.41.3.21 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

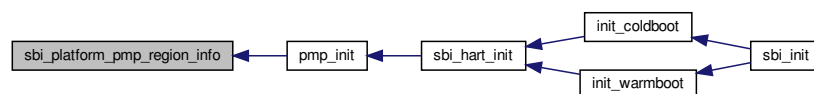
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>hartid</i>	HART ID
<i>index</i>	index of PMP region for which we want details
<i>prot</i>	output pointer for PMP region protection
<i>addr</i>	output pointer for PMP region base address
<i>log2size</i>	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:



21.41.3.22 sbi_platform_system_reboot()

```
static int sbi_platform_system_reboot (
    const struct sbi_platform * plat,
    u32 type ) [inline], [static]
```

Reboot the platform

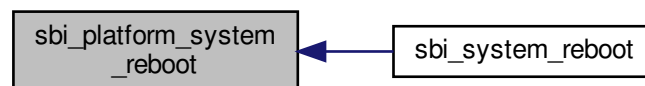
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>type</i>	type of reboot

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



21.41.3.23 sbi_platform_system_shutdown()

```
static int sbi_platform_system_shutdown (
    const struct sbi_platform * plat,
    u32 type ) [inline], [static]
```

Shutdown or poweroff the platform

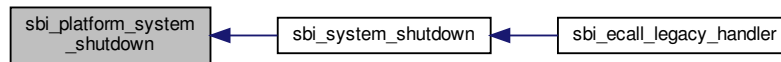
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>type</i>	type of shutdown or poweroff

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



21.41.3.24 sbi_platform_timer_event_start()

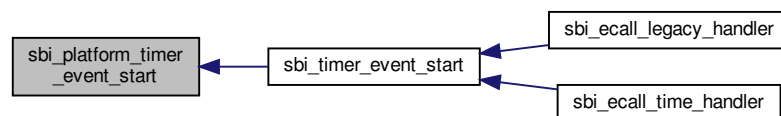
```
static void sbi_platform_timer_event_start (
    const struct sbi_platform * plat,
    u64 next_event ) [inline], [static]
```

Start platform timer event for current HART

Parameters

<i>plat</i>	pointer to struct <code>sbi_platform</code>
<i>next_event</i>	timer value when timer event will happen

Here is the caller graph for this function:



21.41.3.25 sbi_platform_timer_event_stop()

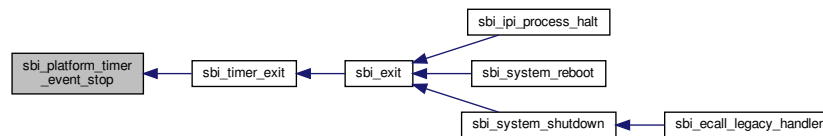
```
static void sbi_platform_timer_event_stop (
    const struct sbi_platform * plat ) [inline], [static]
```

Stop platform timer event for current HART

Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Here is the caller graph for this function:

21.41.3.26 `sbi_platform_timer_exit()`

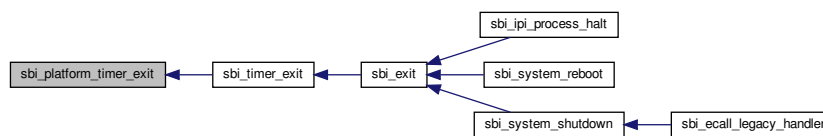
```
static void sbi_platform_timer_exit (
    const struct sbi\_platform * plat ) [inline], [static]
```

Exit the platform timer for current HART

Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Here is the caller graph for this function:

21.41.3.27 `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

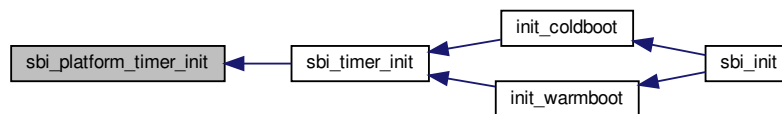
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>cold_boot</i>	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:



21.41.3.28 sbi_platform_timer_value()

```
static u64 sbi_platform_timer_value (
    const struct sbi\_platform * plat ) [inline], [static]
```

Get platform timer value

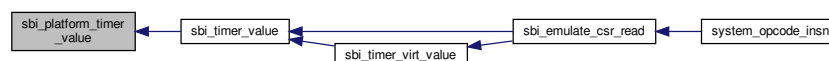
Parameters

<i>plat</i>	pointer to struct sbi_platform
-------------	--

Returns

64-bit timer value

Here is the caller graph for this function:



21.41.3.29 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.

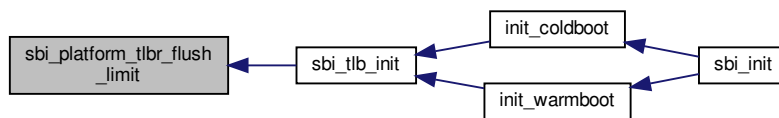
Parameters

<i>plat</i>	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:



21.41.3.30 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.

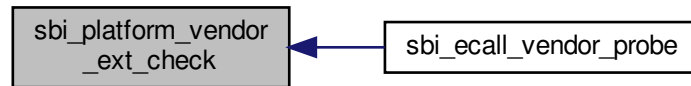
Parameters

<i>plat</i>	pointer to struct sbi_platform
<i>extid</i>	vendor SBI extension id

Returns

0 if extid is not implemented and 1 if implemented

Here is the caller graph for this function:

**21.41.3.31 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.

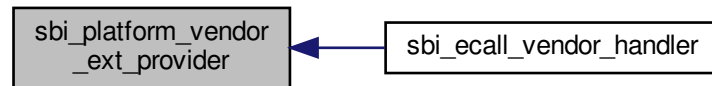
Parameters

<i>plat</i>	pointer to struct <code>sbi_platform</code>
<i>extid</i>	vendor SBI extension id
<i>funcid</i>	SBI function id within the extension id
<i>args</i>	pointer to arguments passed by the caller
<i>out_value</i>	output value that can be filled by the callee
<i>out_trap</i>	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:

**21.41.4 Variable Documentation****21.41.4.1 `__packed`**

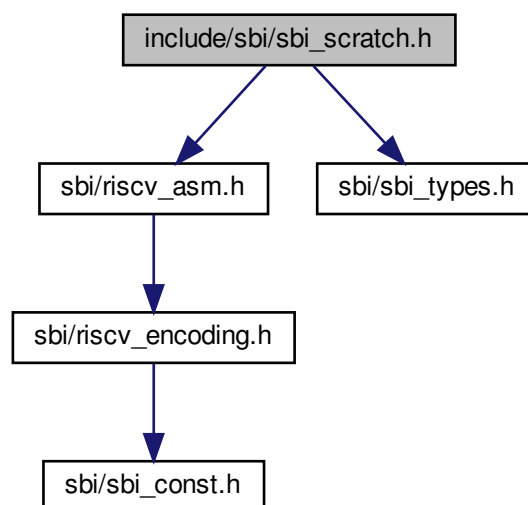
```
struct sbi_platform __packed
```

21.42 `include/sbi/sbi_scratch.h` File Reference

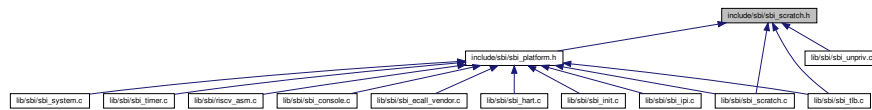
```
#include <sbi/riscv_asm.h>
```

```
#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](#) (64 * __SIZEOF_POINTER__)
- #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))

Enumerations

- enum [sbi_scratch_options](#) { [SBI_SCRATCH_NO_BOOT_PRINTS](#) = (1 << 0), [SBI_SCRATCH_DEBUG_PRINTS](#) = (1 << 1) }

Functions

- 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](#)

21.42.1 Macro Definition Documentation

21.42.1.1 SBI_SCRATCH_EXTRA_SPACE_OFFSET

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

Offset of extra space in [sbi_scratch](#)

21.42.1.2 SBI_SCRATCH_FW_SIZE_OFFSET

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

Offset of fw_size member in [sbi_scratch](#)

21.42.1.3 SBI_SCRATCH_FW_START_OFFSET

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

Offset of fw_start member in [sbi_scratch](#)

21.42.1.4 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](#)

21.42.1.5 SBI_SCRATCH_NEXT_ADDR_OFFSET

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

Offset of next_addr member in [sbi_scratch](#)

21.42.1.6 SBI_SCRATCH_NEXT_ARG1_OFFSET

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

Offset of next_arg1 member in [sbi_scratch](#)

21.42.1.7 SBI_SCRATCH_NEXT_MODE_OFFSET

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

Offset of next_mode member in [sbi_scratch](#)

21.42.1.8 sbi_scratch_offset_ptr

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

Get pointer from offset in [sbi_scratch](#)

21.42.1.9 SBI_SCRATCH_OPTIONS_OFFSET

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

Offset of options member in [sbi_scratch](#)

21.42.1.10 SBI_SCRATCH_PLATFORM_ADDR_OFFSET

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

Offset of platform_addr member in [sbi_scratch](#)

21.42.1.11 SBI_SCRATCH_SIZE

```
#define SBI_SCRATCH_SIZE (64 * __SIZEOF_POINTER__)
```

Maximum size of [sbi_scratch](#)

21.42.1.12 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

21.42.1.13 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

21.42.1.14 sbi_scratch_thishart_ptr

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

Get pointer to [sbi_scratch](#) for current HART

21.42.1.15 SBI_SCRATCH_TMP0_OFFSET

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

Offset of tmp0 member in [sbi_scratch](#)

21.42.1.16 SBI_SCRATCH_WARMBOOT_ADDR_OFFSET

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

Offset of warmboot_addr member in [sbi_scratch](#)

21.42.2 Enumeration Type Documentation

21.42.2.1 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

21.42.3 Function Documentation

21.42.3.1 sbi_scratch_alloc_offset()

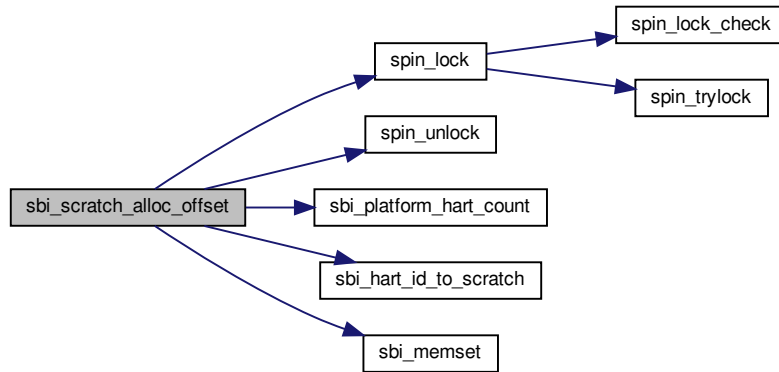
```
unsigned long sbi_scratch_alloc_offset (  
    unsigned long size,  
    const char * owner )
```

Allocate from extra space in [sbi_scratch](#)

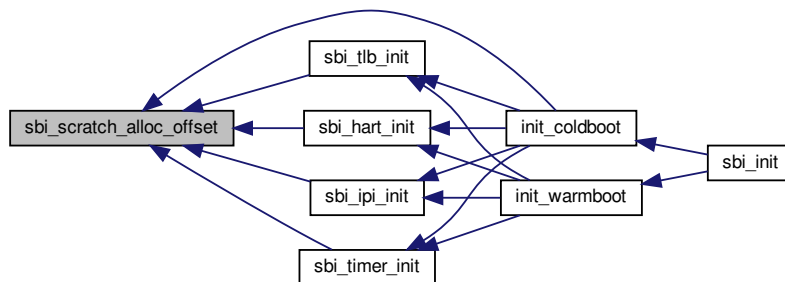
Returns

zero on failure and non-zero (\geq SBI_SCRATCH_EXTRA_SPACE_OFFSET) on success

Here is the call graph for this function:



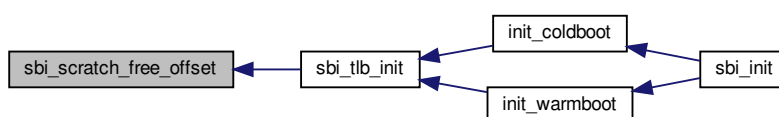
Here is the caller graph for this function:



21.42.3.2 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:



21.43.1 Function Documentation

21.43.1.1 sbi_memchr()

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

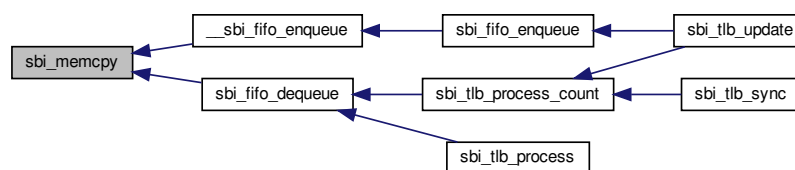
21.43.1.2 sbi_memcmp()

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

21.43.1.3 sbi_memcpy()

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

Here is the caller graph for this function:



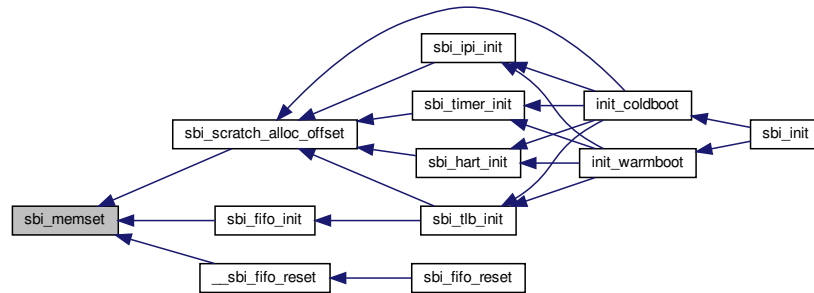
21.43.1.4 sbi_memmove()

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

21.43.1.5 sbi_memset()

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

Here is the caller graph for this function:



21.43.1.6 sbi_strchr()

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

21.43.1.7 sbi_strcmp()

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

21.43.1.8 sbi_strcpy()

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

21.43.1.9 sbi_strlen()

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

Here is the caller graph for this function:



21.43.1.10 sbi_strncpy()

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

21.43.1.11 sbi_strnlen()

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

21.43.1.12 sbi_strchr()

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

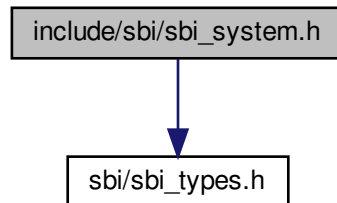
Here is the call graph for this function:



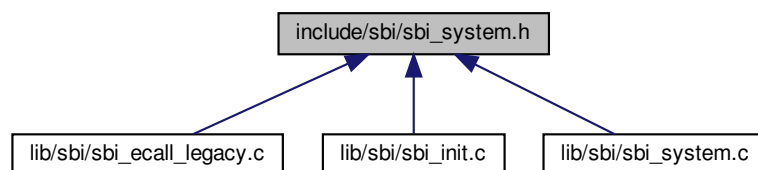
21.44 include/sbi/sbi_system.h File Reference

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

Include dependency graph for sbi_system.h:



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



Functions

- int [sbi_system_early_init](#) (struct [sbi_scratch](#) *scratch, bool cold_boot)
- int [sbi_system_final_init](#) (struct [sbi_scratch](#) *scratch, bool cold_boot)
- void [sbi_system_early_exit](#) (struct [sbi_scratch](#) *scratch)
- void [sbi_system_final_exit](#) (struct [sbi_scratch](#) *scratch)
- void [__noreturn sbi_system_reboot](#) (struct [sbi_scratch](#) *scratch, u32 type)
- void [__noreturn sbi_system_shutdown](#) (struct [sbi_scratch](#) *scratch, u32 type)

21.44.1 Function Documentation

21.44.1.1 sbi_system_early_exit()

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

Here is the call graph for this function:



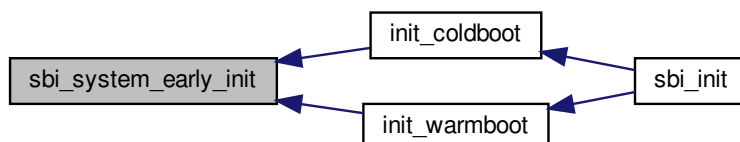
21.44.1.2 sbi_system_early_init()

```
int sbi_system_early_init (
    struct sbi_scratch * scratch,
    bool cold_boot )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.44.1.3 sbi_system_final_exit()

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

Here is the call graph for this function:



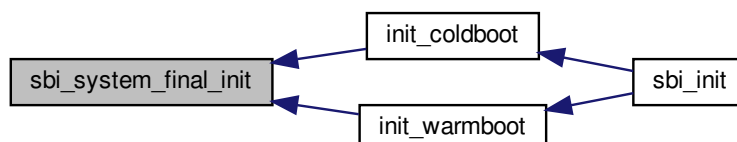
21.44.1.4 sbi_system_final_init()

```
int sbi_system_final_init (
    struct sbi_scratch * scratch,
    bool cold_boot )
```

Here is the call graph for this function:



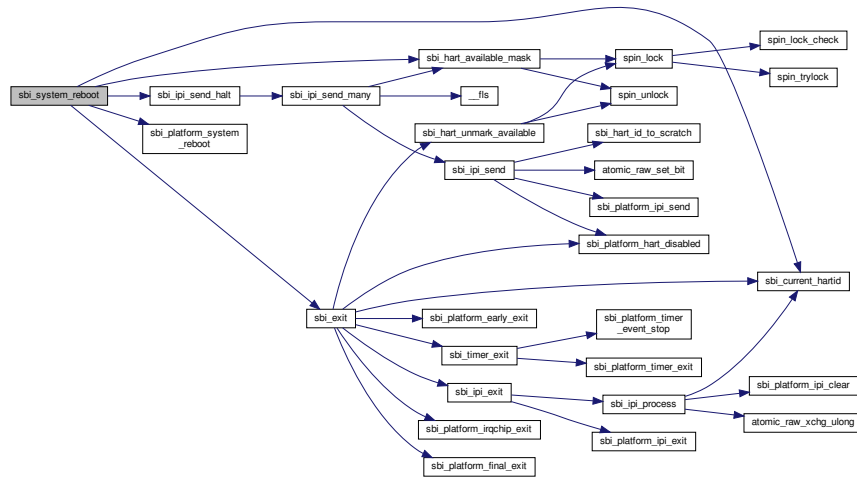
Here is the caller graph for this function:



21.44.1.5 sbi_system_reboot()

```
void __noreturn sbi_system_reboot (
    struct sbi_scratch * scratch,
    u32 type )
```

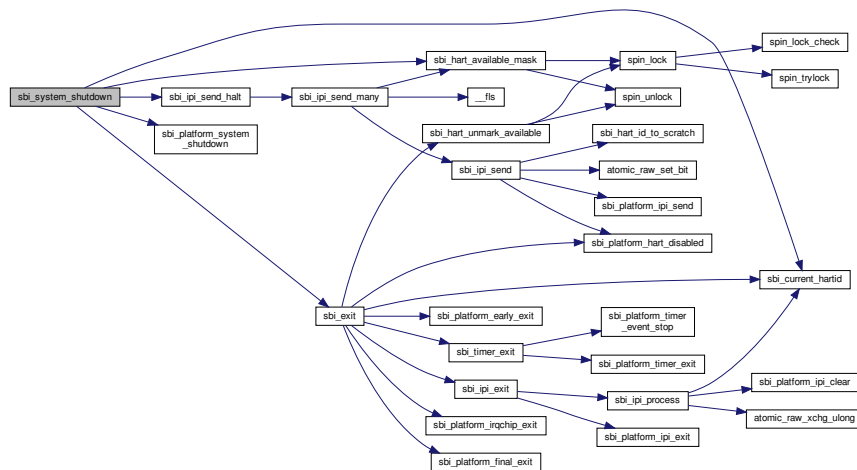
Here is the call graph for this function:



21.44.1.6 sbi_system_shutdown()

```
void __noreturn sbi_system_shutdown (
    struct sbi_scratch * scratch,
    u32 type )
```

Here is the call graph for this function:



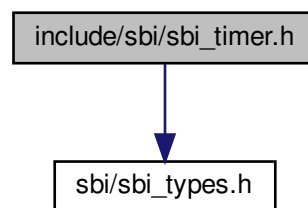
Here is the caller graph for this function:



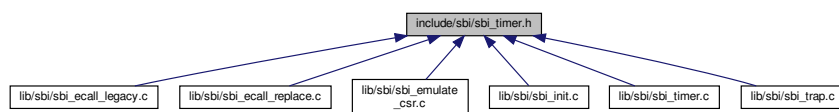
21.45 include/sbi/sbi_timer.h File Reference

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

Include dependency graph for `sbi_timer.h`:



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



Functions

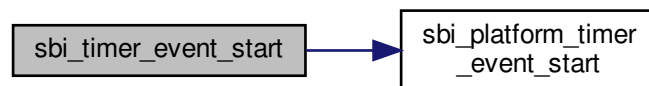
- u64 `sbi_timer_value` (struct `sbi_scratch` *scratch)
- u64 `sbi_timer_virt_value` (struct `sbi_scratch` *scratch)
- u64 `sbi_timer_get_delta` (struct `sbi_scratch` *scratch)
- void `sbi_timer_set_delta` (struct `sbi_scratch` *scratch, `ulong` delta)
- void `sbi_timer_set_delta_upper` (struct `sbi_scratch` *scratch, `ulong` delta_upper)
- void `sbi_timer_event_start` (struct `sbi_scratch` *scratch, u64 next_event)
- void `sbi_timer_process` (struct `sbi_scratch` *scratch)
- int `sbi_timer_init` (struct `sbi_scratch` *scratch, `bool` cold_boot)
- void `sbi_timer_exit` (struct `sbi_scratch` *scratch)

21.45.1 Function Documentation

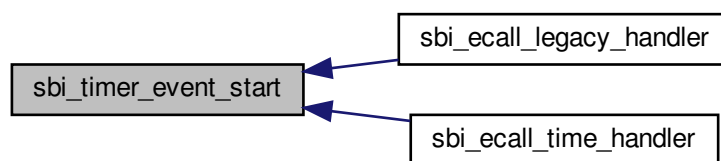
21.45.1.1 sbi_timer_event_start()

```
void sbi_timer_event_start (
    struct sbi_scratch * scratch,
    u64 next_event )
```

Here is the call graph for this function:



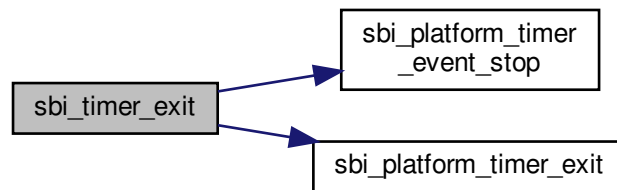
Here is the caller graph for this function:



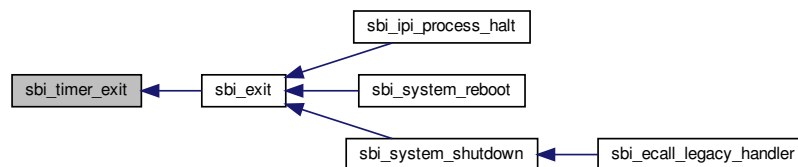
21.45.1.2 sbi_timer_exit()

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

Here is the call graph for this function:



Here is the caller graph for this function:

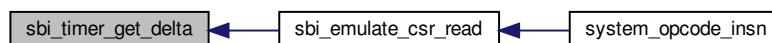


21.45.1.3 sbi_timer_get_delta()

```

u64 sbi_timer_get_delta (
    struct sbi_scratch * scratch )
  
```

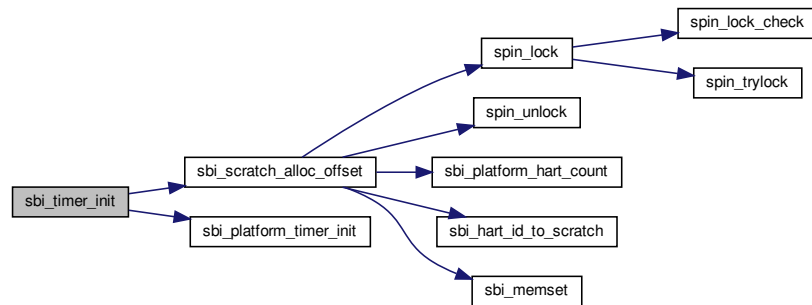
Here is the caller graph for this function:



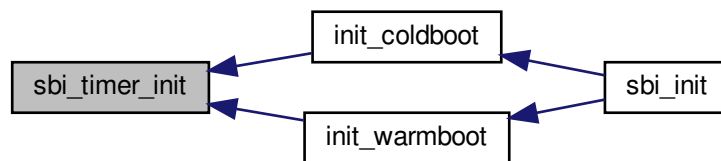
21.45.1.4 sbi_timer_init()

```
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:



21.45.1.5 sbi_timer_process()

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

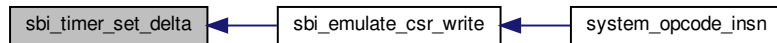
Here is the caller graph for this function:



21.45.1.6 sbi_timer_set_delta()

```
void sbi_timer_set_delta (
    struct sbi_scratch * scratch,
    ulong delta )
```

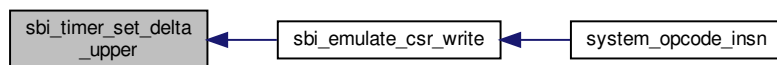
Here is the caller graph for this function:



21.45.1.7 sbi_timer_set_delta_upper()

```
void sbi_timer_set_delta_upper (
    struct sbi_scratch * scratch,
    ulong delta_upper )
```

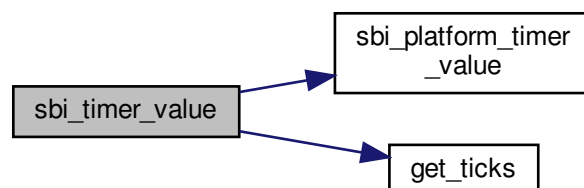
Here is the caller graph for this function:



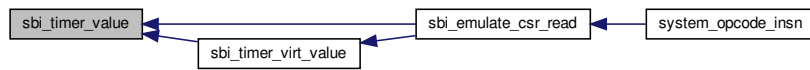
21.45.1.8 sbi_timer_value()

```
u64 sbi_timer_value (
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:



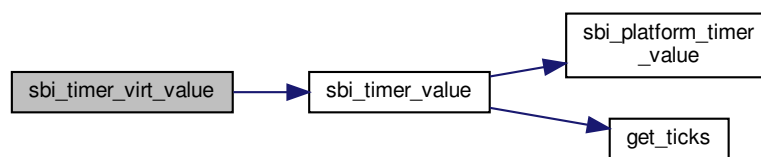
Here is the caller graph for this function:



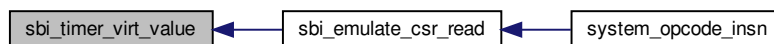
21.45.1.9 sbi_timer_virt_value()

```
u64 sbi_timer_virt_value (
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:



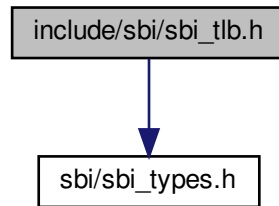
Here is the caller graph for this function:



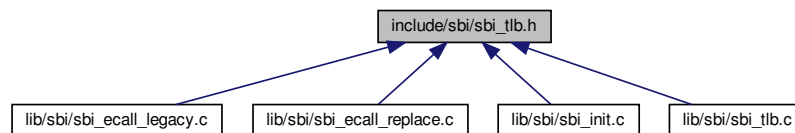
21.46 include/sbi/sbi_tlb.h File Reference

```
#include <sbi/sbi_types.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_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_VPID](#),
[SBI_TLB_FLUSH_VVMA](#), [SBI_TLB_FLUSH_VVMA_ASID](#), [SBI_ITLB_FLUSH](#) }

Functions

- int [sbi_tlb_request](#) (struct [sbi_scratch](#) *scratch, [ulong](#) hmask, [ulong](#) hbase, struct [sbi_tlb_info](#) *tinfo)
- int [sbi_tlb_init](#) (struct [sbi_scratch](#) *scratch, [bool](#) cold_boot)

21.46.1 Macro Definition Documentation

21.46.1.1 SBI_TLB_FIFO_NUM_ENTRIES

```
#define SBI_TLB_FIFO_NUM_ENTRIES 8
```

21.46.1.2 SBI_TLB_FLUSH_ALL

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

21.46.1.3 SBI_TLB_INFO_SIZE

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

21.46.2 Enumeration Type Documentation

21.46.2.1 sbi_tlb_info_types

```
enum sbi_tlb_info_types
```

Enumerator

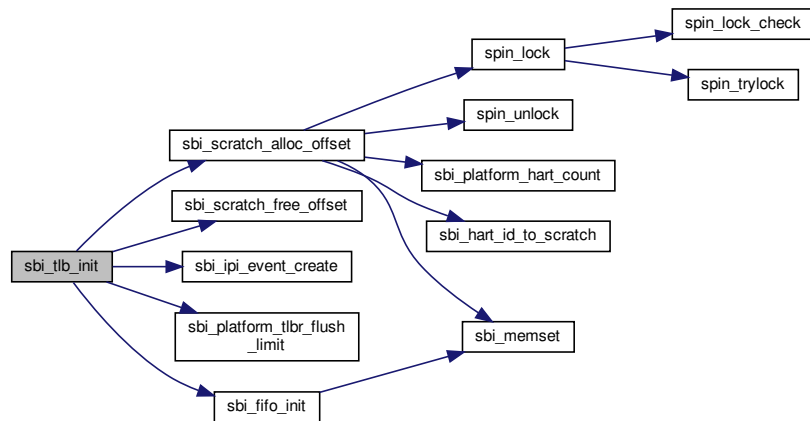
SBI_TLB_FLUSH_VMA	
SBI_TLB_FLUSH_VMA_ASID	
SBI_TLB_FLUSH_GVMA	
SBI_TLB_FLUSH_GVMA_VMID	
SBI_TLB_FLUSH_VVMA	
SBI_TLB_FLUSH_VVMA_ASID	
SBI_ITLB_FLUSH	

21.46.3 Function Documentation

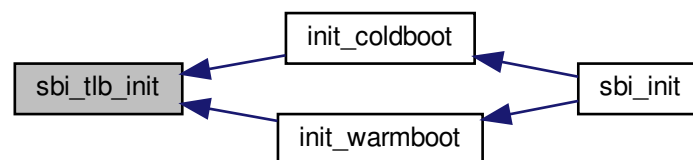
21.46.3.1 sbi_tlb_init()

```
int sbi_tlb_init (
    struct sbi_scratch * scratch,
    bool cold_boot )
```

Here is the call graph for this function:



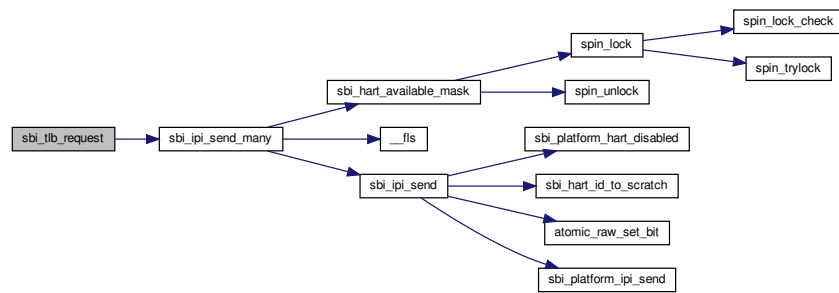
Here is the caller graph for this function:



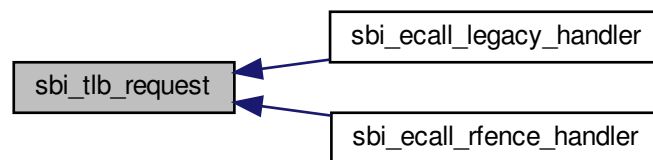
21.46.3.2 sbi_tlb_request()

```
int sbi_tlb_request (
    struct sbi_scratch * scratch,
    ulong hmask,
    ulong hbase,
    struct sbi_tlb_info * tinfo )
```

Here is the call graph for this function:



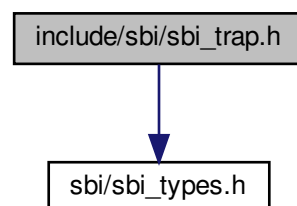
Here is the caller graph for this function:



21.47 include/sbi/sbi_trap.h File Reference

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

Include dependency graph for sbi_trap.h:



Functions

- int [sbi_trap_redirect](#) (struct [sbi_trap_regs](#) *regs, struct [sbi_trap_info](#) *trap, struct [sbi_scratch](#) *scratch)
- void [sbi_trap_handler](#) (struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch)

Variables

- struct [sbi_trap_regs](#) [__packed](#)

21.47.1 Macro Definition Documentation

21.47.1.1 SBI_TRAP_REGS_a0

```
#define SBI_TRAP_REGS_a0 10
```

Index of a0 member in [sbi_trap_regs](#)

21.47.1.2 SBI_TRAP_REGS_a1

```
#define SBI_TRAP_REGS_a1 11
```

Index of a1 member in [sbi_trap_regs](#)

21.47.1.3 SBI_TRAP_REGS_a2

```
#define SBI_TRAP_REGS_a2 12
```

Index of a2 member in [sbi_trap_regs](#)

21.47.1.4 SBI_TRAP_REGS_a3

```
#define SBI_TRAP_REGS_a3 13
```

Index of a3 member in [sbi_trap_regs](#)

21.47.1.5 SBI_TRAP_REGS_a4

```
#define SBI_TRAP_REGS_a4 14
```

Index of a4 member in [sbi_trap_regs](#)

21.47.1.6 SBI_TRAP_REGS_a5

```
#define SBI_TRAP_REGS_a5 15
```

Index of a5 member in [sbi_trap_regs](#)

21.47.1.7 SBI_TRAP_REGS_a6

```
#define SBI_TRAP_REGS_a6 16
```

Index of a6 member in [sbi_trap_regs](#)

21.47.1.8 SBI_TRAP_REGS_a7

```
#define SBI_TRAP_REGS_a7 17
```

Index of a7 member in [sbi_trap_regs](#)

21.47.1.9 SBI_TRAP_REGS_gp

```
#define SBI_TRAP_REGS_gp 3
```

Index of gp member in [sbi_trap_regs](#)

21.47.1.10 SBI_TRAP_REGS_last

```
#define SBI_TRAP_REGS_last 35
```

Last member index in [sbi_trap_regs](#)

21.47.1.11 SBI_TRAP_REGS_mepc

```
#define SBI_TRAP_REGS_mepc 32
```

Index of mepc member in [sbi_trap_regs](#)

21.47.1.12 SBI_TRAP_REGS_mstatus

```
#define SBI_TRAP_REGS_mstatus 33
```

Index of mstatus member in [sbi_trap_regs](#)

21.47.1.13 SBI_TRAP_REGS_mstatusH

```
#define SBI_TRAP_REGS_mstatusH 34
```

Index of mstatusH member in [sbi_trap_regs](#)

21.47.1.14 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](#)

21.47.1.15 SBI_TRAP_REGS_ra

```
#define SBI_TRAP_REGS_ra 1
```

Index of ra member in [sbi_trap_regs](#)

21.47.1.16 SBI_TRAP_REGS_s0

```
#define SBI_TRAP_REGS_s0 8
```

Index of s0 member in [sbi_trap_regs](#)

21.47.1.17 SBI_TRAP_REGS_s1

```
#define SBI_TRAP_REGS_s1 9
```

Index of s1 member in [sbi_trap_regs](#)

21.47.1.18 SBI_TRAP_REGS_s10

```
#define SBI_TRAP_REGS_s10 26
```

Index of s10 member in [sbi_trap_regs](#)

21.47.1.19 SBI_TRAP_REGS_s11

```
#define SBI_TRAP_REGS_s11 27
```

Index of s11 member in [sbi_trap_regs](#)

21.47.1.20 SBI_TRAP_REGS_s2

```
#define SBI_TRAP_REGS_s2 18
```

Index of s2 member in [sbi_trap_regs](#)

21.47.1.21 SBI_TRAP_REGS_s3

```
#define SBI_TRAP_REGS_s3 19
```

Index of s3 member in [sbi_trap_regs](#)

21.47.1.22 SBI_TRAP_REGS_s4

```
#define SBI_TRAP_REGS_s4 20
```

Index of s4 member in [sbi_trap_regs](#)

21.47.1.23 SBI_TRAP_REGS_s5

```
#define SBI_TRAP_REGS_s5 21
```

Index of s5 member in [sbi_trap_regs](#)

21.47.1.24 SBI_TRAP_REGS_s6

```
#define SBI_TRAP_REGS_s6 22
```

Index of s6 member in [sbi_trap_regs](#)

21.47.1.25 SBI_TRAP_REGS_s7

```
#define SBI_TRAP_REGS_s7 23
```

Index of s7 member in [sbi_trap_regs](#)

21.47.1.26 SBI_TRAP_REGS_s8

```
#define SBI_TRAP_REGS_s8 24
```

Index of s8 member in [sbi_trap_regs](#)

21.47.1.27 SBI_TRAP_REGS_s9

```
#define SBI_TRAP_REGS_s9 25
```

Index of s9 member in [sbi_trap_regs](#)

21.47.1.28 SBI_TRAP_REGS_SIZE

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

Size (in bytes) of [sbi_trap_regs](#)

21.47.1.29 SBI_TRAP_REGS_sp

```
#define SBI_TRAP_REGS_sp 2
```

Index of sp member in [sbi_trap_regs](#)

21.47.1.30 SBI_TRAP_REGS_t0

```
#define SBI_TRAP_REGS_t0 5
```

Index of t0 member in [sbi_trap_regs](#)

21.47.1.31 SBI_TRAP_REGS_t1

```
#define SBI_TRAP_REGS_t1 6
```

Index of t1 member in [sbi_trap_regs](#)

21.47.1.32 SBI_TRAP_REGS_t2

```
#define SBI_TRAP_REGS_t2 7
```

Index of t2 member in [sbi_trap_regs](#)

21.47.1.33 SBI_TRAP_REGS_t3

```
#define SBI_TRAP_REGS_t3 28
```

Index of t3 member in [sbi_trap_regs](#)

21.47.1.34 SBI_TRAP_REGS_t4

```
#define SBI_TRAP_REGS_t4 29
```

Index of t4 member in [sbi_trap_regs](#)

21.47.1.35 SBI_TRAP_REGS_t5

```
#define SBI_TRAP_REGS_t5 30
```

Index of t5 member in [sbi_trap_regs](#)

21.47.1.36 SBI_TRAP_REGS_t6

```
#define SBI_TRAP_REGS_t6 31
```

Index of t6 member in [sbi_trap_regs](#)

21.47.1.37 SBI_TRAP_REGS_tp

```
#define SBI_TRAP_REGS_tp 4
```

Index of tp member in [sbi_trap_regs](#)

21.47.1.38 SBI_TRAP_REGS_zero

```
#define SBI_TRAP_REGS_zero 0
```

Index of zero member in [sbi_trap_regs](#)

21.47.2 Function Documentation

21.47.2.1 sbi_trap_handler()

```
void sbi_trap_handler (
    struct sbi\_trap\_regs * regs,
    struct sbi\_scratch * scratch )
```

Handle trap/interrupt

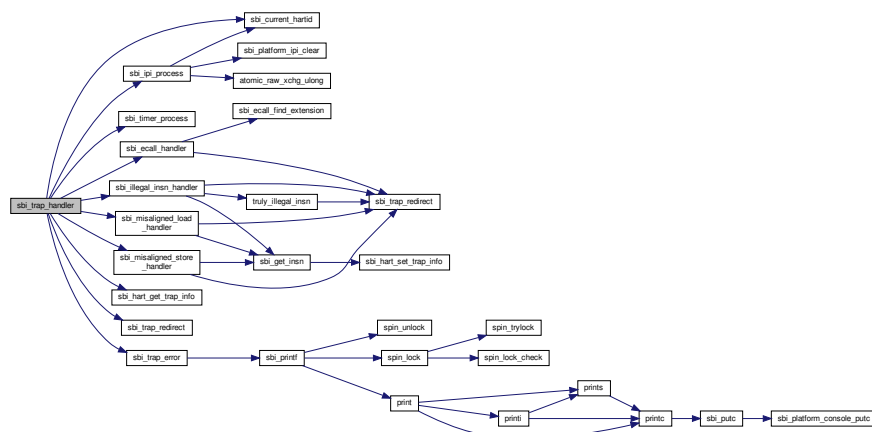
This function is called by firmware linked to OpenSBI library for handling trap/interrupt. It expects the following:

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

<i>regs</i>	pointer to register state
<i>scratch</i>	pointer to sbi_scratch of current HART

Here is the call graph for this function:



21.47.2.2 `sbi_trap_redirect()`

```
int sbi_trap_redirect (
    struct sbi_trap_regs * regs,
    struct sbi_trap_info * trap,
    struct sbi_scratch * scratch )
```

Redirect trap to lower privledge mode (S-mode or U-mode)

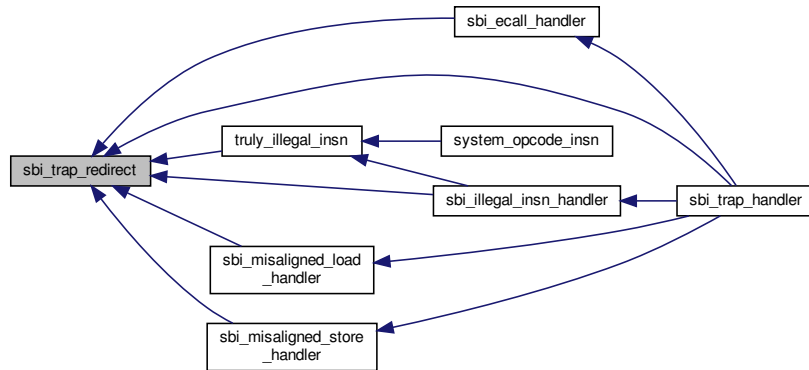
Parameters

<i>regs</i>	pointer to register state
<i>trap</i>	pointer to trap details
<i>scratch</i>	pointer to sbi_scratch of current HART

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

**21.47.3 Variable Documentation****21.47.3.1 __packed**

```
struct sbi_trap_regs __packed
```

21.48 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 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 `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`

21.48.1 Macro Definition Documentation

21.48.1.1 `__noreturn`

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

21.48.1.2 `__packed`

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

21.48.1.3 CLAMP

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

21.48.1.4 container_of

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

Value:

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

21.48.1.5 FALSE

```
#define FALSE 0
```

21.48.1.6 likely

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

21.48.1.7 MAX

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

21.48.1.8 MIN

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


21.48.1.9 NULL

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

21.48.1.10 offsetof

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

21.48.1.11 ROUNDDOWN

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

21.48.1.12 ROUNDUP

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

21.48.1.13 STR

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

21.48.1.14 TRUE

```
#define TRUE 1
```

21.48.1.15 unlikely

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

21.48.1.16 XSTR

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

21.48.2 Typedef Documentation

21.48.2.1 bool

```
typedef int bool
```

21.48.2.2 int16_t

```
typedef short int16_t
```

21.48.2.3 int32_t

```
typedef int int32_t
```

21.48.2.4 physical_addr_t

```
typedef unsigned long physical_addr_t
```

21.48.2.5 physical_size_t

```
typedef unsigned long physical_size_t
```

21.48.2.6 s16

```
typedef short s16
```

21.48.2.7 s32

```
typedef int s32
```

21.48.2.8 s8

```
typedef char s8
```

21.48.2.9 size_t

```
typedef unsigned long size_t
```

21.48.2.10 ssize_t

```
typedef long ssize_t
```

21.48.2.11 u16

```
typedef unsigned short u16
```

21.48.2.12 u32

```
typedef unsigned int u32
```

21.48.2.13 u8

```
typedef unsigned char u8
```

21.48.2.14 uint16_t

```
typedef unsigned short uint16_t
```

21.48.2.15 uint32_t

```
typedef unsigned int uint32_t
```

21.48.2.16 uint8_t

```
typedef unsigned char uint8_t
```

21.48.2.17 uintptr_t

```
typedef unsigned long uintptr_t
```

21.48.2.18 ulong

```
typedef unsigned long ulong
```

21.48.2.19 virtual_addr_t

```
typedef unsigned long virtual_addr_t
```

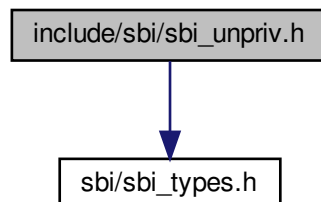
21.48.2.20 virtual_size_t

```
typedef unsigned long virtual_size_t
```

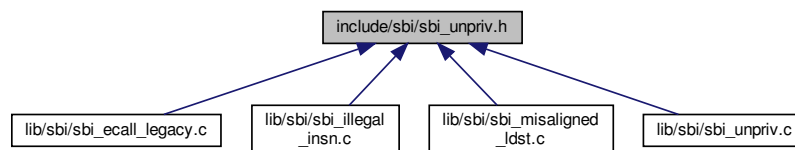
21.49 include/sbi/sbi_unpriv.h File Reference

```
#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_scratch *scratch, struct sbi_trap_info *trap)`

21.49.1 Macro Definition Documentation

21.49.1.1 DECLARE_UNPRIVILEGED_LOAD_FUNCTION

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

Value:

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

21.49.1.2 DECLARE_UNPRIVILEGED_STORE_FUNCTION

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

Value:

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

21.49.2 Function Documentation

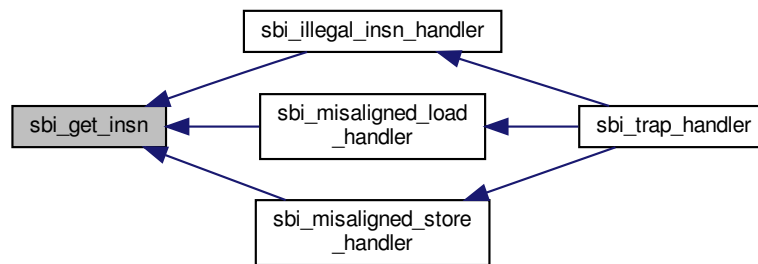
21.49.2.1 sbi_get_insn()

```
ulong sbi_get_insn (  
    ulong mepc,  
    struct sbi_scratch * scratch,  
    struct sbi_trap_info * trap )
```

Here is the call graph for this function:

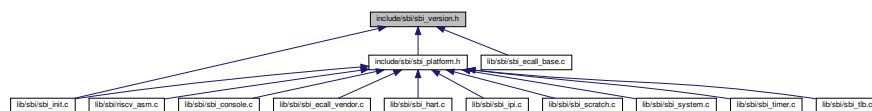


Here is the caller graph for this function:



21.50 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 6`
- `#define OPENSBI_VERSION`

21.50.1 Macro Definition Documentation

21.50.1.1 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

21.50.1.2 OPENSBI_VERSION_MAJOR

```
#define OPENSBI_VERSION_MAJOR 0
```

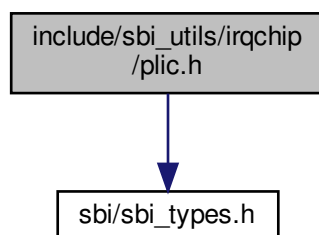
21.50.1.3 OPENSBI_VERSION_MINOR

```
#define OPENSBI_VERSION_MINOR 6
```

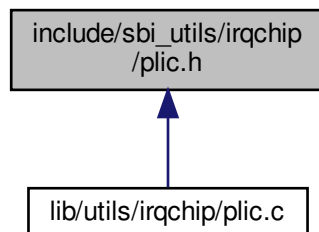
21.51 include/sbi_utils/irqchip/plic.h File Reference

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

Include dependency graph for plic.h:



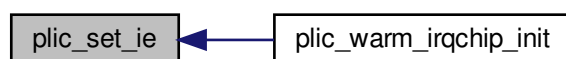
This graph shows which files directly or indirectly include this file:



21.51.1.3 plic_set_ie()

```
void plic_set_ie (
    u32 cntxid,
    u32 word_index,
    u32 val )
```

Here is the caller graph for this function:



21.51.1.4 plic_set_thresh()

```
void plic_set_thresh (
    u32 cntxid,
    u32 val )
```

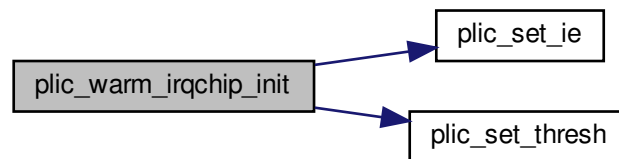
Here is the caller graph for this function:



21.51.1.5 plic_warm_irqchip_init()

```
int plic_warm_irqchip_init (
    u32 target_hart,
    int m_cntx_id,
    int s_cntx_id )
```

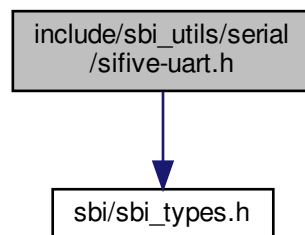
Here is the call graph for this function:



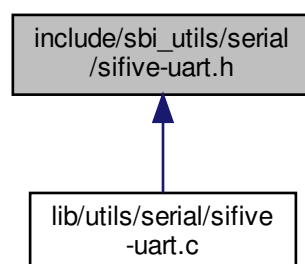
21.52 include/sbi_utils/serial/sifive-uart.h File Reference

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

Include dependency graph for sifive-uart.h:



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



Functions

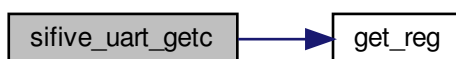
- void [sifive_uart_putc](#) (char ch)
- int [sifive_uart_getc](#) (void)
- int [sifive_uart_init](#) (unsigned long base, [u32](#) in_freq, [u32](#) baudrate)

21.52.1 Function Documentation

21.52.1.1 [sifive_uart_getc](#)()

```
int sifive_uart_getc (  
    void )
```

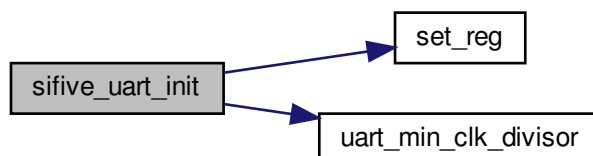
Here is the call graph for this function:



21.52.1.2 [sifive_uart_init](#)()

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

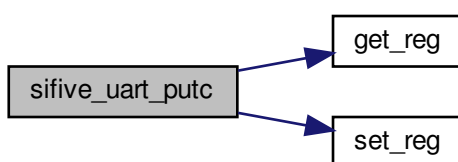
Here is the call graph for this function:



21.52.1.3 sifive_uart_putc()

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

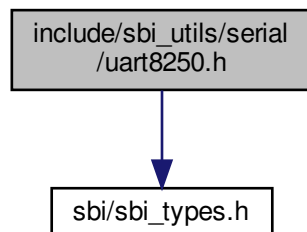
Here is the call graph for this function:



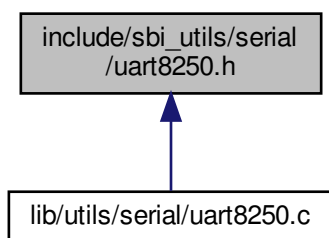
21.53 include/sbi_utils/serial/uart8250.h File Reference

```
#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)

21.53.1 Function Documentation

21.53.1.1 [uart8250_getc\(\)](#)

```
int uart8250\_getc (  
    void )
```

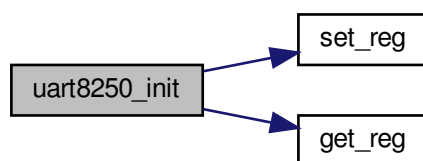
Here is the call graph for this function:



21.53.1.2 uart8250_init()

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

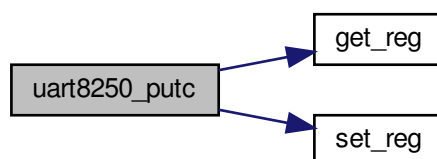
Here is the call graph for this function:



21.53.1.3 uart8250_putc()

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

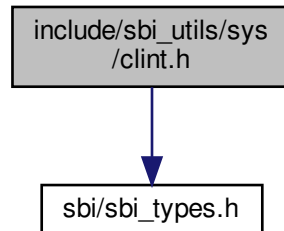
Here is the call graph for this function:



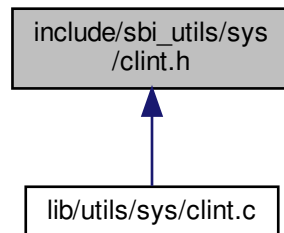
21.54 include/sbi_utils/sys/clint.h File Reference

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

Include dependency graph for clint.h:



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



Functions

- void [clint_ipi_send](#) (u32 target_hart)
- void [clint_ipi_sync](#) (u32 target_hart)
- void [clint_ipi_clear](#) (u32 target_hart)
- int [clint_warm_ipi_init](#) (void)
- int [clint_cold_ipi_init](#) (unsigned long base, u32 hart_count)
- 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](#) (unsigned long base, u32 hart_count, bool has_64bit_mmio)

21.54.1 Function Documentation

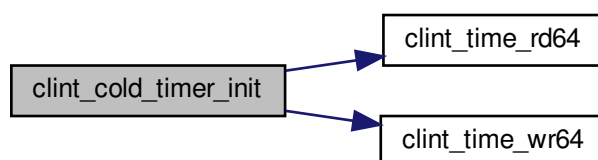
21.54.1.1 clint_cold_ipi_init()

```
int clint_cold_ipi_init (
    unsigned long base,
    u32 hart_count )
```

21.54.1.2 clint_cold_timer_init()

```
int clint_cold_timer_init (
    unsigned long base,
    u32 hart_count,
    bool has_64bit_mmio )
```

Here is the call graph for this function:



21.54.1.3 clint_ipi_clear()

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

Here is the caller graph for this function:



21.54.1.4 clint_ipi_send()

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

21.54.1.5 clint_ipi_sync()

```
void clint_ipi_sync (
    u32 target_hart )
```

21.54.1.6 clint_timer_event_start()

```
void clint_timer_event_start (
    u64 next_event )
```

Here is the call graph for this function:



21.54.1.7 clint_timer_event_stop()

```
void clint_timer_event_stop (
    void )
```

Here is the call graph for this function:



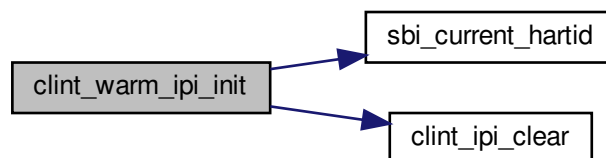
21.54.1.8 clint_timer_value()

```
u64 clint_timer_value (  
    void )
```

21.54.1.9 clint_warm_ipi_init()

```
int clint_warm_ipi_init (  
    void )
```

Here is the call graph for this function:



21.54.1.10 clint_warm_timer_init()

```
int clint_warm_timer_init (  
    void )
```

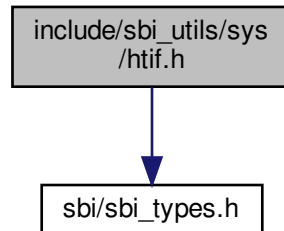
Here is the call graph for this function:



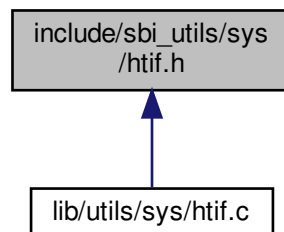
21.55 include/sbi_utils/sys/htif.h File Reference

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

Include dependency graph for htif.h:



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



Functions

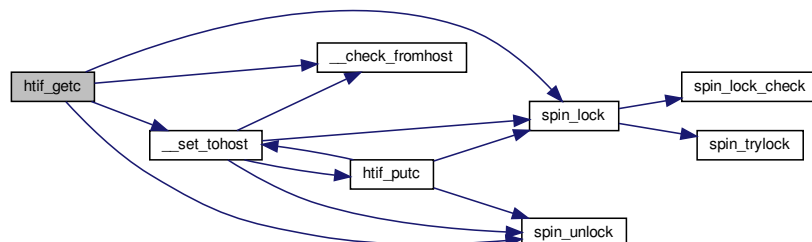
- void [htif_putc](#) (char ch)
- int [htif_getc](#) (void)
- int [htif_system_down](#) (u32 type)

21.55.1 Function Documentation

21.55.1.1 htif_getc()

```
int htif_getc (
    void )
```

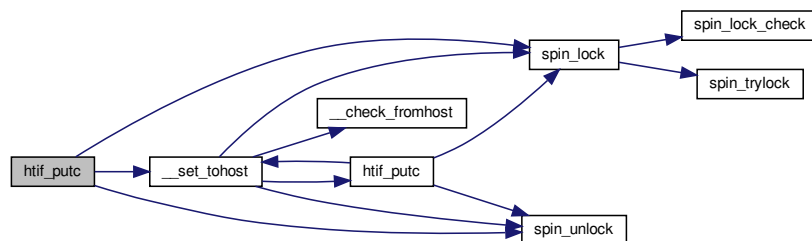
Here is the call graph for this function:



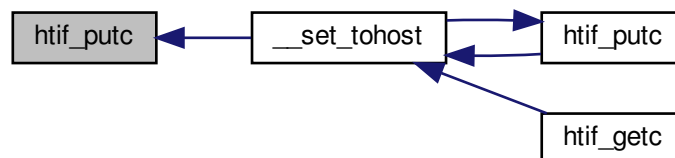
21.55.1.2 htif_putc()

```
void htif_putc (
    char ch )
```

Here is the call graph for this function:



Here is the caller graph for this function:

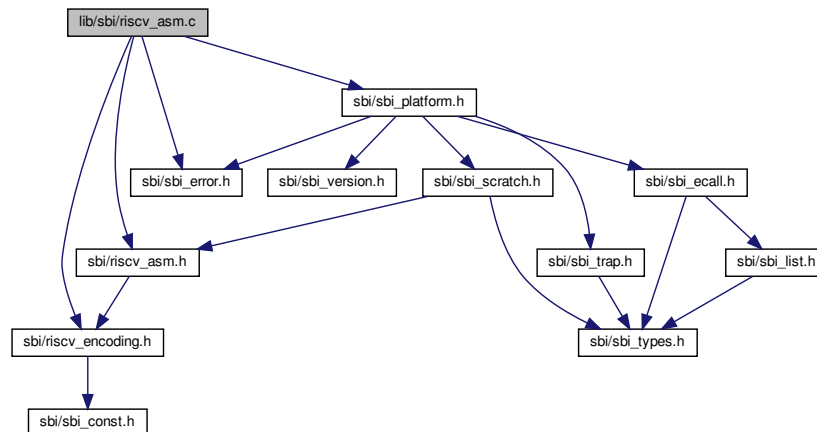


21.55.1.3 htif_system_down()

```
int htif_system_down (
    u32 type )
```

21.56 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

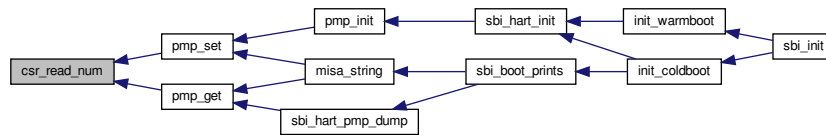
- int [misa_extension_imp](#) (char ext)
- int [misa_xlen](#) (void)
- unsigned long [csr_read_num](#) (int csr_num)
- void [csr_write_num](#) (int csr_num, unsigned long val)
- static unsigned long [ctz](#) (unsigned long x)
- int [pmp_set](#) (unsigned int n, unsigned long prot, unsigned long addr, unsigned long log2len)
- int [pmp_get](#) (unsigned int n, unsigned long *prot_out, unsigned long *addr_out, unsigned long *log2len_out)

21.56.1 Function Documentation

21.56.1.1 csr_read_num()

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

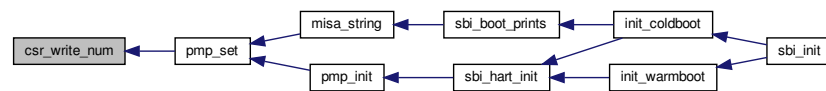
Here is the caller graph for this function:



21.56.1.2 csr_write_num()

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

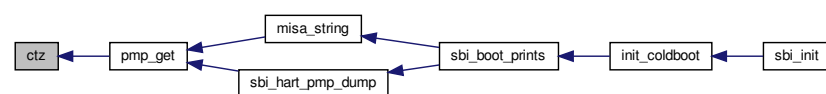
Here is the caller graph for this function:



21.56.1.3 ctz()

```
static unsigned long ctz (
    unsigned long x ) [static]
```

Here is the caller graph for this function:



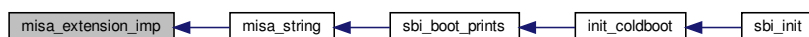
21.56.1.4 misa_extension_imp()

```
int misa_extension_imp (  
    char ext )
```

Here is the call graph for this function:



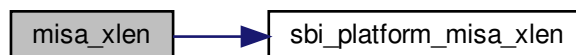
Here is the caller graph for this function:



21.56.1.5 misa_xlen()

```
int misa_xlen (  
    void )
```

Here is the call graph for this function:



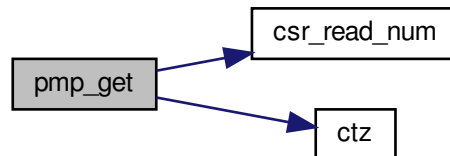
Here is the caller graph for this function:



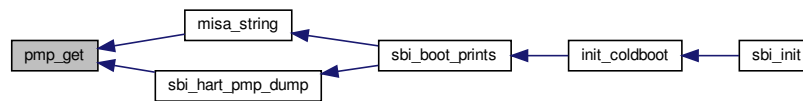
21.56.1.6 pmp_get()

```
int pmp_get (
    unsigned int n,
    unsigned long * prot_out,
    unsigned long * addr_out,
    unsigned long * log2len_out )
```

Here is the call graph for this function:



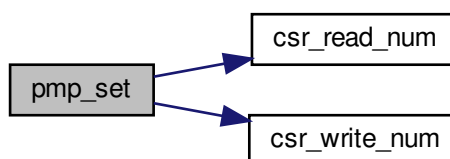
Here is the caller graph for this function:



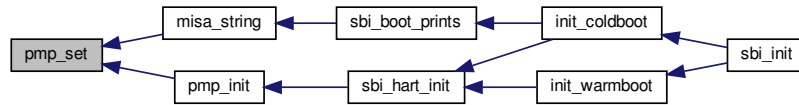
21.56.1.7 pmp_set()

```
int pmp_set (
    unsigned int n,
    unsigned long prot,
    unsigned long addr,
    unsigned long log2len )
```

Here is the call graph for this function:



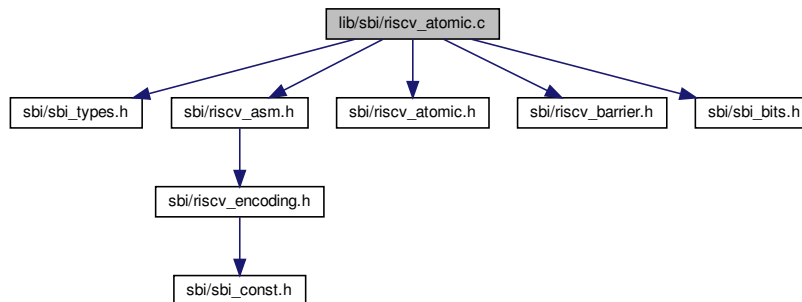
Here is the caller graph for this function:



21.57 lib/sbi/riscv_atomic.c File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_barrier.h>
#include <sbi/sbi_bits.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, .aql)`
- `#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 [arch_atomic_cmpxchg](#) ([atomic_t](#) *atom, long oldval, long newval)
- long [arch_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)

21.57.1 Macro Definition Documentation

21.57.1.1 [__atomic_op_bit](#)

```
#define __atomic_op_bit(  
    op,  
    mod,  
    nr,  
    addr ) \_\_atomic\_op\_bit\_ord(op, mod, nr, addr, .aqr1)
```

21.57.1.2 [__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)]) \br/>        : "r"(mod(__mask)) \br/>        : "memory");  
    __res;  
}))
```

21.57.1.3 __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;
})
```

21.57.1.4 __cmpxchg

```
#define __cmpxchg(
    ptr,
    old,
    new,
    size )
```

21.57.1.5 __NOP

```
#define __NOP(
    x ) (x)
```

21.57.1.6 __NOT

```
#define __NOT(
    x ) (~(x))
```

21.57.1.7 __xchg

```
#define __xchg(
    ptr,
    new,
    size )
```

21.57.1.8 axchg

```
#define axchg(
    ptr,
    x )
```

Value:

```
((
    __typeof__(*(ptr)) _x_ = (x);
    (__typeof__(*(ptr))) __axchg((ptr), _x_, sizeof(*(ptr)));
))
```

21.57.1.9 cmpxchg

```
#define cmpxchg(
    ptr,
    o,
    n )
```

Value:

```
((
    __typeof__(*(ptr)) _o_ = (o);
    __typeof__(*(ptr)) _n_ = (n);
    (__typeof__(*(ptr)))
    __cmpxchg((ptr), _o_, _n_, sizeof(*(ptr)));
))
```

21.57.1.10 xchg

```
#define xchg(
    ptr,
    n )
```

Value:

```
((
    __typeof__(*(ptr)) _n_ = (n);
    (__typeof__(*(ptr))) __xchg((ptr), _n_, sizeof(*(ptr)));
))
```

21.57.2 Function Documentation

21.57.2.1 arch_atomic_cmpxchg()

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

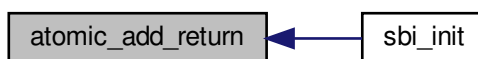
21.57.2.2 arch_atomic_xchg()

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

21.57.2.3 atomic_add_return()

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

Here is the caller graph for this function:



21.57.2.4 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. : Bit to set. : atomic variable to modify Here is the call graph for this function:



21.57.2.5 atomic_raw_clear_bit()

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

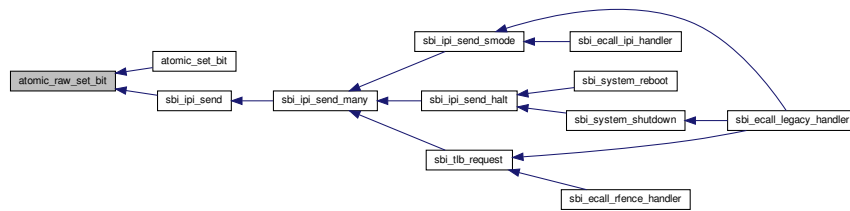
Clear a bit in any address and return the new value . : Bit to set. : Address to modify Here is the caller graph for this function:



21.57.2.6 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 . : Bit to set. : Address to modify Here is the caller graph for this function:



21.57.2.7 atomic_raw_xchg_uint()

```

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

```

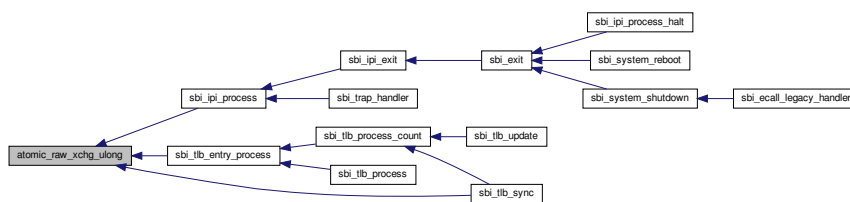
21.57.2.8 atomic_raw_xchg_ulong()

```

unsigned long atomic_raw_xchg_ulong (
    volatile unsigned long * ptr,
    unsigned long newval )

```

Here is the caller graph for this function:



21.57.2.9 atomic_read()

```

long atomic_read (
    atomic_t * atom )

```


21.57.2.10 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. : Bit to set. : atomic variable to modify Here is the call graph for this function:



21.57.2.11 atomic_sub_return()

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

21.57.2.12 atomic_write()

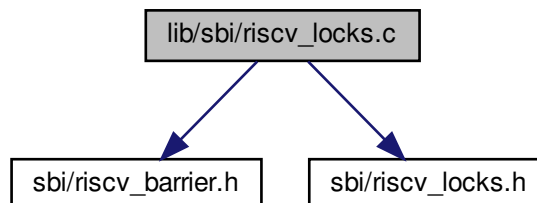
```
void atomic_write (
    atomic_t * atom,
    long value )
```

21.58 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

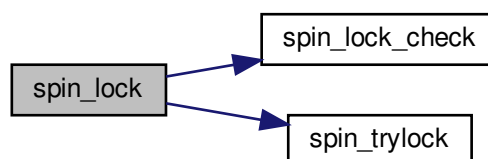
- int [spin_lock_check](#) ([spinlock_t](#) *lock)
- int [spin_trylock](#) ([spinlock_t](#) *lock)
- void [spin_lock](#) ([spinlock_t](#) *lock)
- void [spin_unlock](#) ([spinlock_t](#) *lock)

21.58.1 Function Documentation

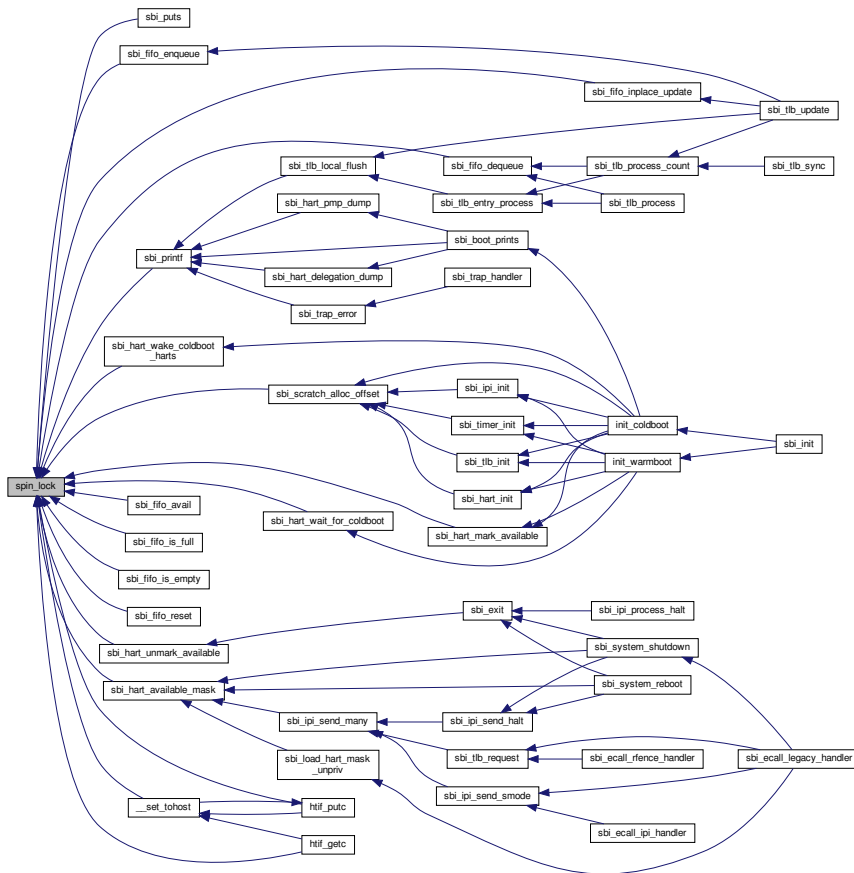
21.58.1.1 [spin_lock\(\)](#)

```
void spin_lock (  
    spinlock\_t * lock )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.58.1.2 spin_lock_check()

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

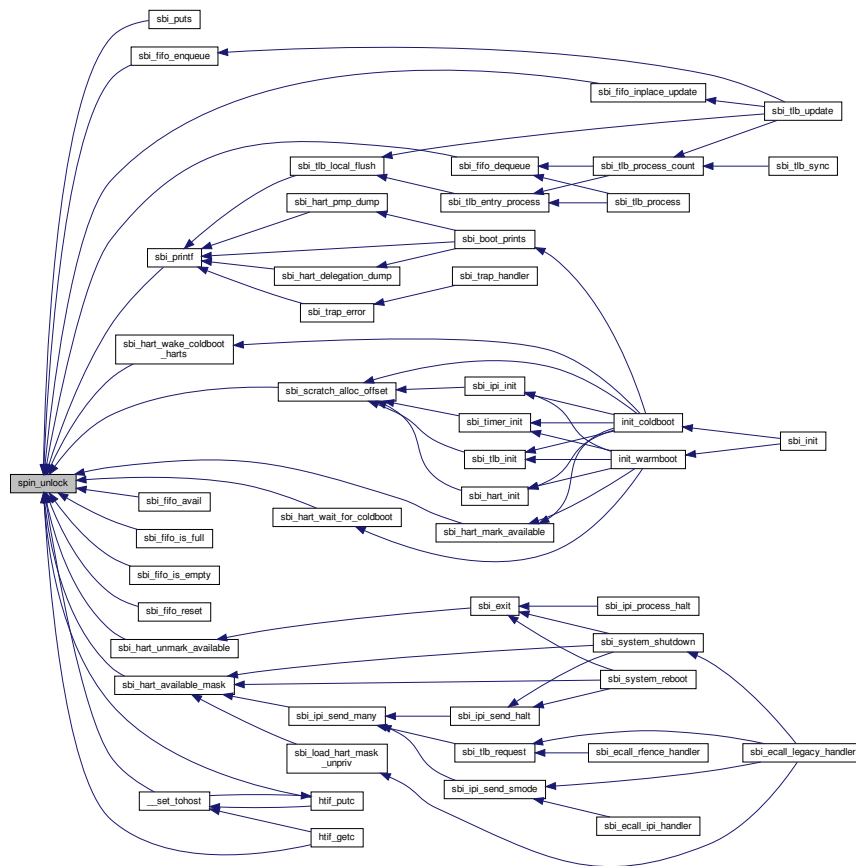

The diagram illustrates a dense network of dependencies between various system components, primarily related to the SBI (System Binary Interface) and kernel operations. The nodes are represented as rectangular boxes, and the connections are directed arrows.

- Central Hub:** `spin_lock` and `spin_trylock` are central nodes with numerous incoming and outgoing dependencies.
- SBI Functions:** A large group of nodes at the top and right, including `sbi_puts`, `sbi_fifo_enqueue`, `sbi_fifo_inplace_update`, `sbi_fifo_update`, `sbi_fifo_local_flush`, `sbi_fifo_dequeue`, `sbi_fifo_process_count`, `sbi_fifo_sync`, `sbi_hart_pmp_dump`, `sbi_tlb_entry_process`, `sbi_tlb_process`, `sbi_print`, `sbi_boot_prints`, `sbi_hart_delegation_dump`, `sbi_trap_handler`, `sbi_trap_error`, `sbi_hart_wake_coldboot_harts`, `sbi_scratch_alloc_offset`, `sbi_init`, `init_coldboot`, `init_warmboot`, `sbi_timer_init`, `sbi_tlb_init`, `sbi_hart_init`, `sbi_hart_wait_for_coldboot`, `sbi_hart_mark_available`, `sbi_exit`, `sbi_ipi_process_halt`, `sbi_system_shutdown`, `sbi_system_reboot`, `sbi_ipi_send_many`, `sbi_ipi_send_halt`, `sbi_load_hart_mask_urpriv`, `sbi_request`, `sbi_ipi_send_smode`, `sbi_ecall_rfence_handler`, `sbi_ecall_legacy_handler`, `sbi_ecall_ipi_handler`.
- Kernel Components:** Nodes like `__set_tohost`, `helf_putc`, and `helf_getc` are located at the bottom.
- Other Functions:** `sbi_fifo_is_full`, `sbi_fifo_is_empty`, `sbi_fifo_reset`, `sbi_hart_unmark_available`, `sbi_hart_available_mask`, `sbi_hart_mark_available`, `sbi_ipi_send_halt`, `sbi_ipi_send_smode`, `sbi_ecall_rfence_handler`, `sbi_ecall_legacy_handler`, `sbi_ecall_ipi_handler`.

The graph shows a highly interconnected system where many functions depend on a small set of core components, particularly those related to locking and SBI communication.

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

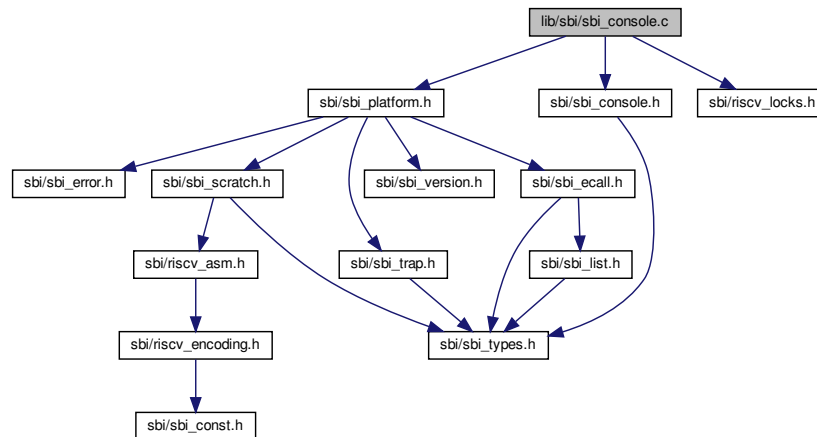
Here is the caller graph for this function:



21.59 lib/sbi/sbi_console.c File Reference

```
#include <sbi/sbi_platform.h>
#include <sbi/sbi_console.h>
#include <sbi/riscv_locks.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 (struct sbi_scratch *scratch, 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

21.59.1 Macro Definition Documentation

21.59.1.1 PAD_ALTERNATE

```
#define PAD_ALTERNATE 4
```

21.59.1.2 PAD_RIGHT

```
#define PAD_RIGHT 1
```

21.59.1.3 PAD_ZERO

```
#define PAD_ZERO 2
```

21.59.1.4 PRINT_BUF_LEN

```
#define PRINT_BUF_LEN 64
```

21.59.1.5 va_arg

```
#define va_arg __builtin_va_arg
```

21.59.1.6 va_end

```
#define va_end __builtin_va_end
```


21.59.1.7 va_start

```
#define va_start(
    v,
    l ) __builtin_va_start((v), l)
```

21.59.2 Typedef Documentation

21.59.2.1 va_list

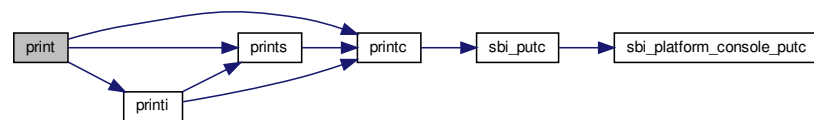
```
typedef __builtin_va_list va_list
```

21.59.3 Function Documentation

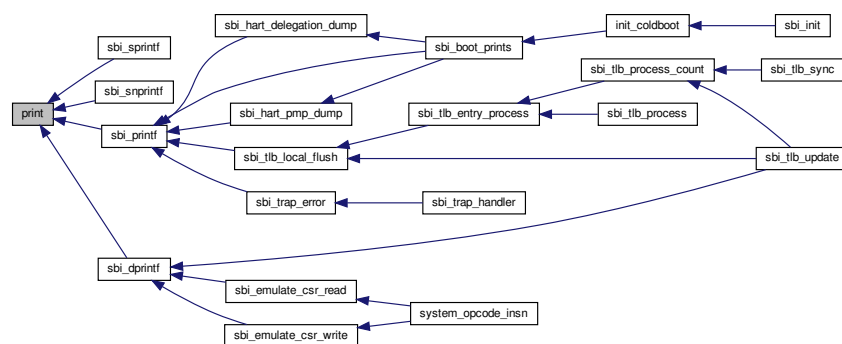
21.59.3.1 print()

```
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:



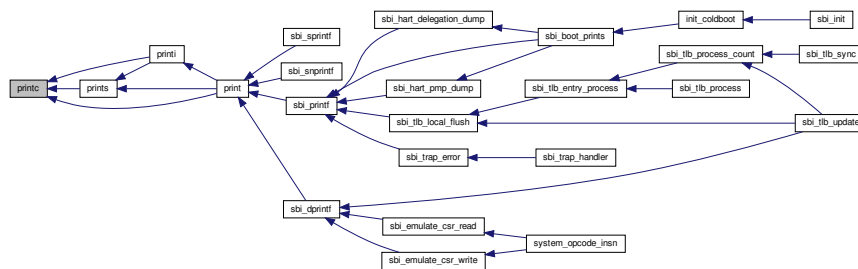
21.59.3.2 printc()

```
static void printc (
    char ** out,
    u32 * out_len,
    char ch ) [static]
```

Here is the call graph for this function:



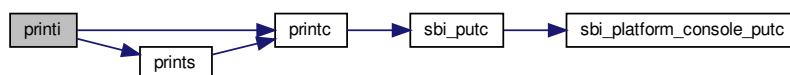
Here is the caller graph for this function:



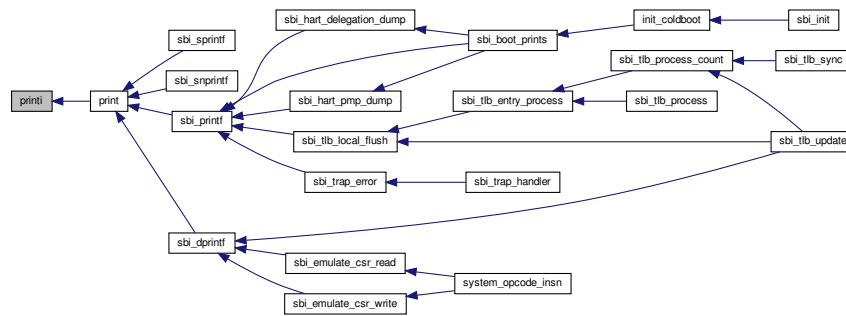
21.59.3.3 printi()

```
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:



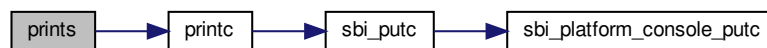
21.59.3.4 prints()

```

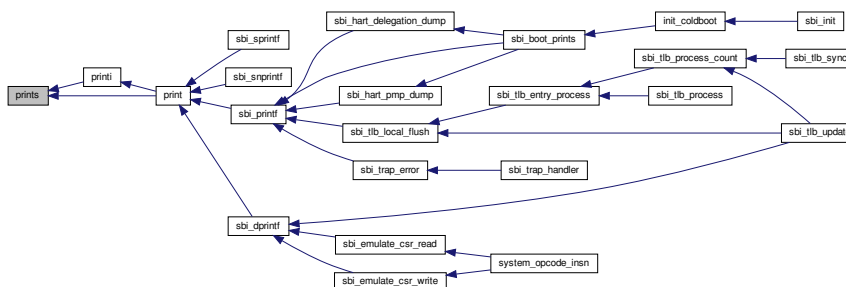
static int prints (
    char ** out,
    u32 * out_len,
    const char * string,
    int width,
    int flags ) [static]

```

Here is the call graph for this function:



Here is the caller graph for this function:



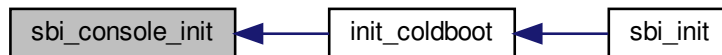
21.59.3.5 sbi_console_init()

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

Here is the call graph for this function:



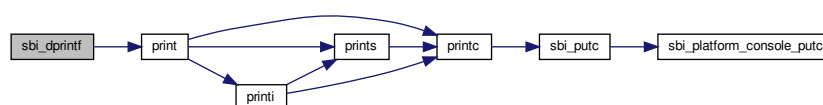
Here is the caller graph for this function:



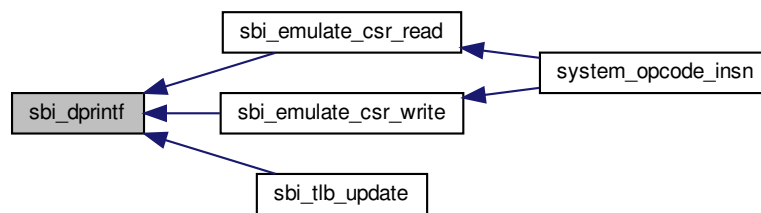
21.59.3.6 sbi_dprintf()

```
int sbi_dprintf (
    struct sbi_scratch * scratch,
    const char * format,
    ... )
```

Here is the call graph for this function:



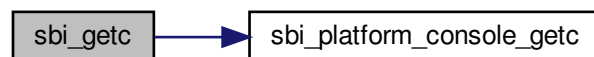
Here is the caller graph for this function:



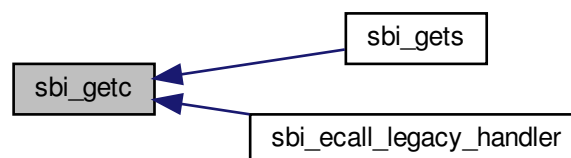
21.59.3.7 sbi_getc()

```
int sbi_getc (  
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.59.3.8 sbi_gets()

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

Here is the call graph for this function:



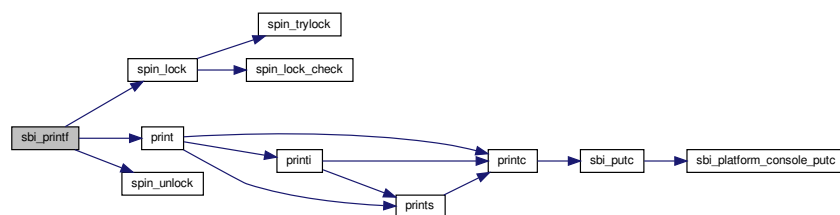
21.59.3.9 sbi_isprintable()

```
bool sbi_isprintable (
    char c )
```

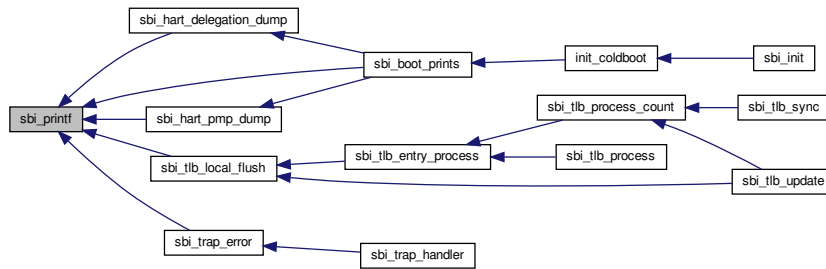
21.59.3.10 sbi_printf()

```
int sbi_printf (
    const char * format,
    ... )
```

Here is the call graph for this function:



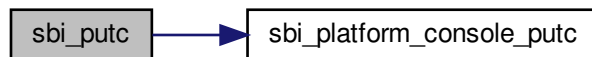
Here is the caller graph for this function:



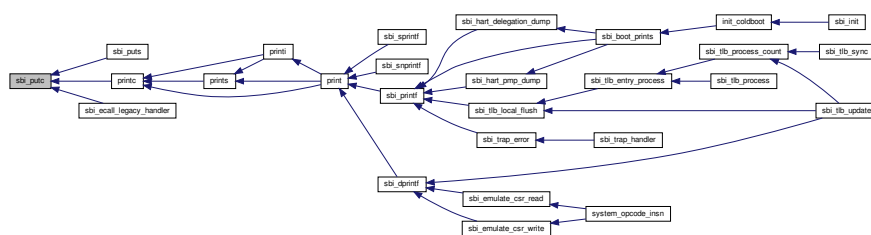
21.59.3.11 sbi_putc()

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

Here is the call graph for this function:



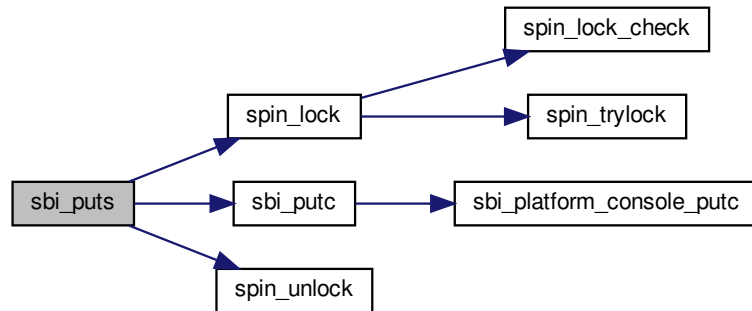
Here is the caller graph for this function:



21.59.3.12 sbi_puts()

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

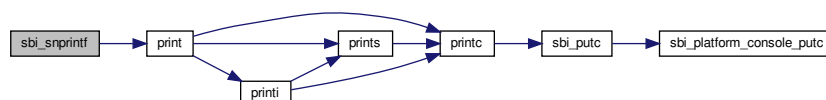
Here is the call graph for this function:



21.59.3.13 sbi_snprintf()

```
int sbi_snprintf (  
    char * out,  
    u32 out_sz,  
    const char * format,  
    ... )
```

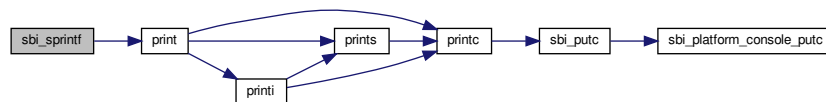
Here is the call graph for this function:



21.59.3.14 sbi_sprintf()

```
int sbi_sprintf (
    char * out,
    const char * format,
    ... )
```

Here is the call graph for this function:



21.59.4 Variable Documentation

21.59.4.1 console_out_lock

```
spinlock_t console_out_lock = SPIN_LOCK_INITIALIZER [static]
```

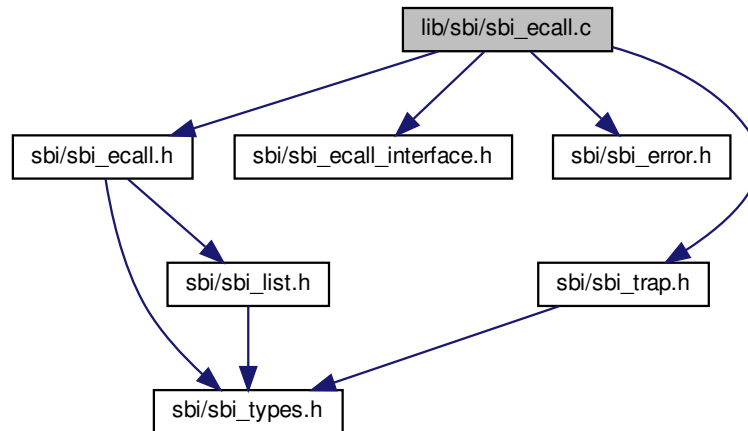
21.59.4.2 console_plat

```
const struct sbi_platform* console_plat = NULL [static]
```

21.60 lib/sbi/sbi_ecall.c File Reference

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

```
#include <sbi/sbi_trap.h>
Include dependency graph for sbi_ecall.c:
```



Functions

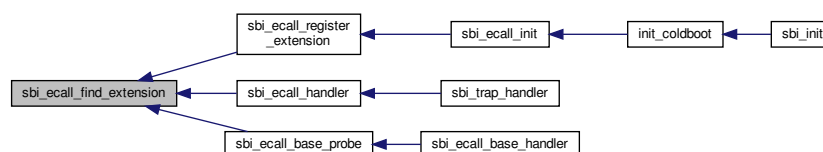
- [u16 sbi_ecall_version_major](#) (void)
- [u16 sbi_ecall_version_minor](#) (void)
- static [SBI_LIST_HEAD](#) (ecall_exts_list)
- struct [sbi_ecall_extension](#) * [sbi_ecall_find_extension](#) (unsigned long extid)
- int [sbi_ecall_register_extension](#) (struct [sbi_ecall_extension](#) *ext)
- void [sbi_ecall_unregister_extension](#) (struct [sbi_ecall_extension](#) *ext)
- int [sbi_ecall_handler](#) (u32 hartid, ulong mcause, struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch)
- int [sbi_ecall_init](#) (void)

21.60.1 Function Documentation

21.60.1.1 sbi_ecall_find_extension()

```
struct sbi\_ecall\_extension* sbi_ecall_find_extension (
    unsigned long extid )
```

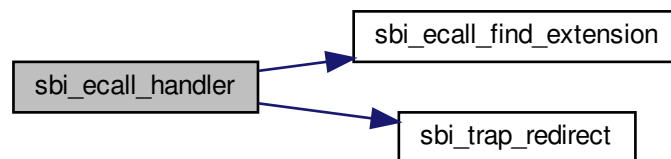
Here is the caller graph for this function:



21.60.1.2 sbi_ecall_handler()

```
int sbi_ecall_handler (
    u32 hartid,
    ulong mcause,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:



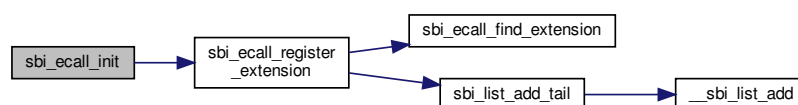
Here is the caller graph for this function:



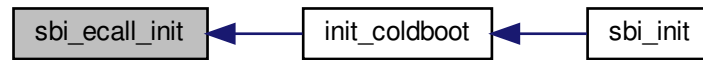
21.60.1.3 sbi_ecall_init()

```
int sbi_ecall_init (
    void )
```

Here is the call graph for this function:



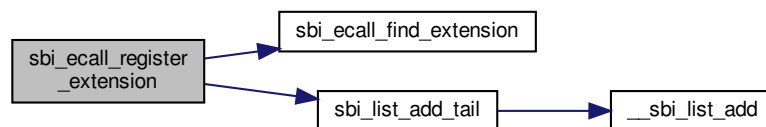
Here is the caller graph for this function:



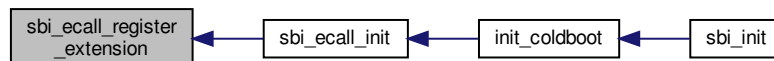
21.60.1.4 sbi_ecall_register_extension()

```
int sbi_ecall_register_extension (
    struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:



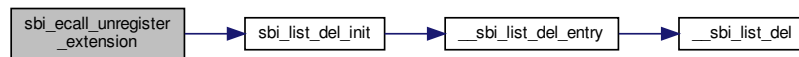
Here is the caller graph for this function:



21.60.1.5 sbi_ecall_unregister_extension()

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

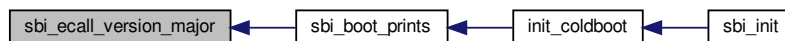
Here is the call graph for this function:



21.60.1.6 `sbi_ecall_version_major()`

```
u16 sbi_ecall_version_major (  
    void )
```

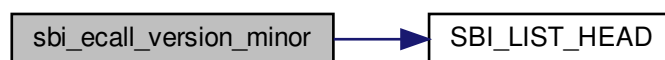
Here is the caller graph for this function:



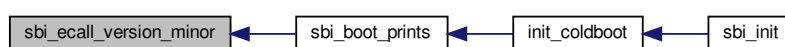
21.60.1.7 `sbi_ecall_version_minor()`

```
u16 sbi_ecall_version_minor (  
    void )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.60.1.8 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
    ecall_exts_list ) [static]
```

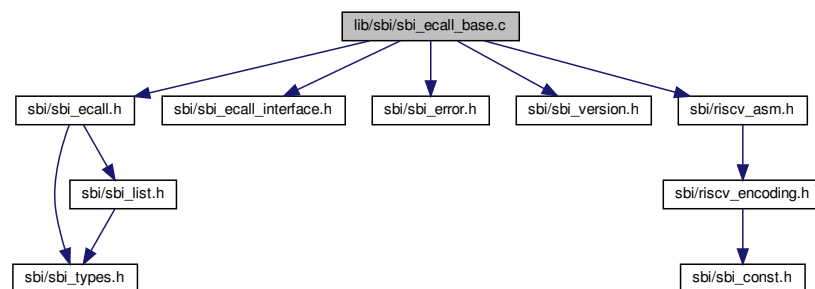
Here is the caller graph for this function:



21.61 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](#) (struct [sbi_scratch](#) *scratch, unsigned long extid, unsigned long *out_val)
- static int [sbi_ecall_base_handler](#) (struct [sbi_scratch](#) *scratch, unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct [sbi_trap_info](#) *out_trap)

Variables

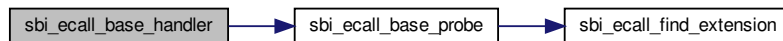
- struct [sbi_ecall_extension](#) [ecall_base](#)

21.61.1 Function Documentation

21.61.1.1 sbi_ecall_base_handler()

```
static int sbi_ecall_base_handler (  
    struct sbi_scratch * scratch,  
    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:



21.61.1.2 sbi_ecall_base_probe()

```
static int sbi_ecall_base_probe (  
    struct sbi_scratch * scratch,  
    unsigned long extid,  
    unsigned long * out_val ) [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.61.2 Variable Documentation

21.61.2.1 ecall_base

```
struct sbi_ecall_extension ecall_base
```

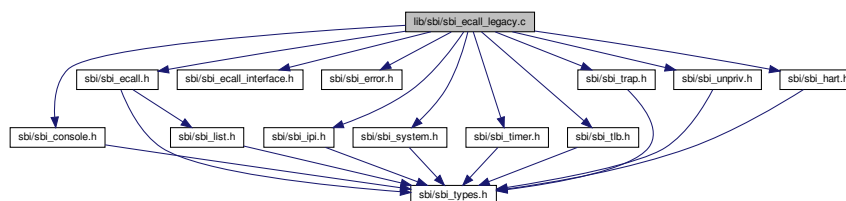
Initial value:

```
= {
    .extid_start = SBI_EXT_BASE,
    .extid_end   = SBI_EXT_BASE,
    .handle      = sbi_ecall_base_handler,
}
```

21.62 lib/sbi/sbi_ecall_legacy.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_ipi.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:



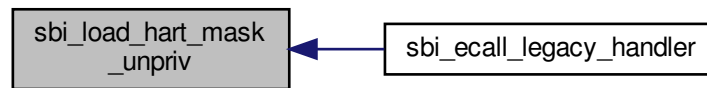
Functions

- static int [sbi_load_hart_mask_unpriv](#) (struct [sbi_scratch](#) *scratch, [ulong](#) *pmask, [ulong](#) *hmask, struct [sbi_trap_info](#) *uptrap)
- static int [sbi_ecall_legacy_handler](#) (struct [sbi_scratch](#) *scratch, 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](#)

Here is the caller graph for this function:



21.62.2 Variable Documentation

21.62.2.1 ecall_legacy

```
struct sbi_ecall_extension ecall_legacy
```

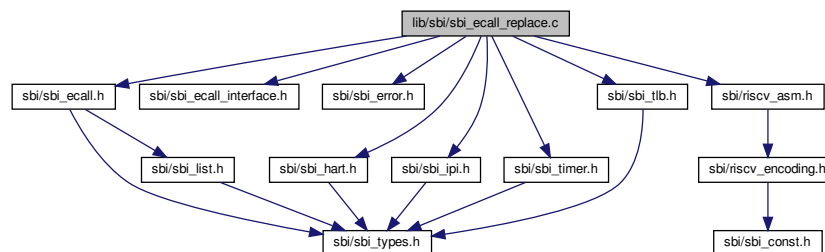
Initial value:

```
= {
    .extid_start = SBI_EXT_0_1_SET_TIMER,
    .extid_end   = SBI_EXT_0_1_SHUTDOWN,
    .handle      = sbi_ecall_legacy_handler,
}
```

21.63 lib/sbi/sbi_ecall_replace.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_tlb.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for `sbi_ecall_replace.c`:



Functions

- static int [sbi_ecall_time_handler](#) (struct [sbi_scratch](#) *scratch, 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](#) (struct [sbi_scratch](#) *scratch, 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](#) (struct [sbi_scratch](#) *scratch, unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct [sbi_trap_info](#) *out_trap)

Variables

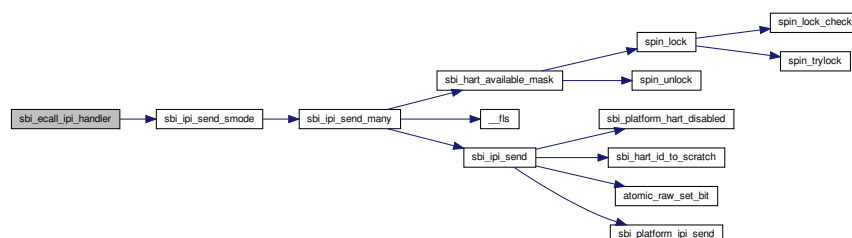
- struct [sbi_ecall_extension](#) [ecall_time](#)
- struct [sbi_ecall_extension](#) [ecall_rfence](#)
- struct [sbi_ecall_extension](#) [ecall_ipi](#)

21.63.1 Function Documentation

21.63.1.1 [sbi_ecall_ipi_handler\(\)](#)

```
static int sbi_ecall_ipi_handler (
    struct sbi\_scratch * scratch,
    unsigned long extid,
    unsigned long funcid,
    unsigned long * args,
    unsigned long * out_val,
    struct sbi\_trap\_info * out_trap ) [static]
```

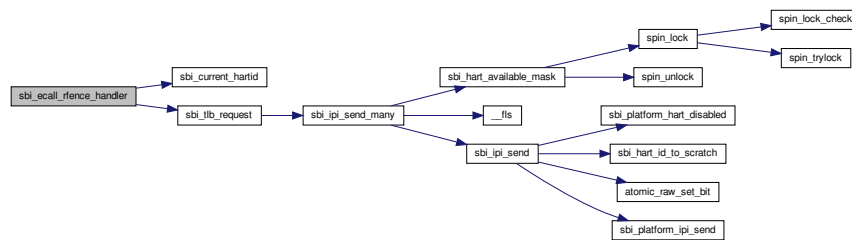
Here is the call graph for this function:



21.63.1.2 sbi_ecall_rfence_handler()

```
static int sbi_ecall_rfence_handler (
    struct sbi_scratch * scratch,
    unsigned long extid,
    unsigned long funcid,
    unsigned long * args,
    unsigned long * out_val,
    struct sbi_trap_info * out_trap ) [static]
```

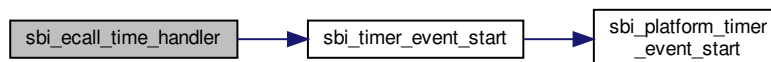
Here is the call graph for this function:



21.63.1.3 sbi_ecall_time_handler()

```
static int sbi_ecall_time_handler (
    struct sbi_scratch * scratch,
    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:



21.63.2 Variable Documentation

21.63.2.1 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,  
}
```

21.63.2.2 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,  
}
```

21.63.2.3 ecall_time

```
struct sbi_ecall_extension ecall_time
```

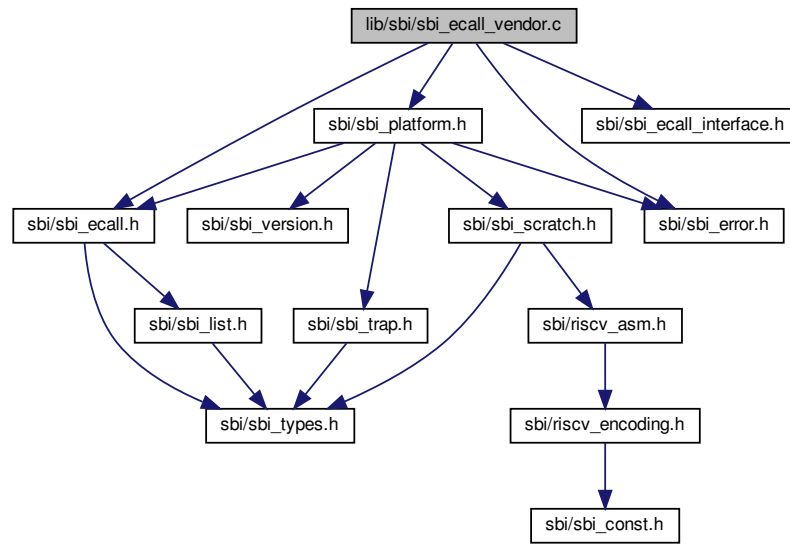
Initial value:

```
= {  
    .extid_start = SBI_EXT_TIME,  
    .extid_end = SBI_EXT_TIME,  
    .handle = sbi_ecall_time_handler,  
}
```

21.64 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](#) (struct [sbi_scratch](#) *scratch, unsigned long extid, unsigned long *out_val)
- static int [sbi_ecall_vendor_handler](#) (struct [sbi_scratch](#) *scratch, unsigned long extid, unsigned long funcid, unsigned long *args, unsigned long *out_val, struct [sbi_trap_info](#) *out_trap)

Variables

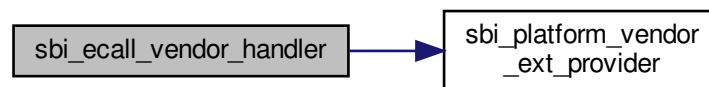
- struct [sbi_ecall_extension](#) [ecall_vendor](#)

21.64.1 Function Documentation

21.64.1.1 sbi_ecall_vendor_handler()

```
static int sbi_ecall_vendor_handler (  
    struct sbi_scratch * scratch,  
    unsigned long extid,  
    unsigned long funcid,  
    unsigned long * args,  
    unsigned long * out_val,  
    struct sbi_trap_info * out_trap ) [static]
```

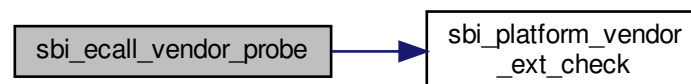
Here is the call graph for this function:



21.64.1.2 sbi_ecall_vendor_probe()

```
static int sbi_ecall_vendor_probe (  
    struct sbi_scratch * scratch,  
    unsigned long extid,  
    unsigned long * out_val ) [static]
```

Here is the call graph for this function:



21.64.2 Variable Documentation

21.64.2.1 ecall_vendor

```
struct sbi_ecall_extension ecall_vendor
```

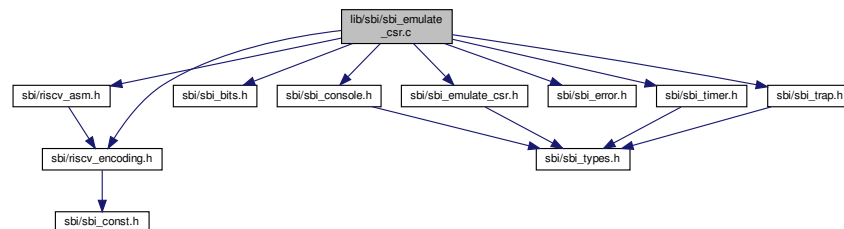
Initial value:

```
= {
    .extid_start = SBI_EXT_VENDOR_START,
    .extid_end = SBI_EXT_VENDOR_END,
    .probe = sbi_ecall_vendor_probe,
    .handle = sbi_ecall_vendor_handler,
}
```

21.65 lib/sbi/sbi_emulate_csr.c File Reference

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

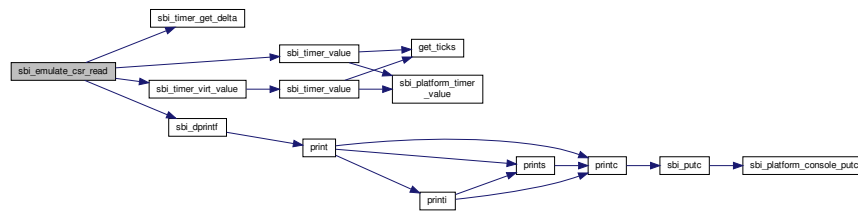
- int [sbi_emulate_csr_read](#) (int csr_num, u32 hartid, struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch, ulong *csr_val)
- int [sbi_emulate_csr_write](#) (int csr_num, u32 hartid, struct [sbi_trap_regs](#) *regs, struct [sbi_scratch](#) *scratch, ulong csr_val)

21.65.1 Function Documentation

21.65.1.1 sbi_emulate_csr_read()

```
int sbi_emulate_csr_read (
    int csr_num,
    u32 hartid,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch,
    ulong * csr_val )
```

Here is the call graph for this function:



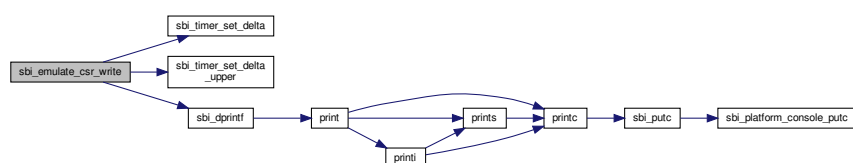
Here is the caller graph for this function:



21.65.1.2 sbi_emulate_csr_write()

```
int sbi_emulate_csr_write (
    int csr_num,
    u32 hartid,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch,
    ulong csr_val )
```

Here is the call graph for this function:



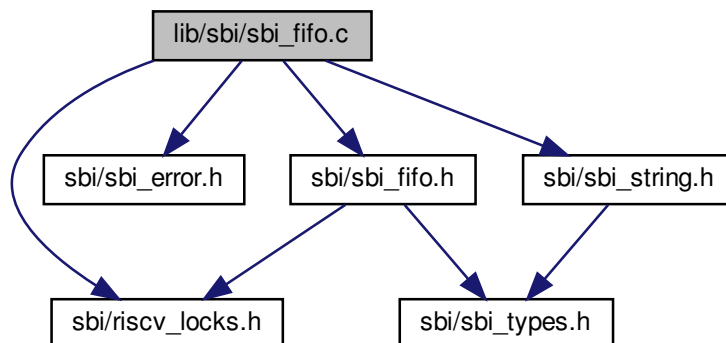
Here is the caller graph for this function:



21.66 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)

21.66.1 Function Documentation

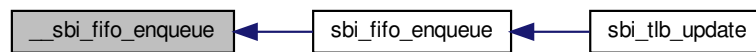
21.66.1.1 __sbi_fifo_enqueue()

```
static void __sbi_fifo_enqueue (
    struct sbi_fifo * fifo,
    void * data ) [inline], [static]
```

Here is the call graph for this function:



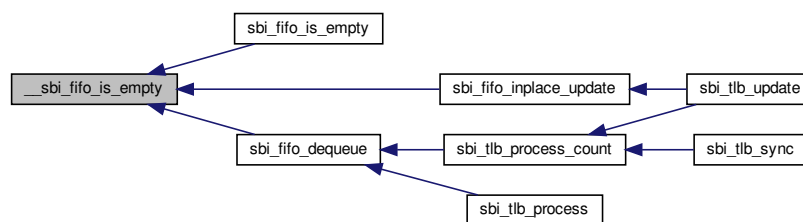
Here is the caller graph for this function:



21.66.1.2 __sbi_fifo_is_empty()

```
static bool __sbi_fifo_is_empty (
    struct sbi_fifo * fifo ) [inline], [static]
```

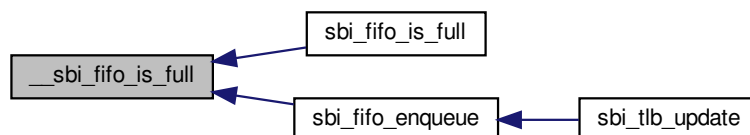
Here is the caller graph for this function:



21.66.1.3 __sbi_fifo_is_full()

```
static bool __sbi_fifo_is_full (
    struct sbi_fifo * fifo ) [inline], [static]
```

Here is the caller graph for this function:



21.66.1.4 __sbi_fifo_reset()

```
static void __sbi_fifo_reset (
    struct sbi_fifo * fifo ) [inline], [static]
```

Here is the call graph for this function:



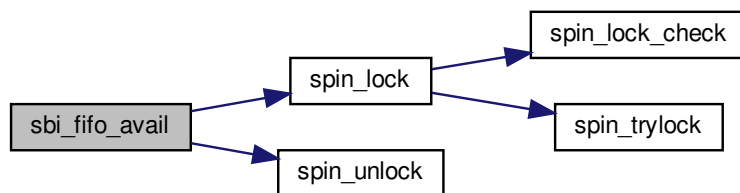
Here is the caller graph for this function:



21.66.1.5 sbi_fifo_avail()

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

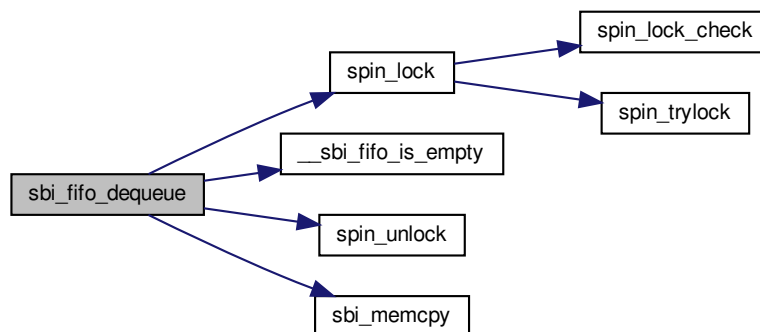
Here is the call graph for this function:



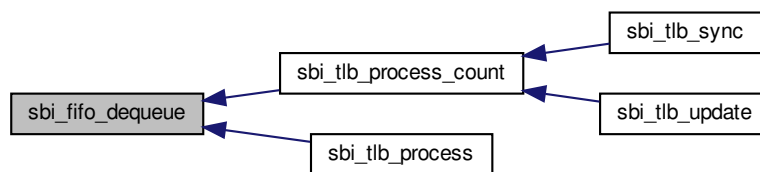
21.66.1.6 sbi_fifo_dequeue()

```
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:



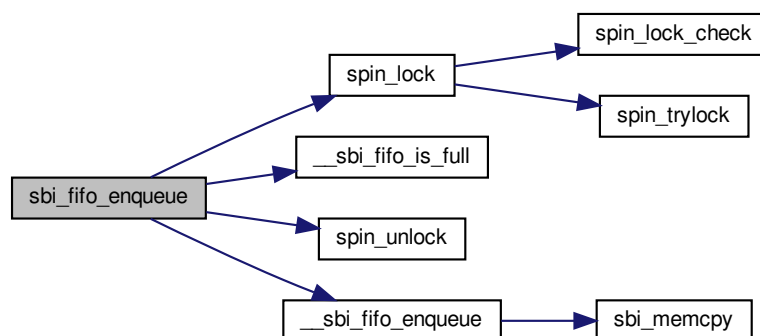
21.66.1.7 sbi_fifo_enqueue()

```

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:



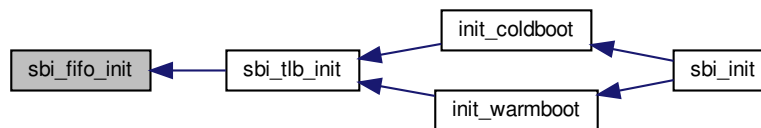
21.66.1.8 sbi_fifo_init()

```
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:

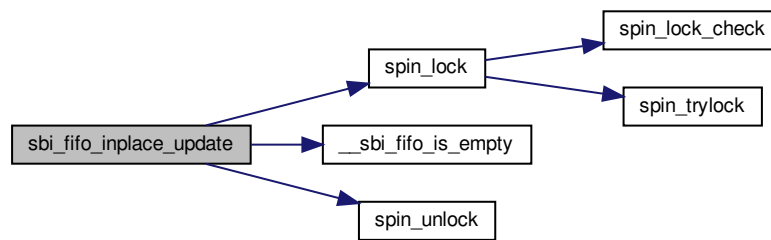


21.66.1.9 sbi_fifo_inplace_update()

```
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:



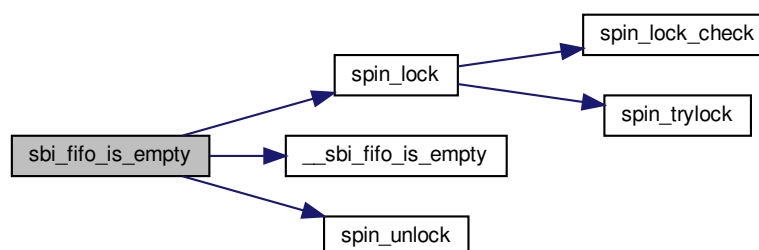
Here is the caller graph for this function:



21.66.1.10 sbi_fifo_is_empty()

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

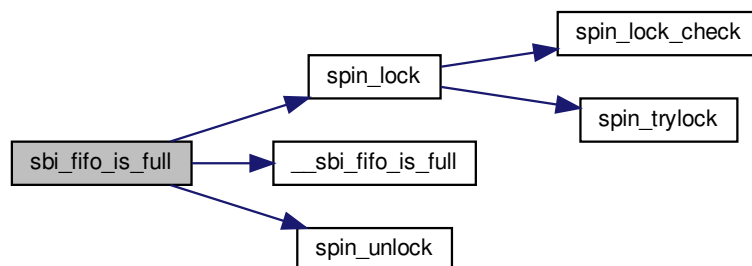
Here is the call graph for this function:



21.66.1.11 sbi_fifo_is_full()

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

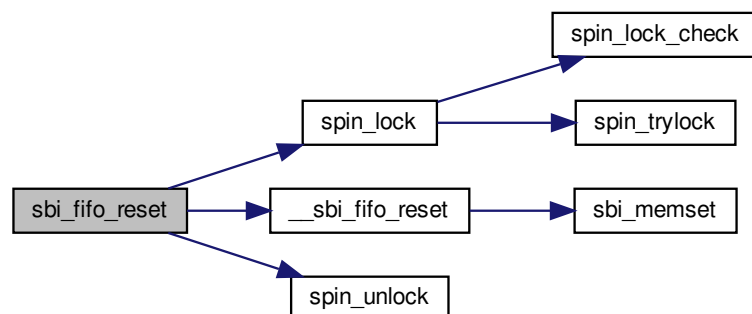
Here is the call graph for this function:



21.66.1.12 sbi_fifo_reset()

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

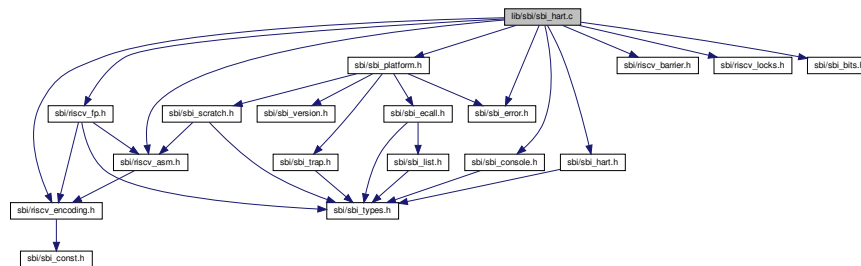
Here is the call graph for this function:



21.67 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/riscv_locks.h>
#include <sbi/sbi_bits.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
```

Include dependency graph for sbi_hart.c:



Macros

- #define `COLDBOOT_WAIT_BITMAP_SIZE` `__riscv_xlen`

Typedefs

- typedef struct `sbi_scratch` `*(h2s) (ulong hartid)`

Functions

- unsigned int `sbi_current_hartid` ()
- 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 long `log2roundup` (unsigned long x)
- void `sbi_hart_pmp_dump` (struct `sbi_scratch` *scratch)
- static int `pmp_init` (struct `sbi_scratch` *scratch, u32 hartid)
- int `sbi_hart_init` (struct `sbi_scratch` *scratch, u32 hartid, bool cold_boot)
- void * `sbi_hart_get_trap_info` (struct `sbi_scratch` *scratch)
- void `sbi_hart_set_trap_info` (struct `sbi_scratch` *scratch, void *data)
- void `__attribute__` ((noreturn))
- void `sbi_hart_mark_available` (u32 hartid)
- void `sbi_hart_unmark_available` (u32 hartid)
- ulong `sbi_hart_available_mask` (void)
- struct `sbi_scratch` * `sbi_hart_id_to_scratch` (struct `sbi_scratch` *scratch, u32 hartid)
- void `sbi_hart_wait_for_coldboot` (struct `sbi_scratch` *scratch, u32 hartid)
- void `sbi_hart_wake_coldboot_harts` (struct `sbi_scratch` *scratch, u32 hartid)

Variables

- static unsigned long `trap_info_offset`
- static `spinlock_t` `avail_hart_mask_lock` = `SPIN_LOCK_INITIALIZER`
- static volatile unsigned long `avail_hart_mask` = 0
- static `spinlock_t` `coldboot_lock` = `SPIN_LOCK_INITIALIZER`
- static unsigned long `coldboot_done` = 0
- static unsigned long `coldboot_wait_bitmap` = 0

21.67.1 Macro Definition Documentation

21.67.1.1 COLDBOOT_WAIT_BITMAP_SIZE

```
#define COLDBOOT_WAIT_BITMAP_SIZE __riscv_xlen
```

21.67.2 Typedef Documentation

21.67.2.1 h2s

```
typedef struct sbi_scratch*(h2s) (ulong hartid)
```

21.67.3 Function Documentation

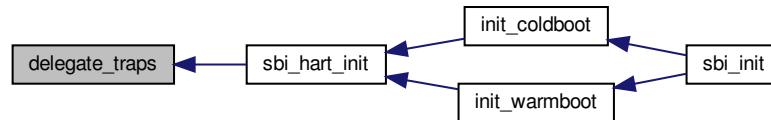
21.67.3.1 __attribute__()

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

21.67.3.2 delegate_traps()

```
static int delegate_traps (
    struct sbi_scratch * scratch,
    u32 hartid ) [static]
```

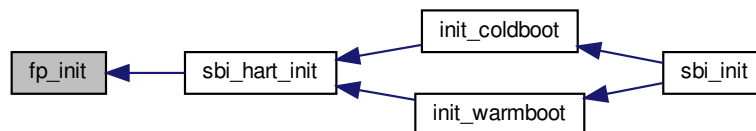
Here is the caller graph for this function:



21.67.3.3 fp_init()

```
static int fp_init (
    u32 hartid ) [static]
```

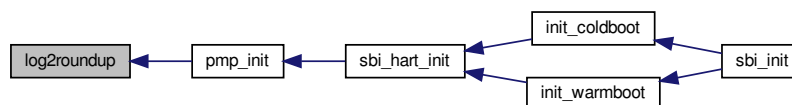
Here is the caller graph for this function:



21.67.3.4 log2roundup()

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

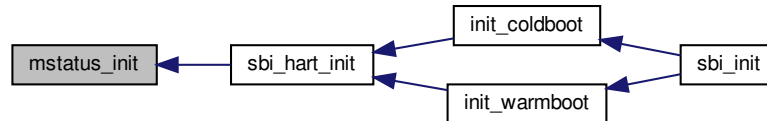
Here is the caller graph for this function:



21.67.3.5 mstatus_init()

```
static void mstatus_init (
    struct sbi_scratch * scratch,
    u32 hartid ) [static]
```

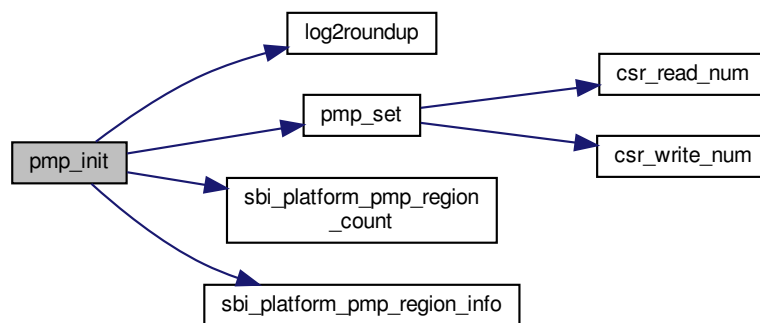
Here is the caller graph for this function:



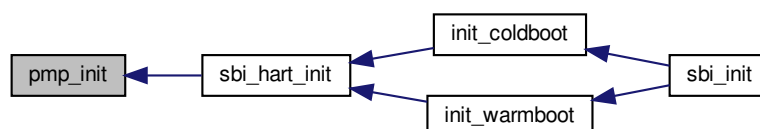
21.67.3.6 pmp_init()

```
static int pmp_init (
    struct sbi_scratch * scratch,
    u32 hartid ) [static]
```

Here is the call graph for this function:



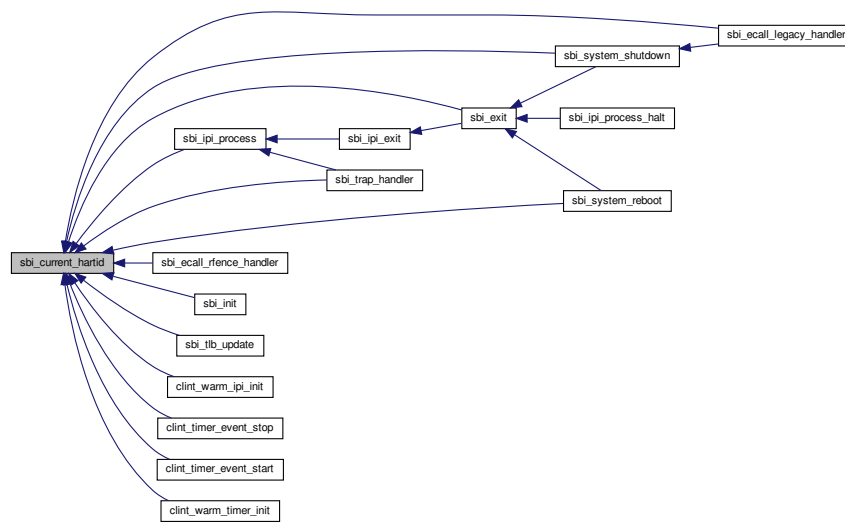
Here is the caller graph for this function:



21.67.3.7 sbi_current_hartid()

```
unsigned int sbi_current_hartid (
    void )
```

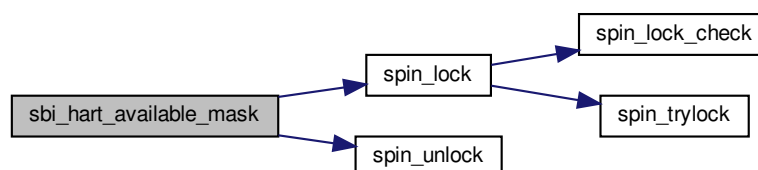
Return HART ID of the caller. Here is the caller graph for this function:



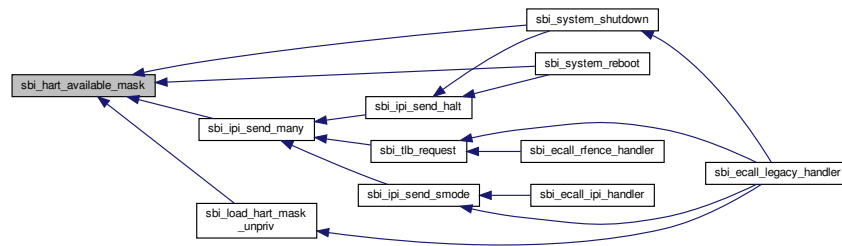
21.67.3.8 sbi_hart_available_mask()

```
ulong sbi_hart_available_mask (
    void )
```

Here is the call graph for this function:



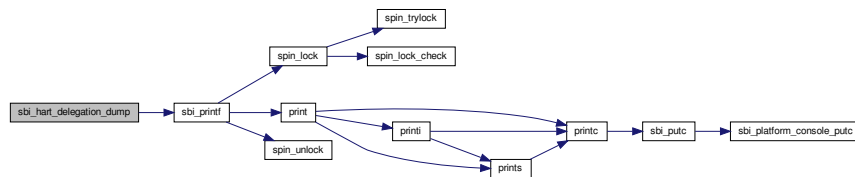
Here is the caller graph for this function:



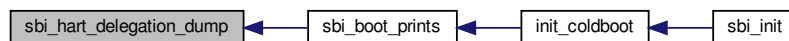
21.67.3.9 sbi_hart_delegation_dump()

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

Here is the call graph for this function:



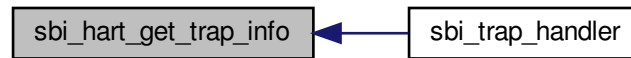
Here is the caller graph for this function:



21.67.3.10 sbi_hart_get_trap_info()

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

Here is the caller graph for this function:

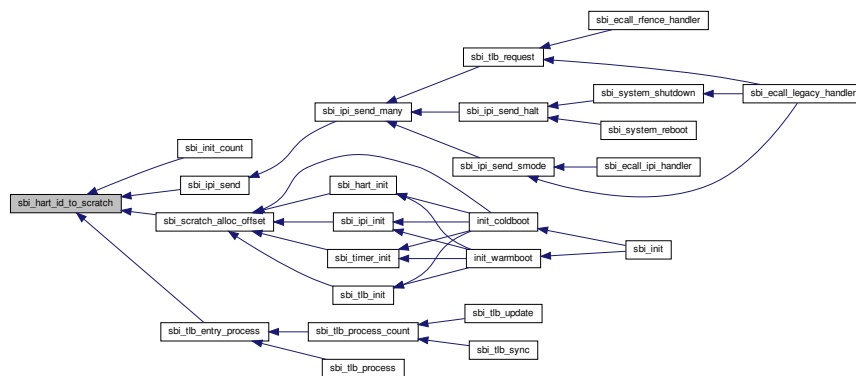


21.67.3.11 sbi_hart_id_to_scratch()

```

struct sbi_scratch* sbi_hart_id_to_scratch (
    struct sbi_scratch * scratch,
    u32 hartid )
  
```

Here is the caller graph for this function:

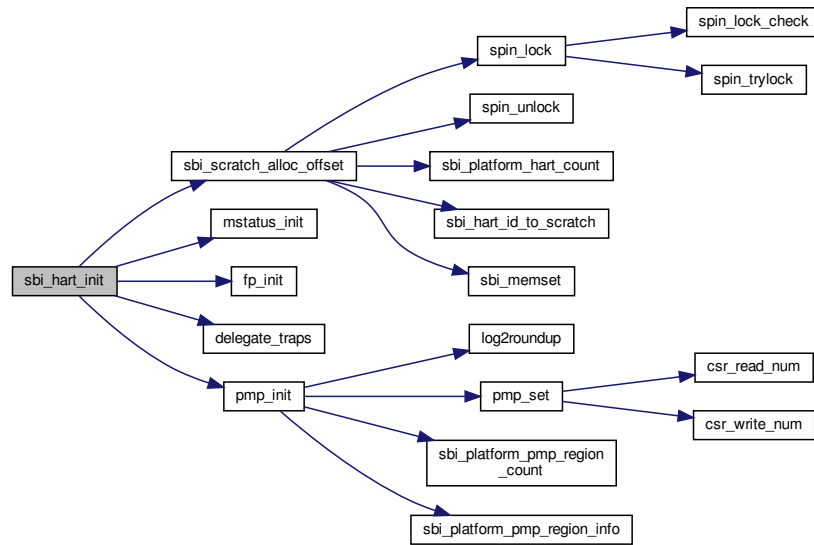


21.67.3.12 sbi_hart_init()

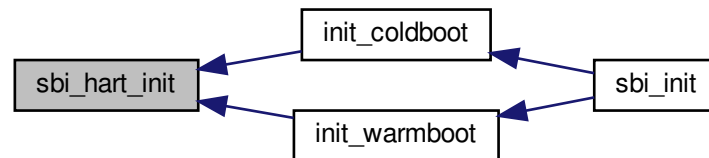
```

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:

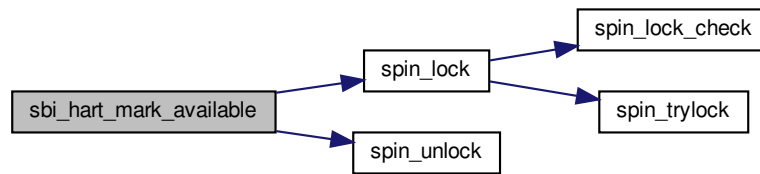


21.67.3.13 sbi_hart_mark_available()

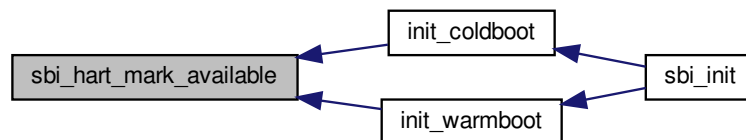
```

void sbi_hart_mark_available (
    u32 hartid )
  
```

Here is the call graph for this function:



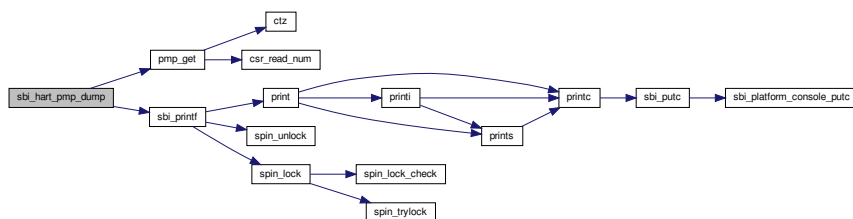
Here is the caller graph for this function:



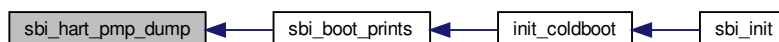
21.67.3.14 sbi_hart_pmp_dump()

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

Here is the call graph for this function:



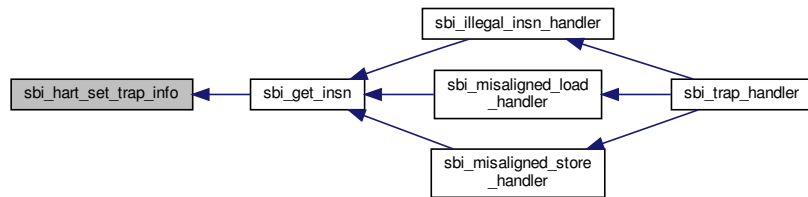
Here is the caller graph for this function:



21.67.3.15 sbi_hart_set_trap_info()

```
void sbi_hart_set_trap_info (
    struct sbi_scratch * scratch,
    void * data )
```

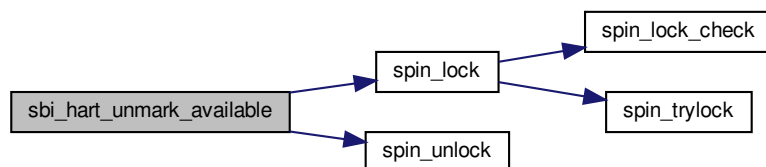
Here is the caller graph for this function:



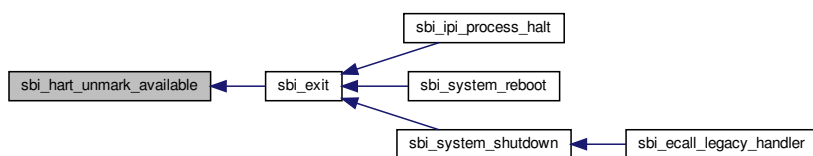
21.67.3.16 sbi_hart_unmark_available()

```
void sbi_hart_unmark_available (
    u32 hartid )
```

Here is the call graph for this function:



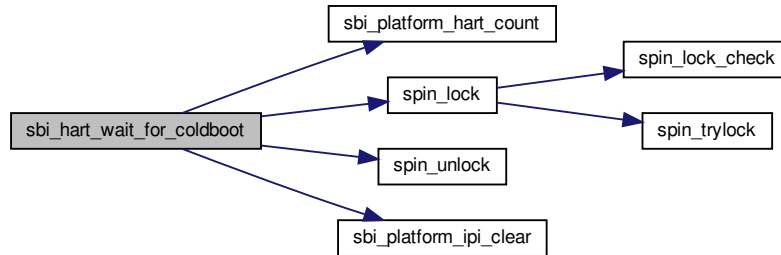
Here is the caller graph for this function:



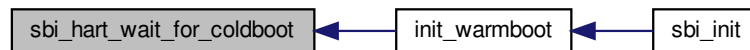
21.67.3.17 sbi_hart_wait_for_coldboot()

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

Here is the call graph for this function:



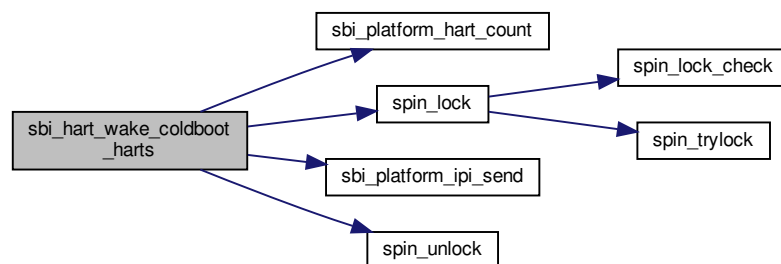
Here is the caller graph for this function:



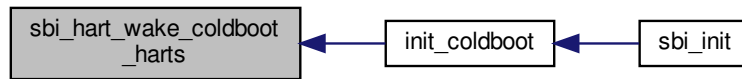
21.67.3.18 sbi_hart_wake_coldboot_harts()

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

Here is the call graph for this function:



Here is the caller graph for this function:



21.67.4 Variable Documentation

21.67.4.1 `avail_hart_mask`

```
volatile unsigned long avail_hart_mask = 0 [static]
```

21.67.4.2 `avail_hart_mask_lock`

```
spinlock_t avail_hart_mask_lock = SPIN_LOCK_INITIALIZER [static]
```

21.67.4.3 `coldboot_done`

```
unsigned long coldboot_done = 0 [static]
```

21.67.4.4 `coldboot_lock`

```
spinlock_t coldboot_lock = SPIN_LOCK_INITIALIZER [static]
```

21.67.4.5 `coldboot_wait_bitmap`

```
unsigned long coldboot_wait_bitmap = 0 [static]
```

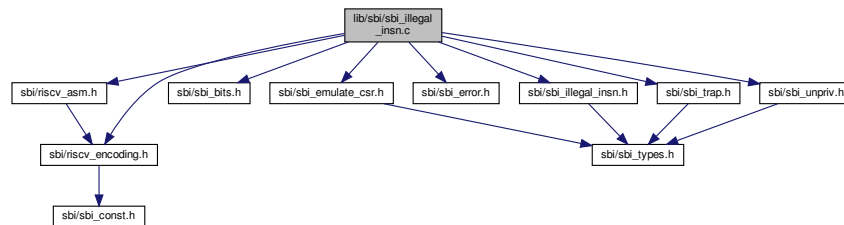
21.67.4.6 trap_info_offset

```
unsigned long trap_info_offset [static]
```

21.68 lib/sbi/sbi_illegal_insn.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bits.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, `u32` hartid, `ulong` mcause, struct `sbi_trap_regs` *regs, struct `sbi_trap_info` *scratch *scratch)

Functions

- static int `truly_illegal_insn` (`ulong` insn, `u32` hartid, `ulong` mcause, struct `sbi_trap_regs` *regs, struct `sbi_trap_info` *scratch *scratch)
- static int `system_opcode_insn` (`ulong` insn, `u32` hartid, `ulong` mcause, struct `sbi_trap_regs` *regs, struct `sbi_trap_info` *scratch *scratch)
- int `sbi_illegal_insn_handler` (`u32` hartid, `ulong` mcause, `ulong` insn, struct `sbi_trap_regs` *regs, struct `sbi_trap_info` *scratch *scratch)

Variables

- static `illegal_insn_func` `illegal_insn_table` [32]

21.68.1 Typedef Documentation

21.68.1.1 illegal_insn_func

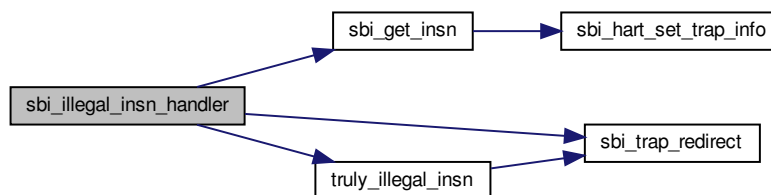
```
typedef int(* illegal_insn_func) (ulong insn, u32 hartid, ulong mcause, struct sbi_trap_regs  
*regs, struct sbi_scratch *scratch)
```

21.68.2 Function Documentation

21.68.2.1 sbi_illegal_insn_handler()

```
int sbi_illegal_insn_handler (  
    u32 hartid,  
    ulong mcause,  
    ulong insn,  
    struct sbi_trap_regs * regs,  
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:



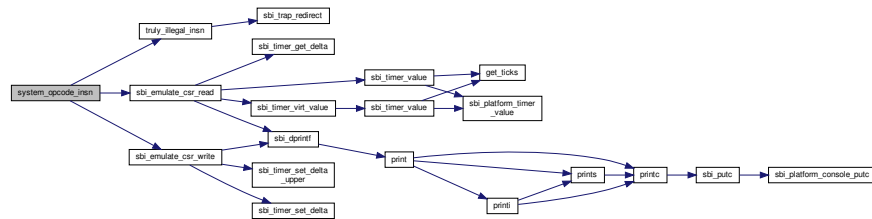
Here is the caller graph for this function:



21.68.2.2 system_opcode_insn()

```
static int system_opcode_insn (
    ulong insn,
    u32 hartid,
    ulong mcause,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch ) [static]
```

Here is the call graph for this function:



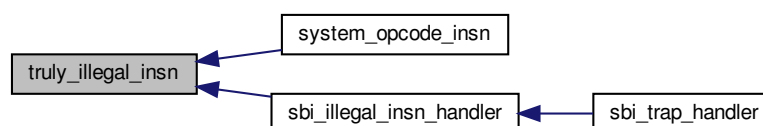
21.68.2.3 truly_illegal_insn()

```
static int truly_illegal_insn (
    ulong insn,
    u32 hartid,
    ulong mcause,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch ) [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.68.3 Variable Documentation

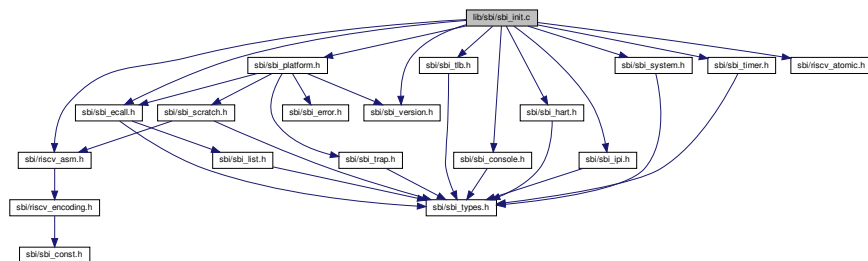
21.68.3.1 illegal_insn_table

```
illegal_insn_func illegal_insn_table[32] [static]
```

21.69 lib/sbi/sbi_init.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_hart.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_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 **__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)

- static unsigned long `init_count_offset`
- static `atomic_t` `coldboot_lottery` = `ATOMIC_INITIALIZER(0)`

21.69.1.1 BANNER

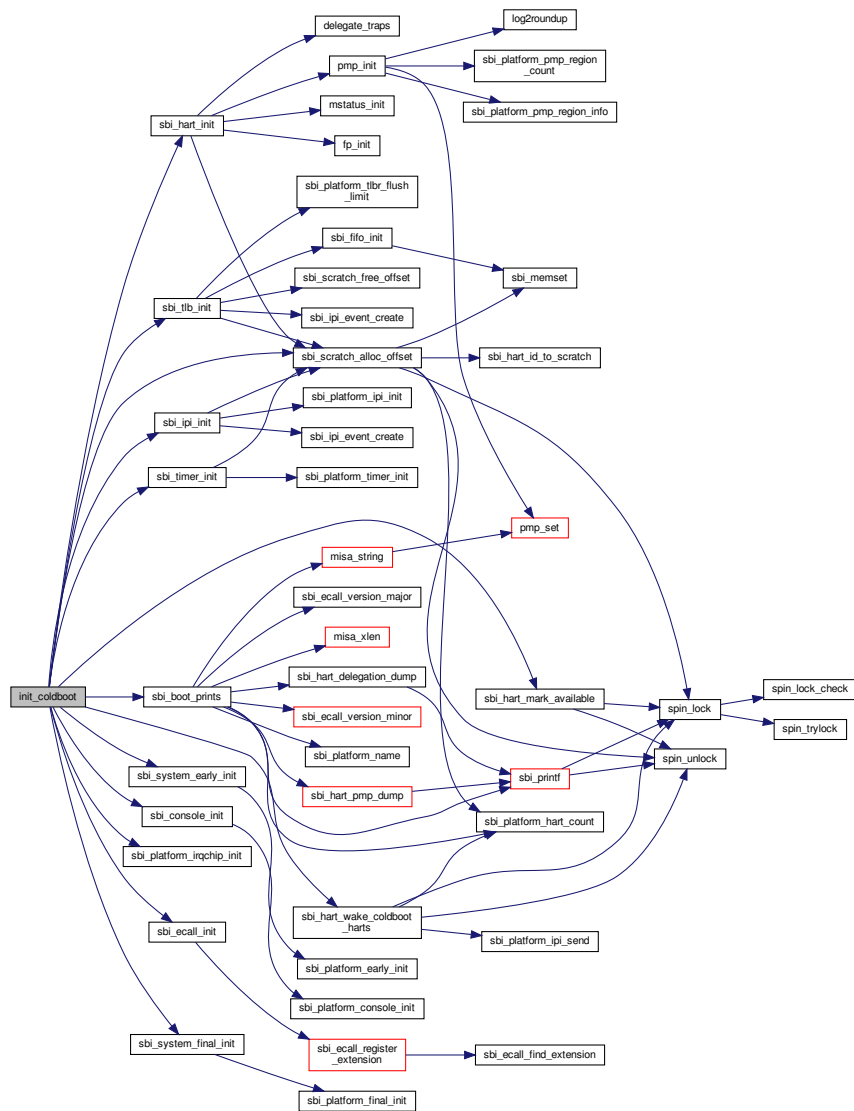
Value:

[illegible]

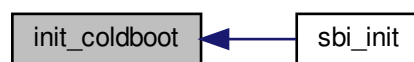
21.69.2.1 init_coldboot()

```
static void __noreturn init_coldboot (
    struct sbi_scratch * scratch,
    u32 hartid ) [static]
```

Here is the call graph for this function:



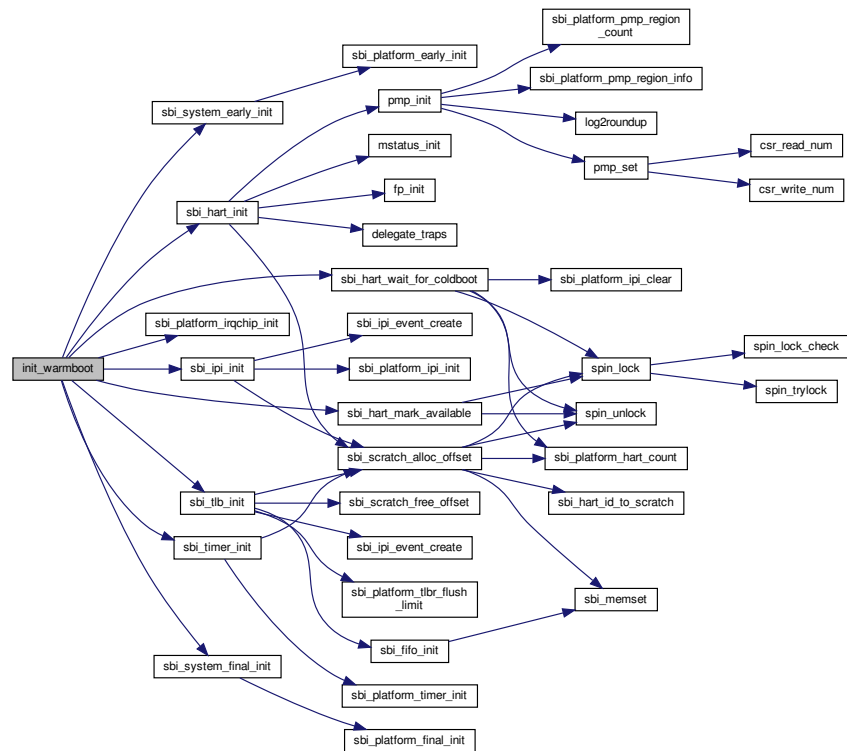
Here is the caller graph for this function:



21.69.2.2 init_warmboot()

```
static void __noreturn init_warmboot (
    struct sbi_scratch * scratch,
    u32 hartid ) [static]
```

Here is the call graph for this function:



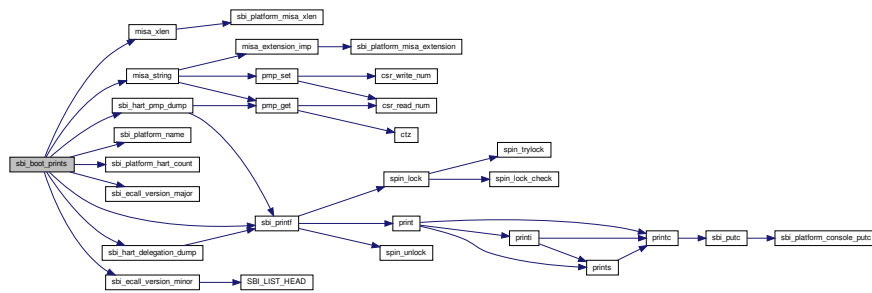
Here is the caller graph for this function:



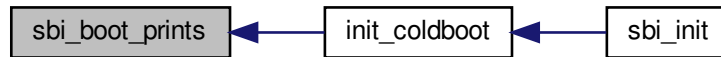
21.69.2.3 sbi_boot_prints()

```
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:



21.69.2.4 sbi_exit()

```
void __noreturn sbi_exit (
    struct sbi_scratch * scratch )
```

Exit OpenSBI library for current HART and stop HART

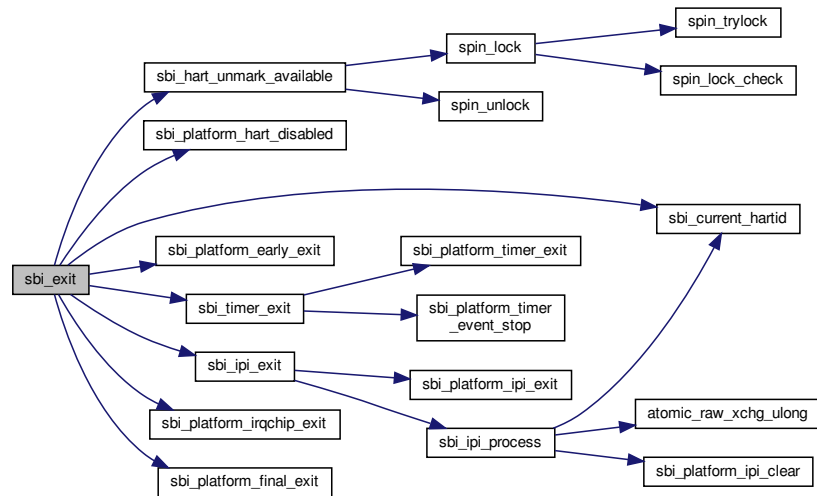
The function expects following:

1. The 'mscratch' CSR is pointing to [sbi_scratch](#) of current HART
2. Stack pointer (SP) is setup for current HART

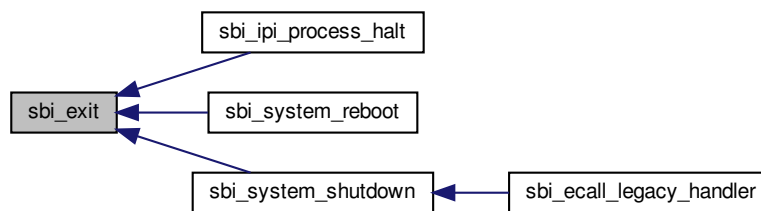
Parameters

<i>scratch</i>	pointer to sbi_scratch of current HART
----------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



21.69.2.5 sbi_init()

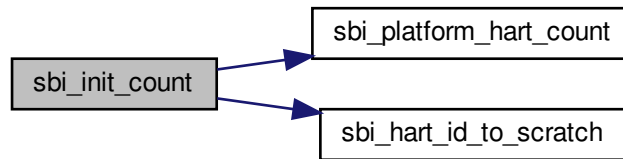
```
void __noreturn sbi_init (
    struct sbi_scratch * scratch )
```

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

Here is the call graph for this function:



21.69.3 Variable Documentation

21.69.3.1 coldboot_lottery

```
atomic_t coldboot_lottery = ATOMIC_INITIALIZER(0) [static]
```

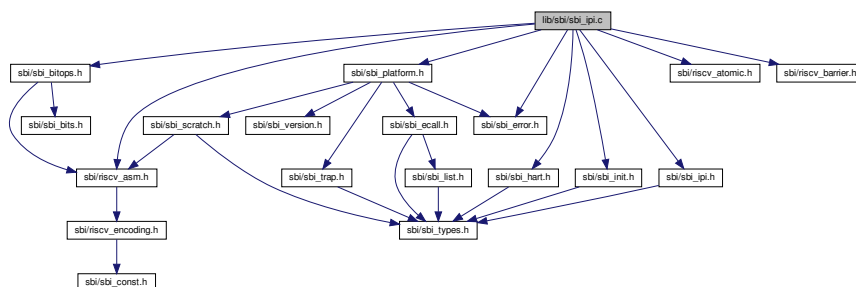
21.69.3.2 init_count_offset

```
unsigned long init_count_offset [static]
```

21.70 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_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` (struct `sbi_scratch` *scratch, `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` (struct `sbi_scratch` *scratch, `ulong` hmask, `ulong` hbase)
- void `sbi_ipi_clear_smode` (struct `sbi_scratch` *scratch)
- static void `sbi_ipi_process_halt` (struct `sbi_scratch` *scratch)
- int `sbi_ipi_send_halt` (struct `sbi_scratch` *scratch, `ulong` hmask, `ulong` hbase)
- void `sbi_ipi_process` (struct `sbi_scratch` *scratch)
- 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`

21.70.1 Function Documentation

21.70.1.1 `sbi_ipi_clear_smode()`

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

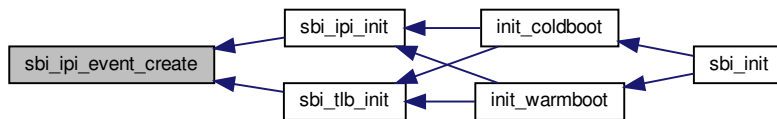
Here is the caller graph for this function:



21.70.1.2 sbi_ipi_event_create()

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

Here is the caller graph for this function:



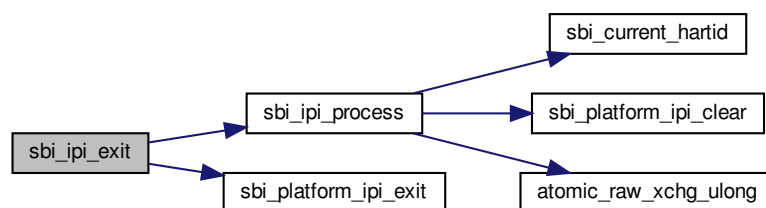
21.70.1.3 sbi_ipi_event_destroy()

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

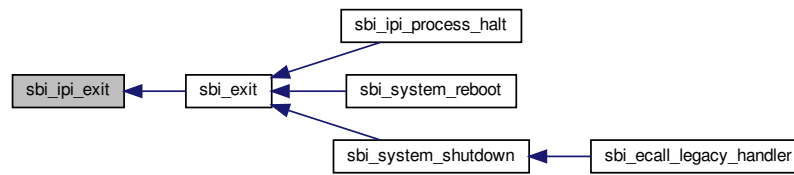
21.70.1.4 sbi_ipi_exit()

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

Here is the call graph for this function:



Here is the caller graph for this function:



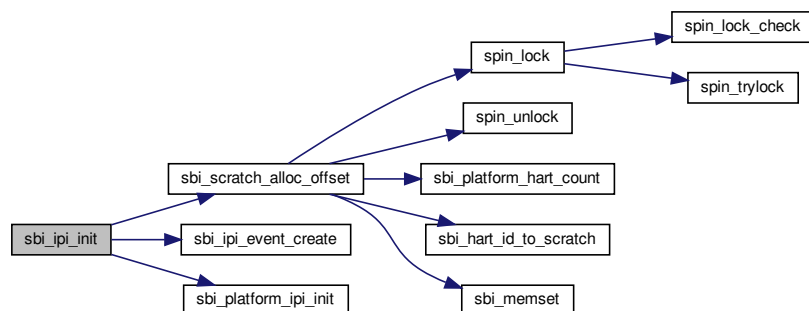
21.70.1.5 sbi_ipi_init()

```

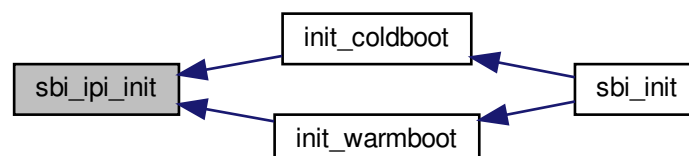
int sbi_ipi_init (
    struct sbi_scratch * scratch,
    bool cold_boot )

```

Here is the call graph for this function:



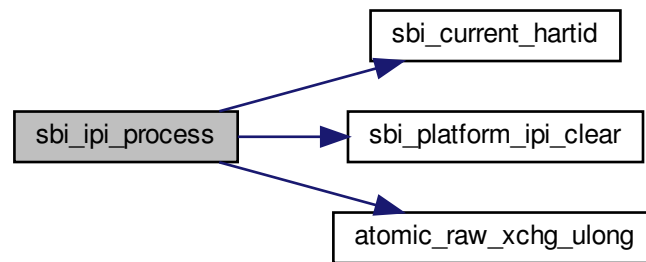
Here is the caller graph for this function:



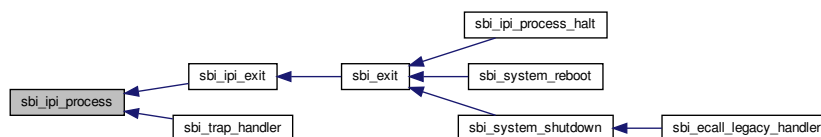
21.70.1.6 sbi_ipi_process()

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

Here is the call graph for this function:



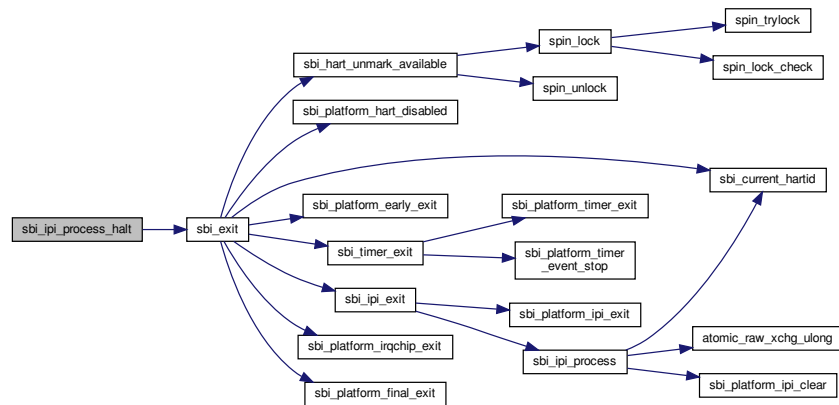
Here is the caller graph for this function:



21.70.1.7 sbi_ipi_process_halt()

```
static void sbi_ipi_process_halt (
    struct sbi_scratch * scratch ) [static]
```

Here is the call graph for this function:



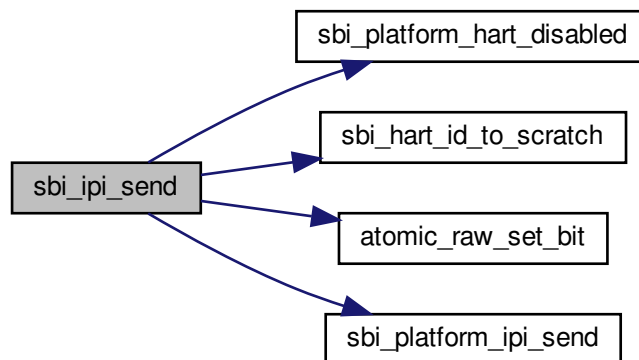
21.70.1.8 sbi_ipi_process_smode()

```
static void sbi_ipi_process_smode (
    struct sbi_scratch * scratch ) [static]
```

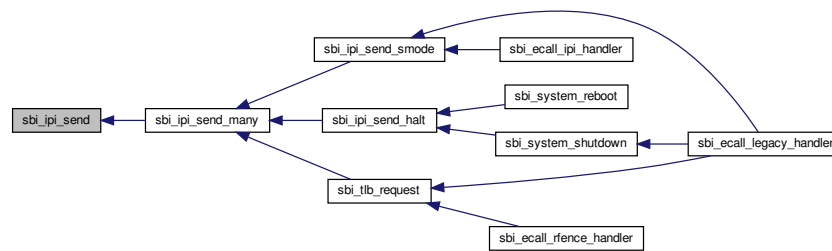
21.70.1.9 sbi_ipi_send()

```
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:



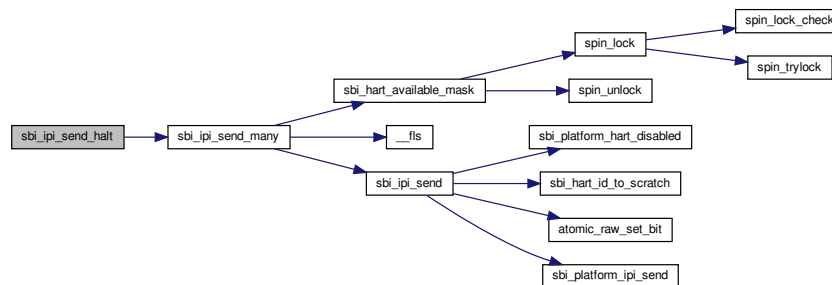
21.70.1.10 sbi_ipi_send_halt()

```

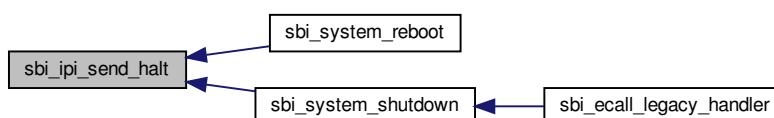
int sbi_ipi_send_halt (
    struct sbi_scratch * scratch,
    ulong hmask,
    ulong hbase )

```

Here is the call graph for this function:



Here is the caller graph for this function:



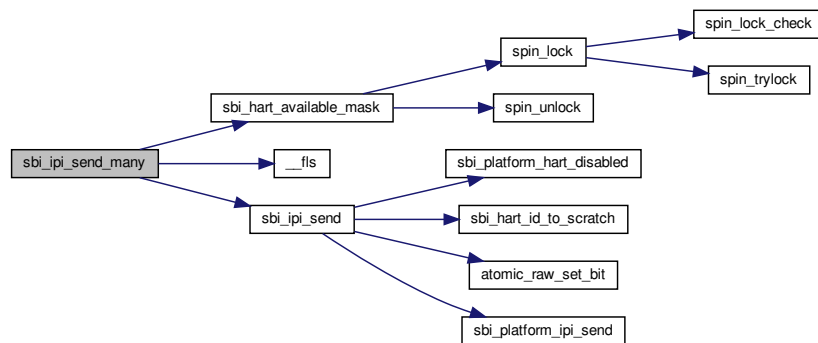
21.70.1.11 sbi_ipi_send_many()

```

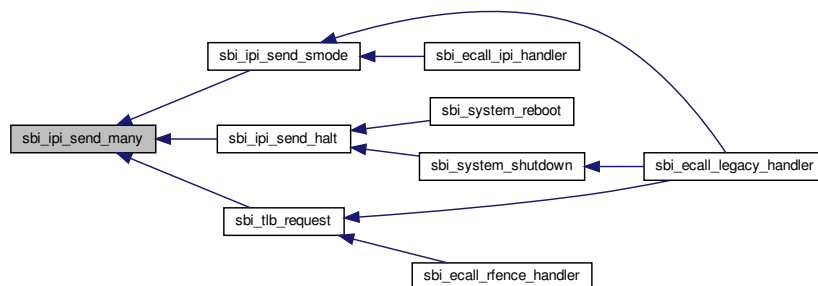
int sbi_ipi_send_many (
    struct sbi_scratch * scratch,
    ulong hmask,
    ulong hbase,
    u32 event,
    void * data )

```

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. FIXME: This check is valid only ULONG size. This is okay for now as avaialble hart mask can support upto ULONG size only. Here is the call graph for this function:



Here is the caller graph for this function:



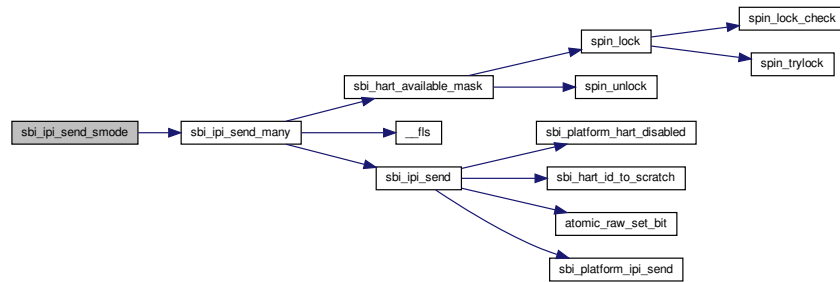
21.70.1.12 sbi_ipi_send_smode()

```

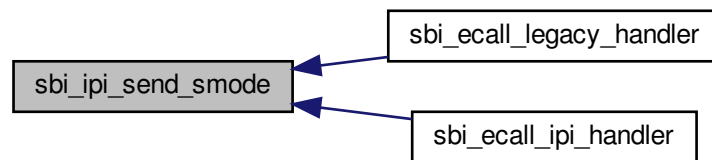
int sbi_ipi_send_smode (
    struct sbi_scratch * scratch,
    ulong hmask,
    ulong hbase )

```

Here is the call graph for this function:



Here is the caller graph for this function:



21.70.2 Variable Documentation

21.70.2.1 ipi_data_off

```
unsigned long ipi_data_off [static]
```

21.70.2.2 ipi_halt_event

```
u32 ipi_halt_event = SBI_IPI_EVENT_MAX [static]
```


21.70.2.3 ipi_halt_ops

```
struct sbi_ipi_event_ops ipi_halt_ops [static]
```

Initial value:

```
= {  
    .name = "IPI_HALT",  
    .process = sbi_ipi_process_halt,  
}
```

21.70.2.4 ipi_ops_array

```
const struct sbi_ipi_event_ops* ipi_ops_array[SBI_IPI_EVENT_MAX] [static]
```

21.70.2.5 ipi_smode_event

```
u32 ipi_smode_event = SBI_IPI_EVENT_MAX [static]
```

21.70.2.6 ipi_smode_ops

```
struct sbi_ipi_event_ops ipi_smode_ops [static]
```

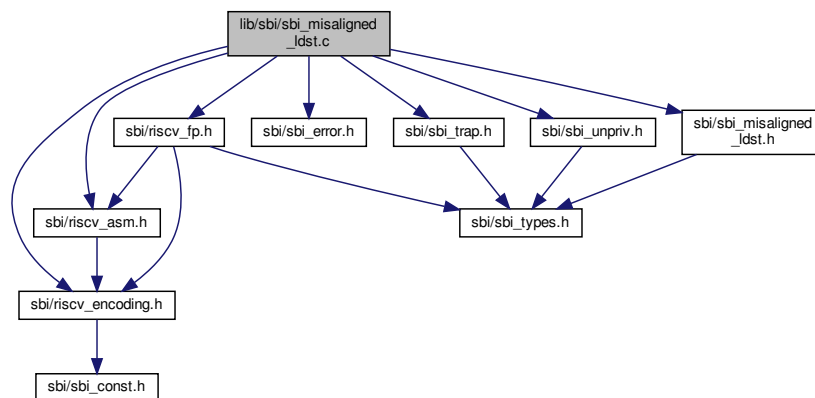
Initial value:

```
= {  
    .name = "IPI_SMODE",  
    .process = sbi_ipi_process_smode,  
}
```

21.71 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

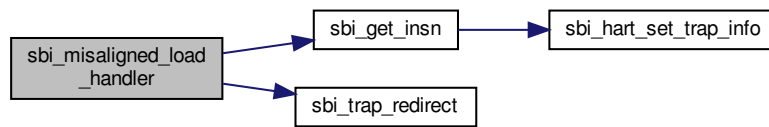
- `int sbi_misaligned_load_handler (u32 hartid, ulong mcause, ulong addr, ulong tval2, ulong tinst, struct sbi_trap_regs *regs, struct sbi_scratch *scratch)`
- `int sbi_misaligned_store_handler (u32 hartid, ulong mcause, ulong addr, ulong tval2, ulong tinst, struct sbi_trap_regs *regs, struct sbi_scratch *scratch)`

21.71.1 Function Documentation

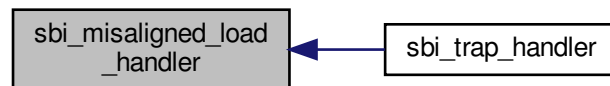
21.71.1.1 sbi_misaligned_load_handler()

```
int sbi_misaligned_load_handler (
    u32 hartid,
    ulong mcause,
    ulong addr,
    ulong tval2,
    ulong tinst,
    struct sbi\_trap\_regs * regs,
    struct sbi\_scratch * scratch )
```

Here is the call graph for this function:



Here is the caller graph for this function:



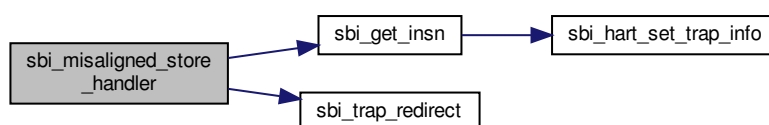
21.71.1.2 sbi_misaligned_store_handler()

```

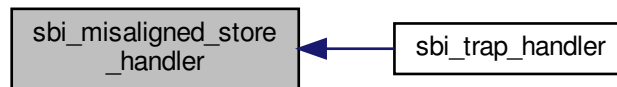
int sbi_misaligned_store_handler (
    u32 hartid,
    ulong mcause,
    ulong addr,
    ulong tval2,
    ulong tinst,
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch )

```

Here is the call graph for this function:



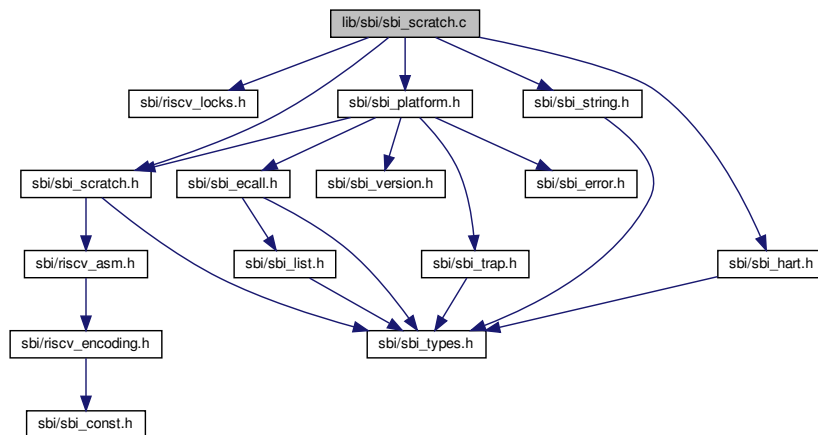
Here is the caller graph for this function:



21.72 lib/sbi/sbi_scratch.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for `sbi_scratch.c`:



Functions

- unsigned long `sbi_scratch_alloc_offset` (unsigned long size, const char *owner)
- void `sbi_scratch_free_offset` (unsigned long offset)

Variables

- static `spinlock_t` `extra_lock` = `SPIN_LOCK_INITIALIZER`
- static unsigned long `extra_offset` = `SBI_SCRATCH_EXTRA_SPACE_OFFSET`

21.72.1 Function Documentation

21.72.1.1 sbi_scratch_alloc_offset()

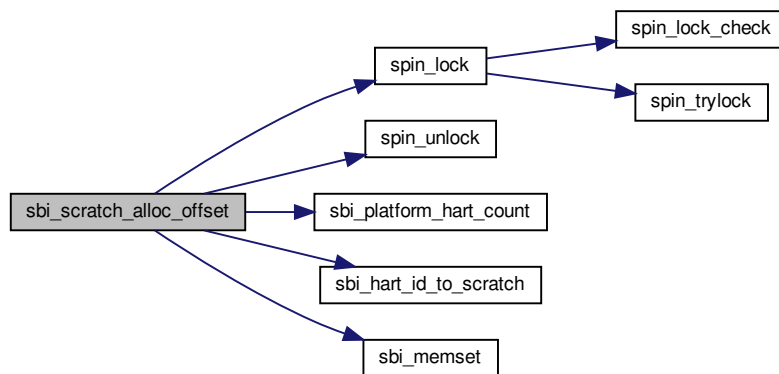
```
unsigned long sbi_scratch_alloc_offset (
    unsigned long size,
    const char * owner )
```

Allocate from extra space in [sbi_scratch](#)

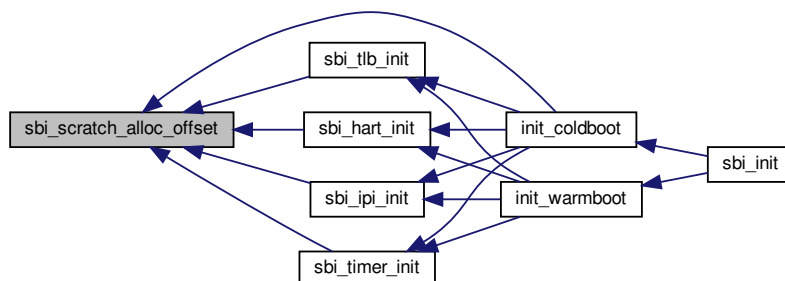
Returns

zero on failure and non-zero (\geq SBI_SCRATCH_EXTRA_SPACE_OFFSET) on success

Here is the call graph for this function:



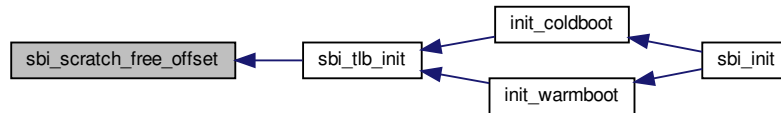
Here is the caller graph for this function:



21.72.1.2 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:



21.72.2 Variable Documentation

21.72.2.1 extra_lock

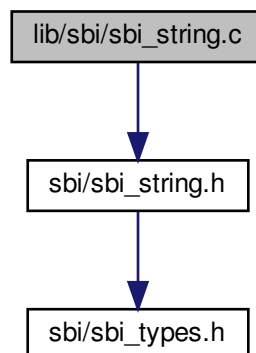
```
spinlock_t extra_lock = SPIN_LOCK_INITIALIZER [static]
```

21.72.2.2 extra_offset

```
unsigned long extra_offset = SBI_SCRATCH_EXTRA_SPACE_OFFSET [static]
```

21.73 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)

21.73.1 Function Documentation

21.73.1.1 [sbi_memchr\(\)](#)

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

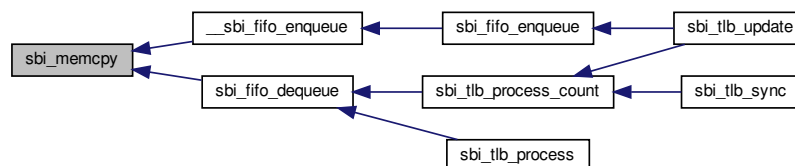
21.73.1.2 [sbi_memcmp\(\)](#)

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

21.73.1.3 [sbi_memcpy\(\)](#)

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

Here is the caller graph for this function:



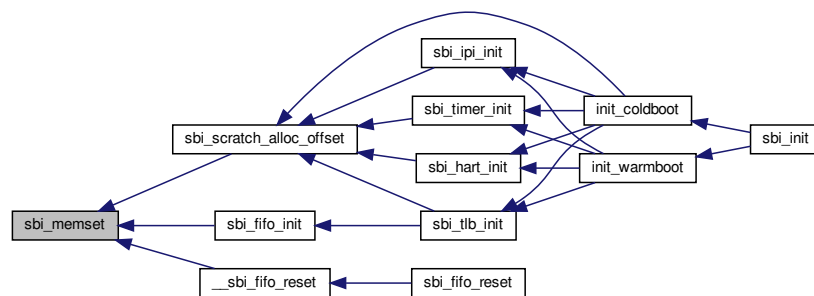
21.73.1.4 sbi_memmove()

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

21.73.1.5 sbi_memset()

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

Here is the caller graph for this function:



21.73.1.6 sbi_strchr()

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

21.73.1.7 sbi_strcmp()

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


21.73.1.8 sbi_strcpy()

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

21.73.1.9 sbi_strlen()

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

Here is the caller graph for this function:



21.73.1.10 sbi_strncpy()

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

21.73.1.11 sbi_strnlen()

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

21.73.1.12 sbi_strchr()

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

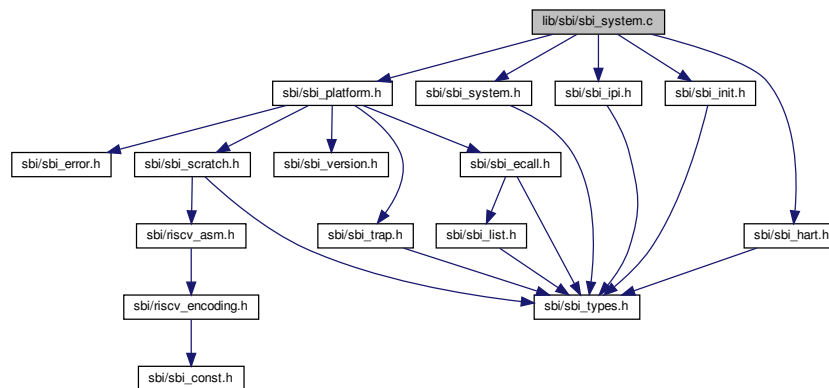
Here is the call graph for this function:



21.74 lib/sbi/sbi_system.c File Reference

```
#include <sbi/sbi_hart.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

- int [sbi_system_early_init](#) (struct [sbi_scratch](#) *scratch, bool cold_boot)
- int [sbi_system_final_init](#) (struct [sbi_scratch](#) *scratch, bool cold_boot)
- void [sbi_system_early_exit](#) (struct [sbi_scratch](#) *scratch)
- void [sbi_system_final_exit](#) (struct [sbi_scratch](#) *scratch)
- void [__noreturn sbi_system_reboot](#) (struct [sbi_scratch](#) *scratch, u32 type)
- void [__noreturn sbi_system_shutdown](#) (struct [sbi_scratch](#) *scratch, u32 type)

21.74.1 Function Documentation

21.74.1.1 sbi_system_early_exit()

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

Here is the call graph for this function:



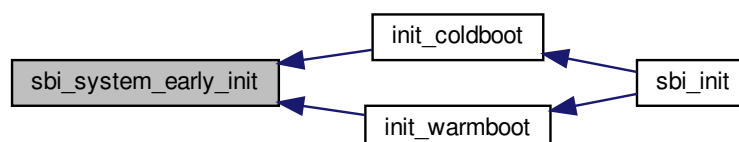
21.74.1.2 sbi_system_early_init()

```
int sbi_system_early_init (
    struct sbi_scratch * scratch,
    bool cold_boot )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.74.1.3 sbi_system_final_exit()

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

Here is the call graph for this function:



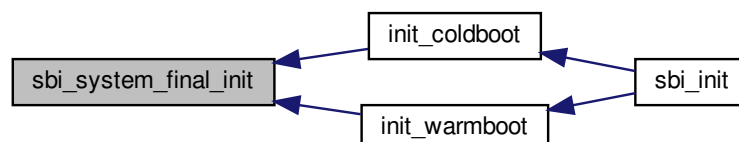
21.74.1.4 sbi_system_final_init()

```
int sbi_system_final_init (
    struct sbi_scratch * scratch,
    bool cold_boot )
```

Here is the call graph for this function:



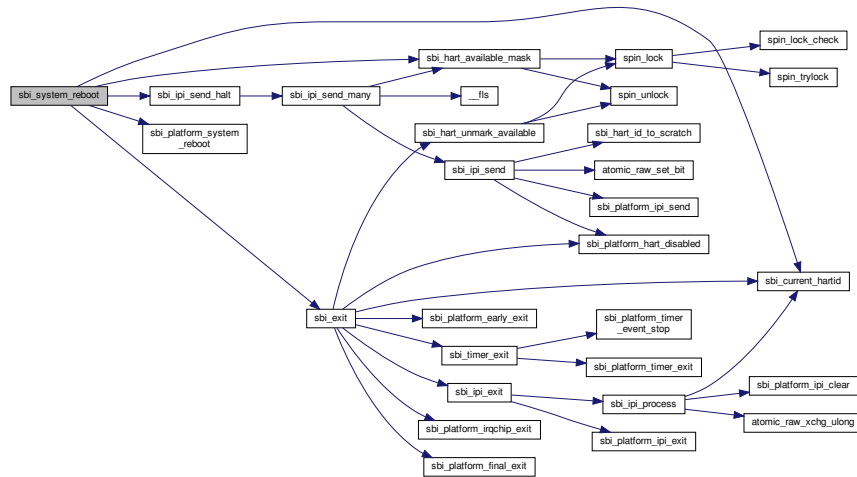
Here is the caller graph for this function:



21.74.1.5 sbi_system_reboot()

```
void __noreturn sbi_system_reboot (
    struct sbi_scratch * scratch,
    u32 type )
```

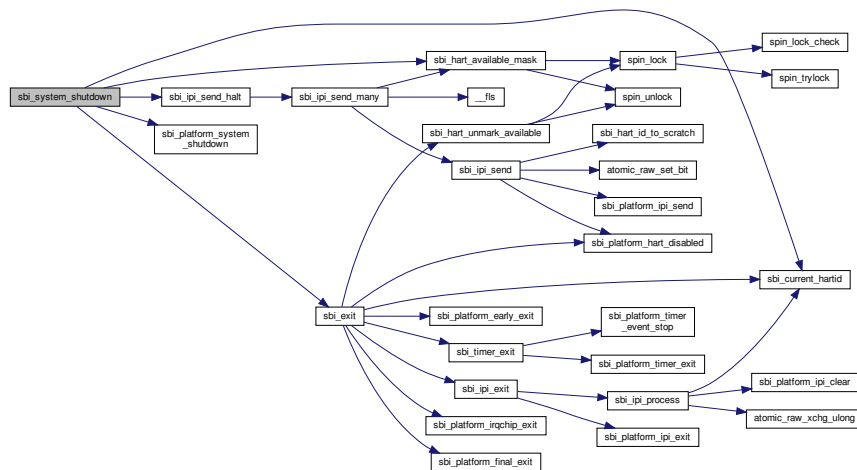
Here is the call graph for this function:



21.74.1.6 sbi_system_shutdown()

```
void __noreturn sbi_system_shutdown (
    struct sbi_scratch * scratch,
    u32 type )
```

Here is the call graph for this function:



Here is the caller graph for this function:



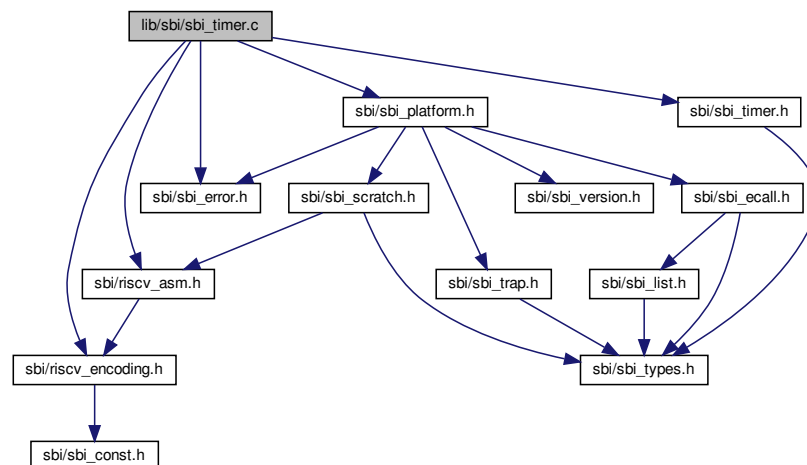
21.75 lib/sbi/sbi_timer.c File Reference

```

#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_timer.h>

```

Include dependency graph for sbi_timer.c:



Functions

- u64 [get_ticks](#) (void)
- u64 [sbi_timer_value](#) (struct [sbi_scratch](#) *scratch)
- u64 [sbi_timer_virt_value](#) (struct [sbi_scratch](#) *scratch)
- u64 [sbi_timer_get_delta](#) (struct [sbi_scratch](#) *scratch)
- void [sbi_timer_set_delta](#) (struct [sbi_scratch](#) *scratch, [ulong](#) delta)
- void [sbi_timer_set_delta_upper](#) (struct [sbi_scratch](#) *scratch, [ulong](#) delta_upper)
- void [sbi_timer_event_start](#) (struct [sbi_scratch](#) *scratch, u64 next_event)
- void [sbi_timer_process](#) (struct [sbi_scratch](#) *scratch)
- int [sbi_timer_init](#) (struct [sbi_scratch](#) *scratch, [bool](#) cold_boot)
- void [sbi_timer_exit](#) (struct [sbi_scratch](#) *scratch)

Variables

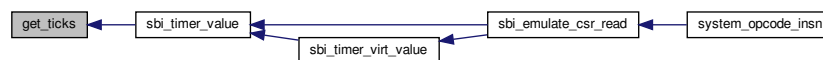
- static unsigned long [time_delta_off](#)

21.75.1 Function Documentation

21.75.1.1 `get_ticks()`

```
u64 get_ticks (
    void )
```

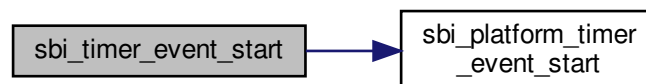
Here is the caller graph for this function:



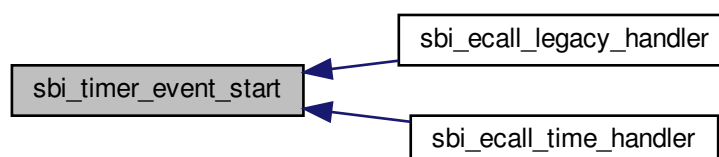
21.75.1.2 `sbi_timer_event_start()`

```
void sbi_timer_event_start (
    struct sbi\_scratch * scratch,
    u64 next_event )
```

Here is the call graph for this function:



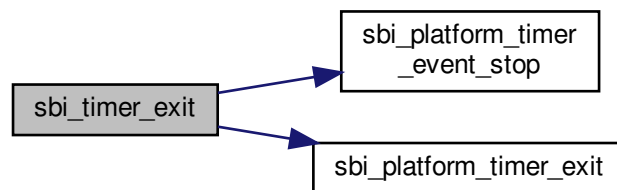
Here is the caller graph for this function:



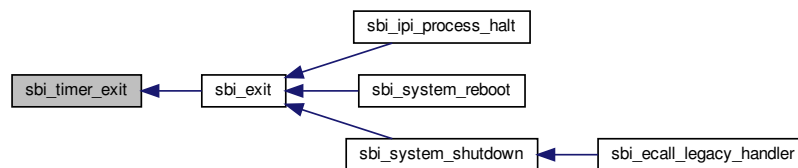
21.75.1.3 sbi_timer_exit()

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

Here is the call graph for this function:



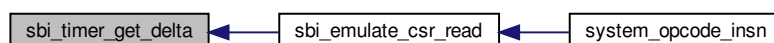
Here is the caller graph for this function:



21.75.1.4 sbi_timer_get_delta()

```
u64 sbi_timer_get_delta (
    struct sbi_scratch * scratch )
```

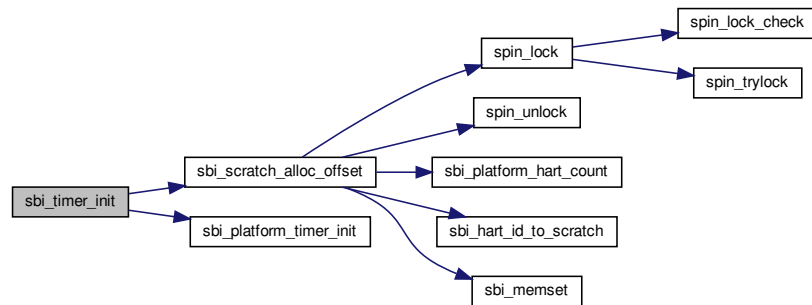
Here is the caller graph for this function:



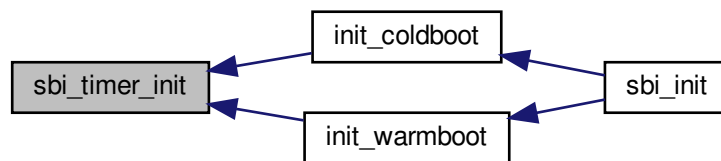
21.75.1.5 sbi_timer_init()

```
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:



21.75.1.6 sbi_timer_process()

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

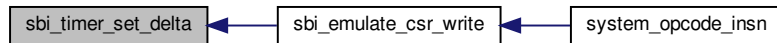
Here is the caller graph for this function:



21.75.1.7 sbi_timer_set_delta()

```
void sbi_timer_set_delta (
    struct sbi_scratch * scratch,
    ulong delta )
```

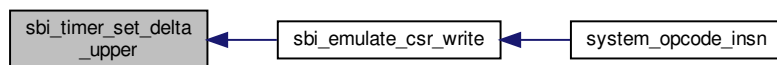
Here is the caller graph for this function:



21.75.1.8 sbi_timer_set_delta_upper()

```
void sbi_timer_set_delta_upper (
    struct sbi_scratch * scratch,
    ulong delta_upper )
```

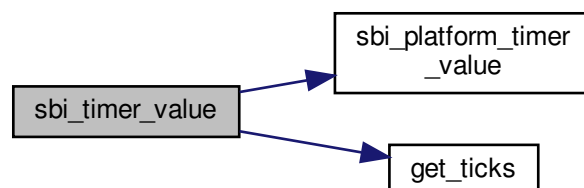
Here is the caller graph for this function:



21.75.1.9 sbi_timer_value()

```
u64 sbi_timer_value (
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:



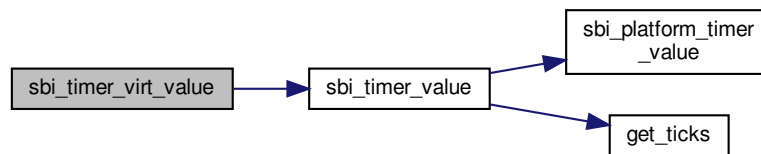
Here is the caller graph for this function:



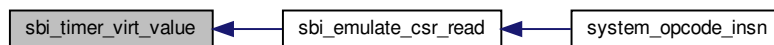
21.75.1.10 sbi_timer_virt_value()

```
u64 sbi_timer_virt_value (
    struct sbi_scratch * scratch )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.75.2 Variable Documentation

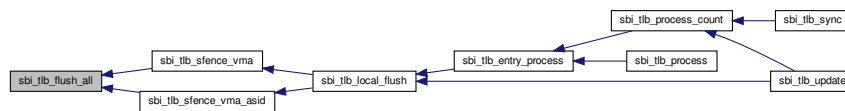
21.75.2.1 time_delta_off

```
unsigned long time_delta_off [static]
```


21.76.1.3 sbi_tlb_flush_all()

```
static void sbi_tlb_flush_all (
    void ) [static]
```

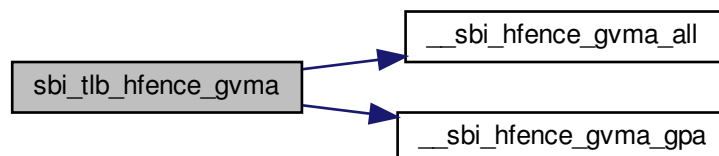
Here is the caller graph for this function:



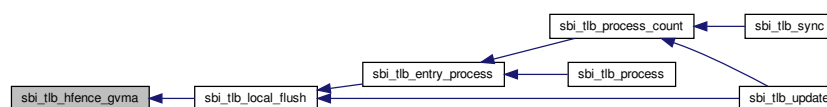
21.76.1.4 sbi_tlb_hfence_gvma()

```
static void sbi_tlb_hfence_gvma (
    struct sbi_tlb_info * tinfo ) [static]
```

Here is the call graph for this function:



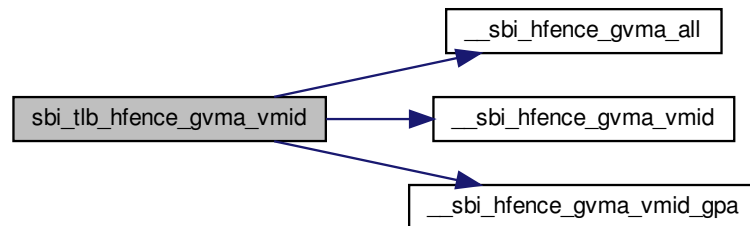
Here is the caller graph for this function:



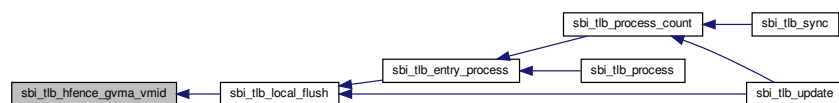
21.76.1.5 sbi_tlb_hfence_gvma_vmid()

```
static void sbi_tlb_hfence_gvma_vmid (
    struct sbi_tlb_info * tinfo ) [static]
```

Here is the call graph for this function:



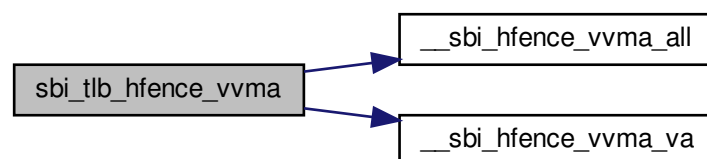
Here is the caller graph for this function:



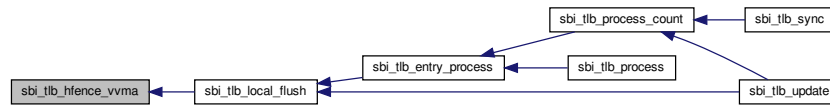
21.76.1.6 sbi_tlb_hfence_vvma()

```
static void sbi_tlb_hfence_vvma (
    struct sbi_tlb_info * tinfo ) [static]
```

Here is the call graph for this function:



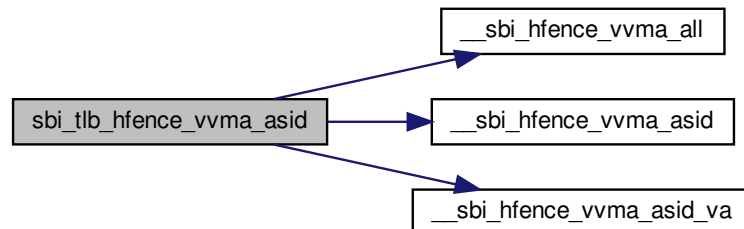
Here is the caller graph for this function:



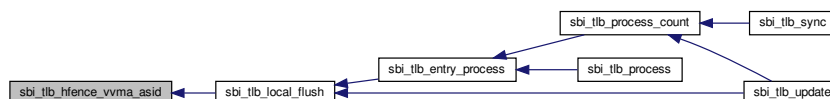
21.76.1.7 `sbi_tlb_hfence_vvma_asid()`

```
static void sbi_tlb_hfence_vvma_asid (
    struct sbi_tlb_info * tinfo ) [static]
```

Here is the call graph for this function:



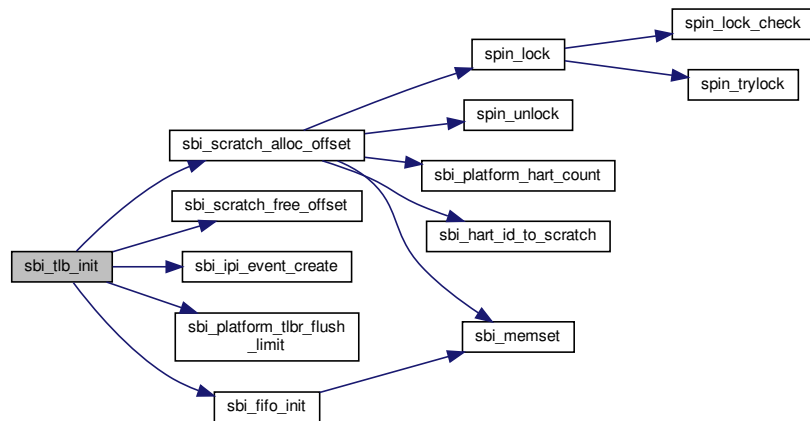
Here is the caller graph for this function:



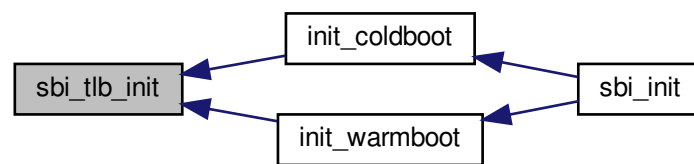
21.76.1.8 sbi_tlb_init()

```
int sbi_tlb_init (
    struct sbi_scratch * scratch,
    bool cold_boot )
```

Here is the call graph for this function:



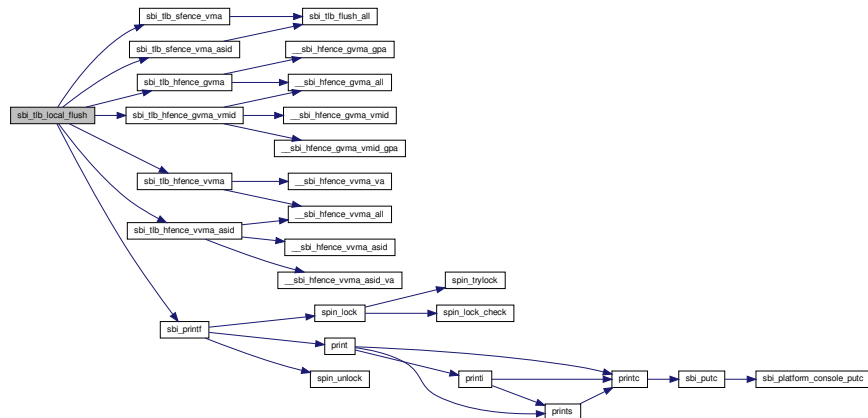
Here is the caller graph for this function:



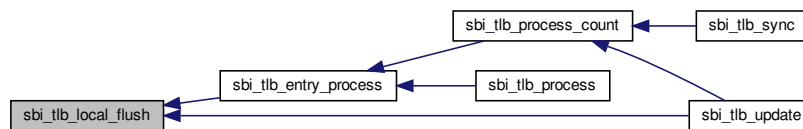
21.76.1.9 sbi_tlb_local_flush()

```
static void sbi_tlb_local_flush (
    struct sbi_tlb_info * tinfo ) [static]
```

Here is the call graph for this function:



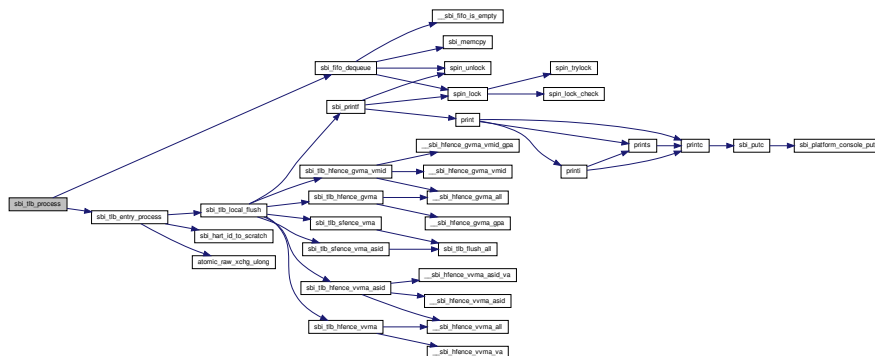
Here is the caller graph for this function:



21.76.1.10 sbi_tlb_process()

```
static void sbi_tlb_process (
    struct sbi\_scratch * scratch ) [static]
```

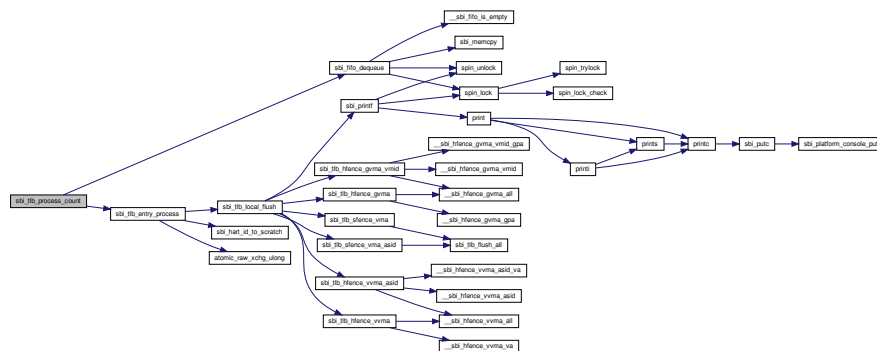
Here is the call graph for this function:



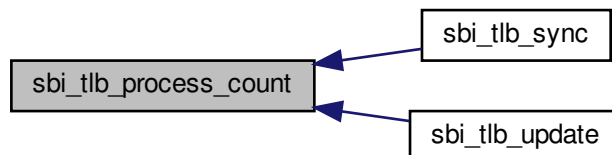
21.76.1.11 `sbi_tlb_process_count()`

```
static void sbi_tlb_process_count (
    struct sbi_scratch * scratch,
    int count ) [static]
```

Here is the call graph for this function:



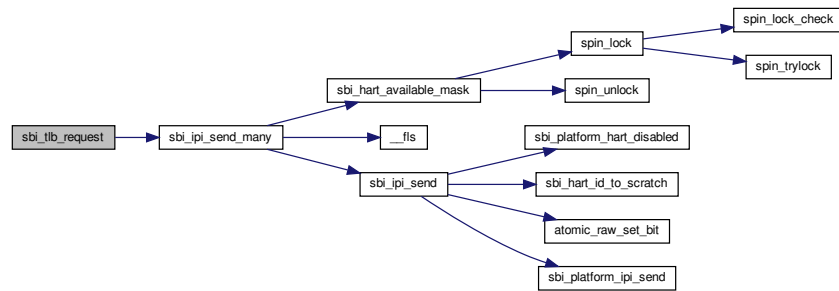
Here is the caller graph for this function:



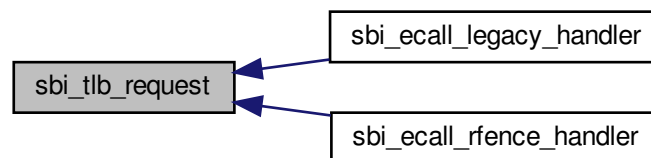
21.76.1.12 `sbi_tlb_request()`

```
int sbi_tlb_request (
    struct sbi_scratch * scratch,
    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:



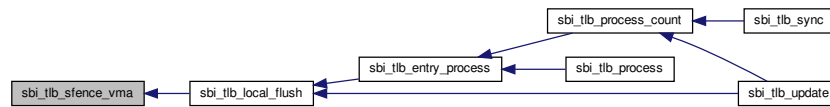
21.76.1.13 sbi_tlb_sfence_vma()

```
static void sbi_tlb_sfence_vma (
    struct sbi_tlb_info * tinfo ) [static]
```

Here is the call graph for this function:



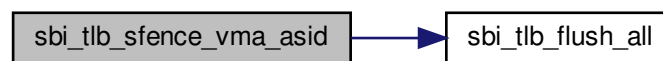
Here is the caller graph for this function:



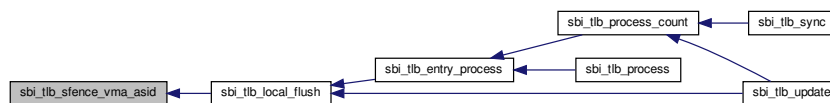
21.76.1.14 sbi_tlb_sfence_vma_asid()

```
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:



21.76.1.15 sbi_tlb_sync()

```
static void sbi_tlb_sync (
    struct sbi_scratch * scratch ) [static]
```


Call back to decide if an inplace fifo update is required or next entry 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:



21.76.2 Variable Documentation

21.76.2.1 tlb_event

```
u32 tlb_event = SBI_IPI_EVENT_MAX [static]
```

21.76.2.2 tlb_fifo_mem_off

```
unsigned long tlb_fifo_mem_off [static]
```

21.76.2.3 tlb_fifo_off

```
unsigned long tlb_fifo_off [static]
```

21.76.2.4 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,
}
```

21.76.2.5 tlb_range_flush_limit

```
unsigned long tlb_range_flush_limit [static]
```

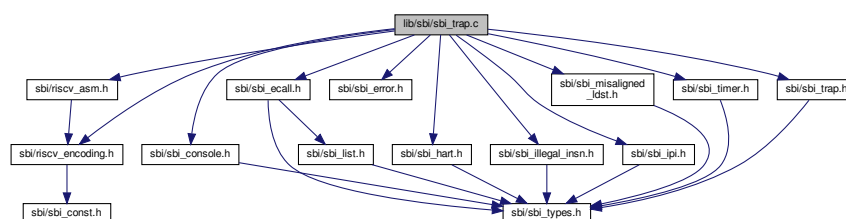
21.76.2.6 tlb_sync_off

```
unsigned long tlb_sync_off [static]
```

21.77 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

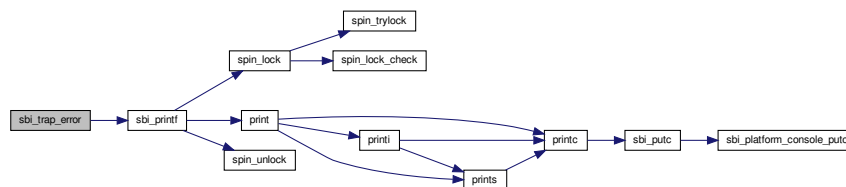
- static void `__noreturn sbi_trap_error` (const char *msg, int rc, u32 hartid, ulong mcause, ulong mtval, ulong mtval2, ulong mtinst, struct `sbi_trap_regs` *regs)
- int `sbi_trap_redirect` (struct `sbi_trap_regs` *regs, struct `sbi_trap_info` *trap, struct `sbi_scratch` *scratch)
- void `sbi_trap_handler` (struct `sbi_trap_regs` *regs, struct `sbi_scratch` *scratch)

21.77.1 Function Documentation

21.77.1.1 `sbi_trap_error()`

```
static void __noreturn sbi_trap_error (
    const char * msg,
    int rc,
    u32 hartid,
    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:



21.77.1.2 sbi_trap_handler()

```
void sbi_trap_handler (
    struct sbi_trap_regs * regs,
    struct sbi_scratch * scratch )
```

Handle trap/interrupt

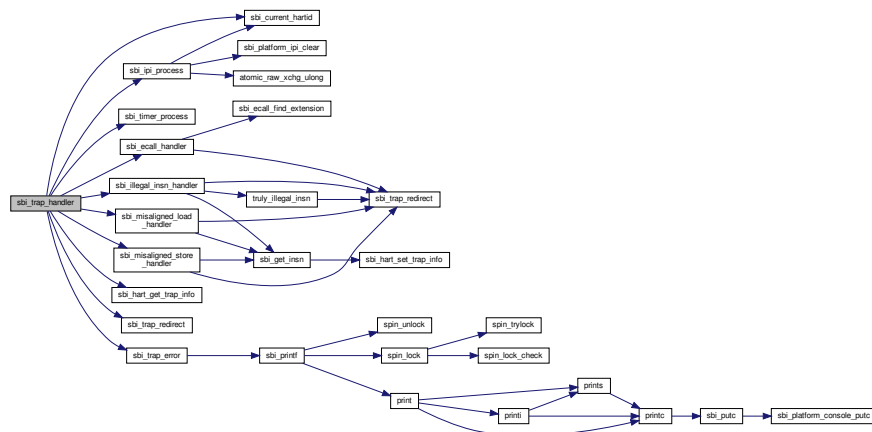
This function is called by firmware linked to OpenSBI library for handling trap/interrupt. It expects the following:

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

<i>regs</i>	pointer to register state
<i>scratch</i>	pointer to <code>sbi_scratch</code> of current HART

Here is the call graph for this function:



21.77.1.3 sbi_trap_redirect()

```
int sbi_trap_redirect (
    struct sbi_trap_regs * regs,
```

```

struct sbi_trap_info * trap,
struct sbi_scratch * scratch )

```

Redirect trap to lower privilege mode (S-mode or U-mode)

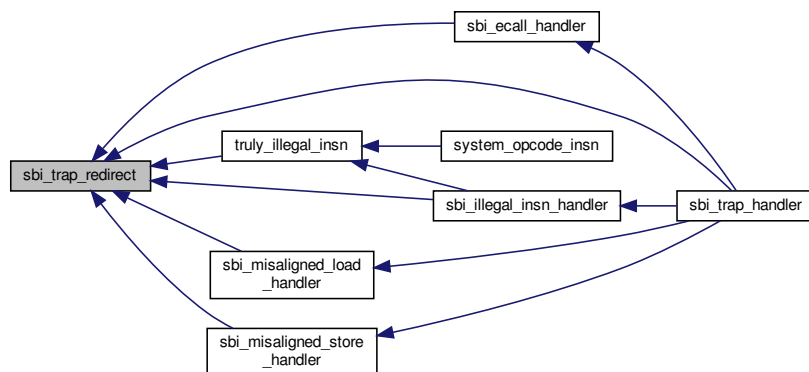
Parameters

<i>regs</i>	pointer to register state
<i>trap</i>	pointer to trap details
<i>scratch</i>	pointer to sbi_scratch of current HART

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



21.78 lib/sbi/sbi_unpriv.c File Reference

```

#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bits.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>

```



```

sbi_hart_set_trap_info(scratch, trap);
asm volatile(
    "csrrs %0, " STR(CSR_MSTATUS) ", %3\n"
    ".option push\n"
    ".option norvc\n"
    #insn " %1, %2\n"
    ".option pop\n"
    "csrw " STR(CSR_MSTATUS) ", %0"
    : "+&r" (__mstatus), "=&r"(val)
    : "m"(*addr), "r"(MSTATUS_MPRV));
sbi_hart_set_trap_info(scratch, NULL);
return val;
}

```

21.78.1.2 DEFINE_UNPRIVILEGED_STORE_FUNCTION

```

#define DEFINE_UNPRIVILEGED_STORE_FUNCTION(
    type,
    insn )

```

Value:

```

void sbi_store_##type(type *addr, type val,
    struct sbi_scratch *scratch,
    struct sbi_trap_info *trap)
{
    register ulong __mstatus asm("a3");
    trap->epc = 0;
    trap->cause = 0;
    trap->tval = 0;
    trap->tval2 = 0;
    trap->tinst = 0;
    sbi_hart_set_trap_info(scratch, trap);
    asm volatile(
        "csrrs %0, " STR(CSR_MSTATUS) ", %3\n"
        ".option push\n"
        ".option norvc\n"
        #insn " %1, %2\n"
        ".option pop\n"
        "csrw " STR(CSR_MSTATUS) ", %0"
        : "+&r" (__mstatus)
        : "r"(val), "m"(*addr), "r"(MSTATUS_MPRV));
    sbi_hart_set_trap_info(scratch, NULL);
}

```

21.78.2 Function Documentation

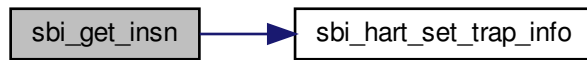
21.78.2.1 sbi_get_insn()

```

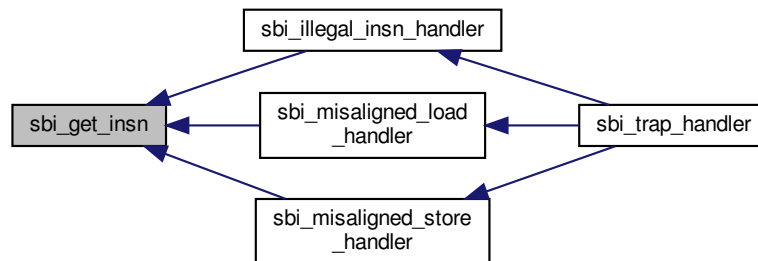
ulong sbi_get_insn (
    ulong mepc,
    struct sbi_scratch * scratch,
    struct sbi_trap_info * trap )

```

Here is the call graph for this function:



Here is the caller graph for this function:



21.78.2.2 sbi_load_u64()

```

u64 sbi_load_u64 (
    const u64 * addr,
    struct sbi_scratch * scratch,
    struct sbi_trap_info * trap )
  
```

21.78.2.3 sbi_store_u64()

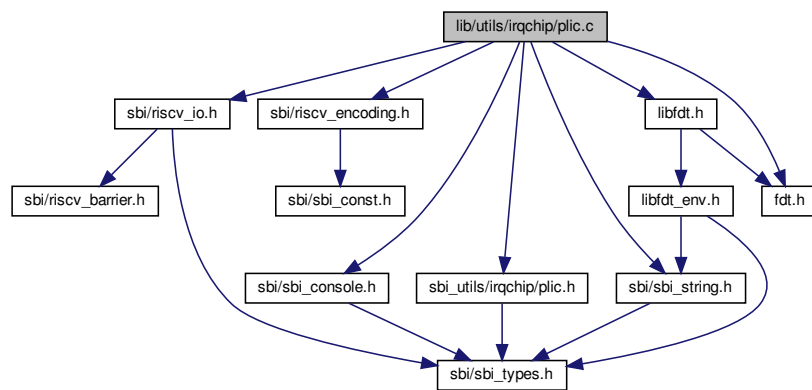
```

void sbi_store_u64 (
    u64 * addr,
    u64 val,
    struct sbi_scratch * scratch,
    struct sbi_trap_info * trap )
  
```

21.79 lib/utls/irqchip/plic.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_string.h>
#include <sbi_utls/irqchip/plic.h>
#include <libfdt.h>
#include <fdt.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 `plc_set_priority` (u32 source, u32 val)
- void `plc_set_thresh` (u32 cntxid, u32 val)
- void `plc_set_ie` (u32 cntxid, u32 word_index, u32 val)
- void `plc_fdt_fixup` (void *fdt, const char *compat)
- int `plc_warm_irqchip_init` (u32 target_hart, int m_cntx_id, int s_cntx_id)
- int `plc_cold_irqchip_init` (unsigned long base, u32 num_sources, u32 hart_count)

Variables

- static u32 `plc_hart_count`
- static u32 `plc_num_sources`
- static volatile void * `plc_base`

21.79.1 Macro Definition Documentation

21.79.1.1 PLIC_CONTEXT_BASE

```
#define PLIC_CONTEXT_BASE 0x200000
```

21.79.1.2 PLIC_CONTEXT_STRIDE

```
#define PLIC_CONTEXT_STRIDE 0x1000
```

21.79.1.3 PLIC_ENABLE_BASE

```
#define PLIC_ENABLE_BASE 0x2000
```

21.79.1.4 PLIC_ENABLE_STRIDE

```
#define PLIC_ENABLE_STRIDE 0x80
```

21.79.1.5 PLIC_PENDING_BASE

```
#define PLIC_PENDING_BASE 0x1000
```

21.79.1.6 PLIC_PRIORITY_BASE

```
#define PLIC_PRIORITY_BASE 0x0
```

21.79.2 Function Documentation


```
int plic_cold_irqchip_init (
    unsigned long base,
    u32 num_sources,
    u32 hart_count )
```

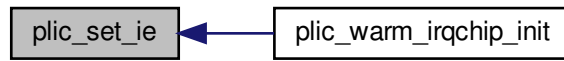
```
graph LR; A[plic_cold_irqchip_init] --> B[plic_set_priority]
```

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

[illegible]

```
void plic_set_ie (
    u32 cntxid,
    u32 word_index,
    u32 val )
```

Here is the caller graph for this function:



21.79.2.4 plic_set_priority()

```
static void plic_set_priority (  
    u32 source,  
    u32 val ) [static]
```

Here is the caller graph for this function:



21.79.2.5 plic_set_thresh()

```
void plic_set_thresh (  
    u32 cntxid,  
    u32 val )
```

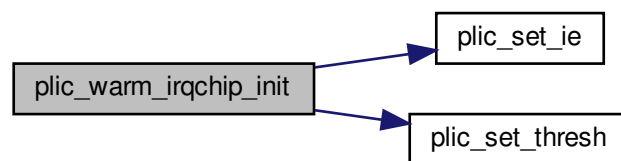
Here is the caller graph for this function:



21.79.2.6 plic_warm_irqchip_init()

```
int plic_warm_irqchip_init (
    u32 target_hart,
    int m_cntx_id,
    int s_cntx_id )
```

Here is the call graph for this function:



21.79.3 Variable Documentation

21.79.3.1 plic_base

```
volatile void* plic_base [static]
```

21.79.3.2 plic_hart_count

```
u32 plic_hart_count [static]
```

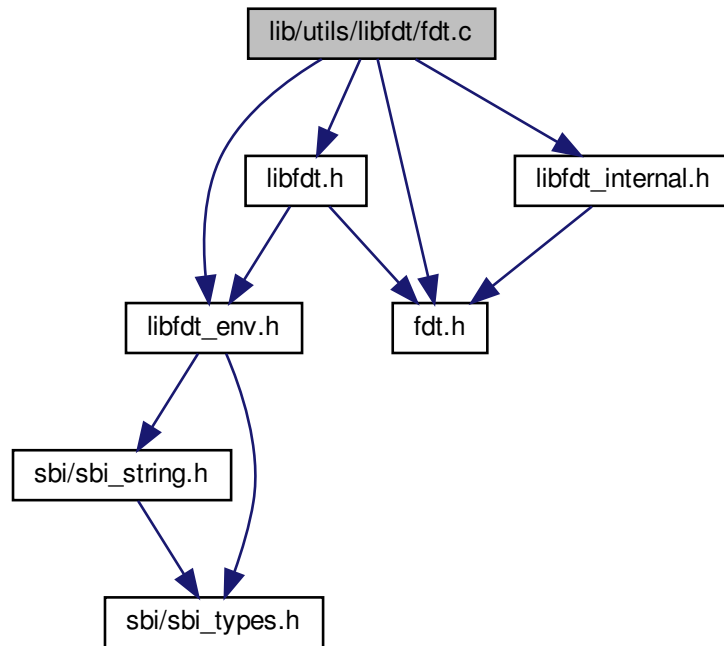
21.79.3.3 plic_num_sources

```
u32 plic_num_sources [static]
```

21.80 lib/utls/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

- `int 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 bufsz`)

21.80.1 Function Documentation

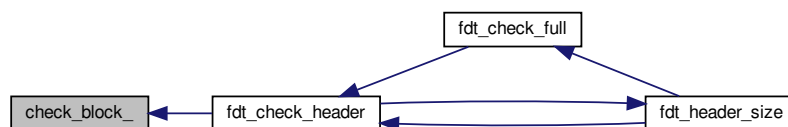
21.80.1.1 check_block_()

```
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:



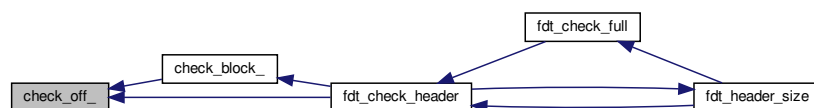
Here is the caller graph for this function:



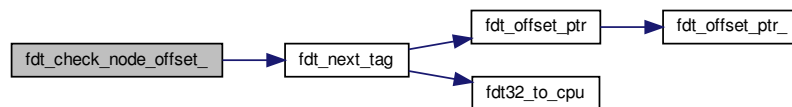
21.80.1.2 check_off_()

```
static int check_off_ (  
    uint32_t hdrsize,  
    uint32_t totalsize,  
    uint32_t off ) [static]
```

Here is the caller graph for this function:

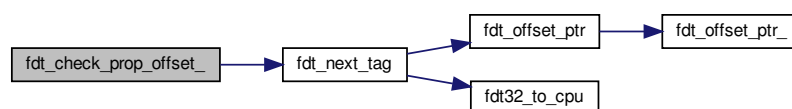


Here is the caller graph for this function:



21.80.1.5 fdt_check_prop_offset()

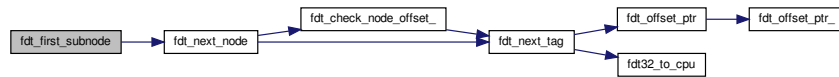
Here is the call graph for this function:



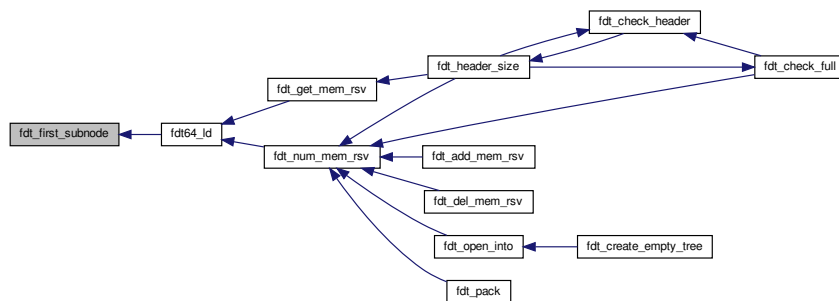
Returns

offset of first subnode, or -FDT_ERR_NOTFOUND if there is none

Here is the call graph for this function:



Here is the caller graph for this function:

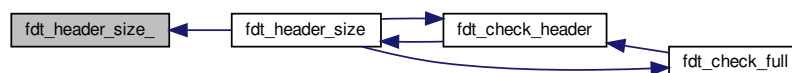


21.80.1.8 fdt_header_size_()

```

size_t fdt_header_size_ (
    uint32_t version )
  
```

fdt_header_size - return the size of the tree's header : pointer to a flattened device tree Here is the caller graph for this function:



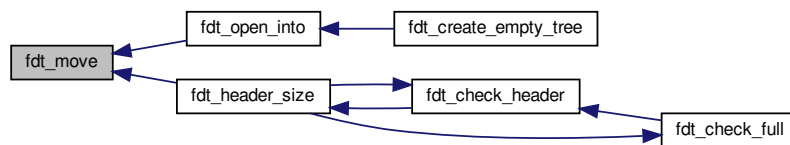
21.80.1.9 fdt_move()

```
int fdt_move (
    const void * fdt,
    void * buf,
    int bufsize )
```

fdt_move - move a device tree around in memory : pointer to the device tree to move : pointer to memory where the device is to be moved : size of the memory space at buf

fdt_move() relocates, if possible, the device tree blob located at *fdt* to the buffer at *buf* of size *bufsize*. The buffer may overlap with the existing device tree blob at *fdt*. Therefore, `fdt_move(fdt, fdt, fdt_totalsize(fdt))` should always succeed.

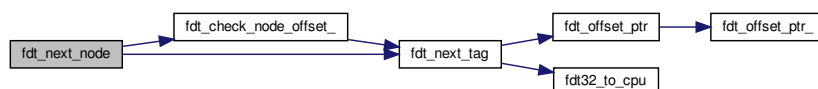
returns: 0, on success -FDT_ERR_NOSPACE, *bufsize* is insufficient to contain the device tree -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the caller graph for this function:



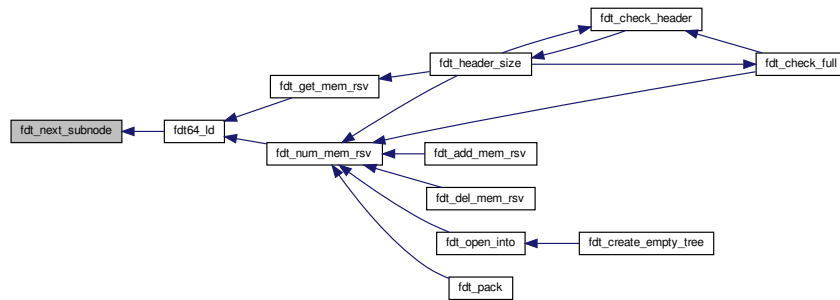
21.80.1.10 fdt_next_node()

```
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:

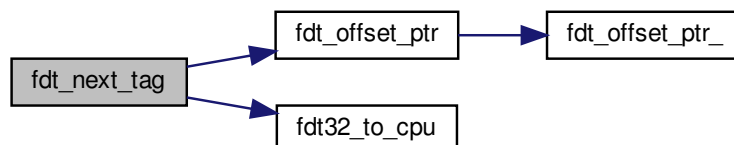


21.80.1.12 fdt_next_tag()

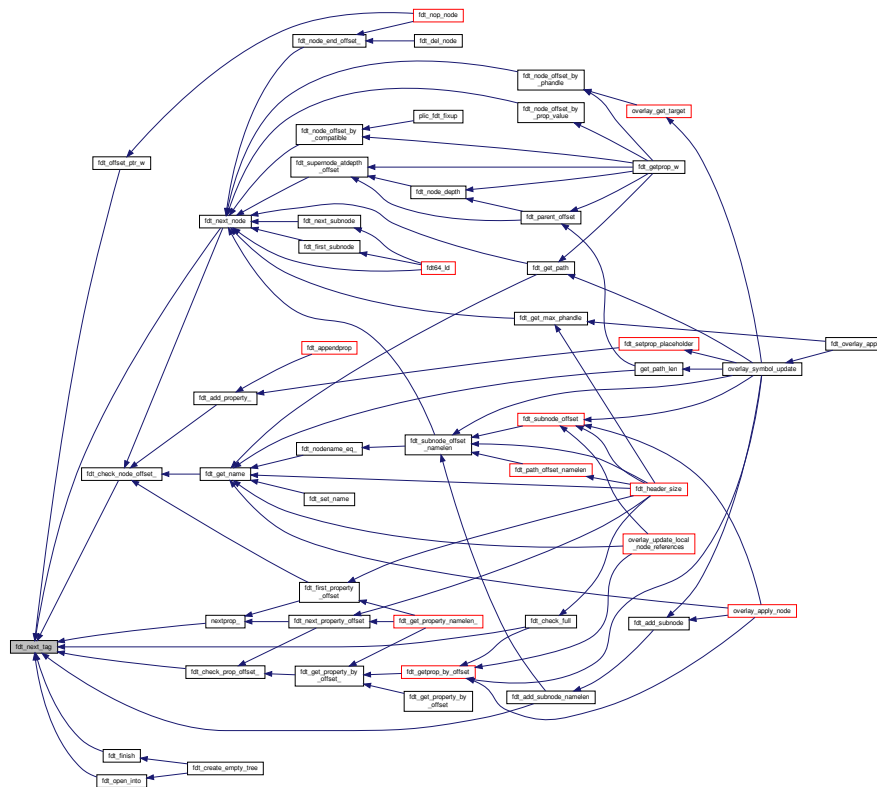
```

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:



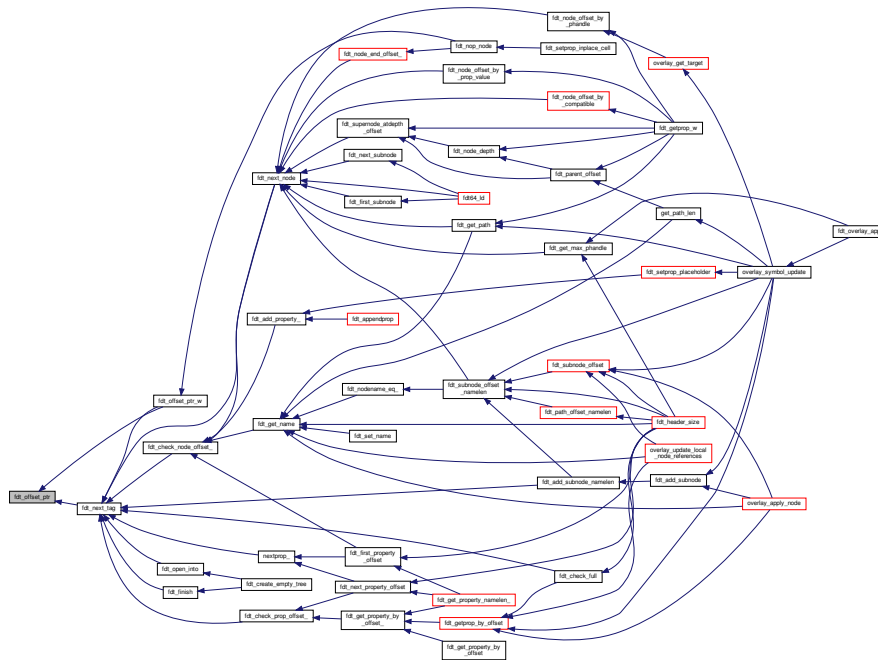
21.80.1.13 fdt_offset_ptr()

```
const void* fdt_offset_ptr (
    const void * fdt,
    int offset,
    unsigned int len )
```

Here is the call graph for this function:



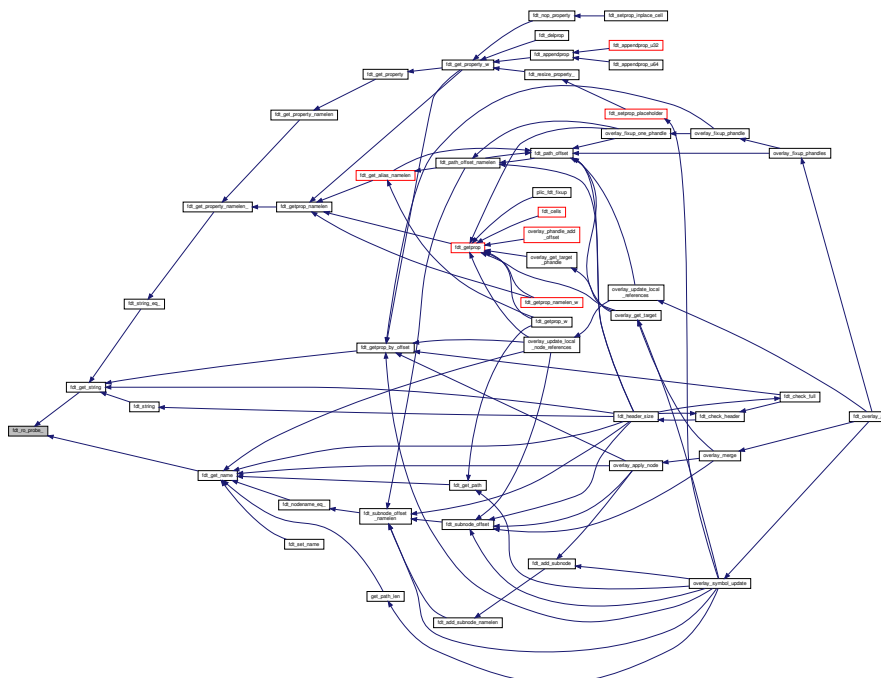
Here is the caller graph for this function:



21.80.1.14 fdt_ro_probe_()

```
int fdt_ro_probe_ (
    const void * fdt )
```

Here is the caller graph for this function:



21.81.1.3 FDT_END_NODE

```
#define FDT_END_NODE 0x2 /* End node */
```

21.81.1.4 FDT_MAGIC

```
#define FDT_MAGIC 0xd00dfeed /* 4: version, 4: total size */
```

21.81.1.5 FDT_NOP

```
#define FDT_NOP 0x4 /* nop */
```

21.81.1.6 FDT_PROP

```
#define FDT_PROP
```

Value:

```
0x3      /* Property: name off,  
          size, content */
```

21.81.1.7 FDT_TAGSIZE

```
#define FDT_TAGSIZE sizeof(fdt32_t)
```

21.81.1.8 FDT_V16_SIZE

```
#define FDT_V16_SIZE FDT_V3_SIZE
```

21.81.1.9 FDT_V17_SIZE

```
#define FDT_V17_SIZE (FDT_V16_SIZE + sizeof(fdt32_t))
```


21.81.1.10 FDT_V1_SIZE

```
#define FDT_V1_SIZE (7*sizeof(fdt32_t))
```

21.81.1.11 FDT_V2_SIZE

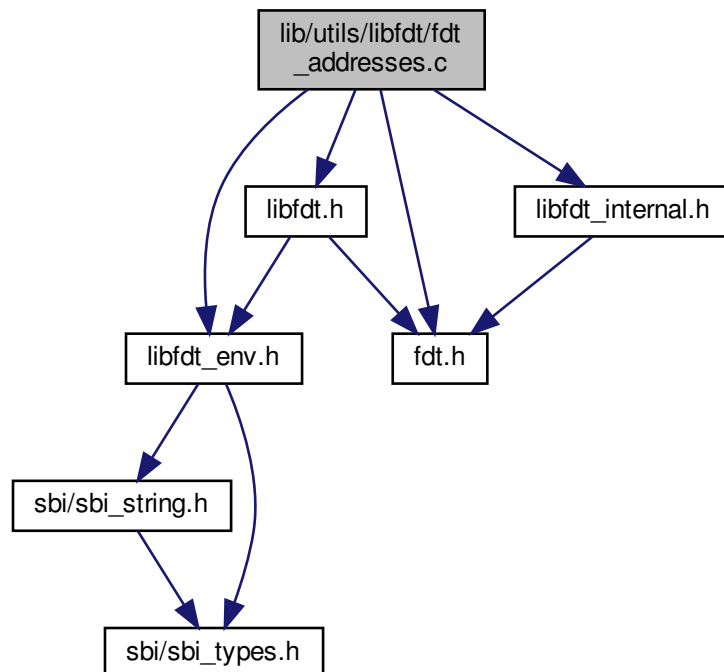
```
#define FDT_V2_SIZE (FDT_V1_SIZE + sizeof(fdt32_t))
```

21.81.1.12 FDT_V3_SIZE

```
#define FDT_V3_SIZE (FDT_V2_SIZE + sizeof(fdt32_t))
```

21.82 lib/utls/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)

21.82.1 Function Documentation

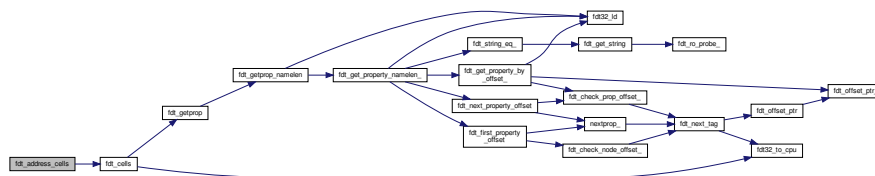
21.82.1.1 fdt_address_cells()

```
int fdt_address_cells (
    const void * fdt,
    int nodeoffset )
```

`fdt_address_cells` - retrieve address size for a bus represented in the tree : pointer to the device tree blob : offset of the node to find the address size for

When the node has a valid #address-cells property, returns its value.

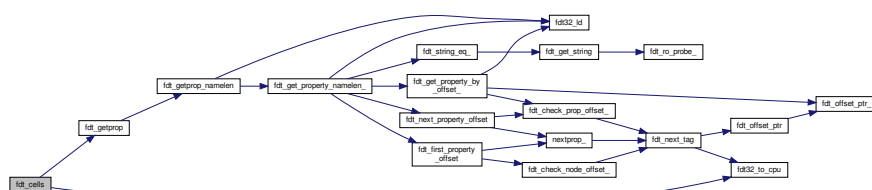
returns: $0 \leq n < \text{FDT_MAX_NCELLS}$, on success 2, if the node has no #address-cells property -FDT_ERR_BADNCELLS, if the node has a badly formatted or invalid #address-cells property -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



21.82.1.2 fdt_cells()

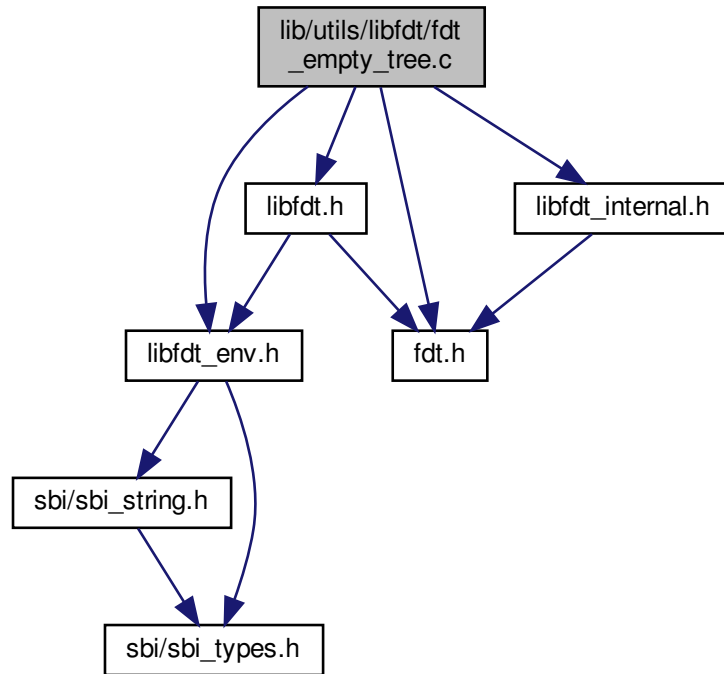
```
static int fdt_cells (
    const void * fdt,
    int nodeoffset,
    const char * name ) [static]
```

Here is the call graph for this function:




```
#include "libfdt_internal.h"
```

Include dependency graph for `fdt_empty_tree.c`:



Functions

- [int `fdt_create_empty_tree`](#) (void *buf, int bufsize)

21.83.1 Function Documentation

21.83.1.1 `fdt_create_empty_tree()`

```
int fdt_create_empty_tree (
    void * buf,
    int bufsize )
```


- 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 : Base device tree blob : Device tree overlay blob

: 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 : Base Device Tree blob : Device tree overlay blob : Node offset of the symbols node in the base device tree : Path to a node holding a phandle in the overlay : number of path characters to consider

: number of name characters to consider : Offset within the overlay property where the phandle is stored : 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)

21.84.1 Function Documentation

21.84.1.1 fdt_overlay_apply()

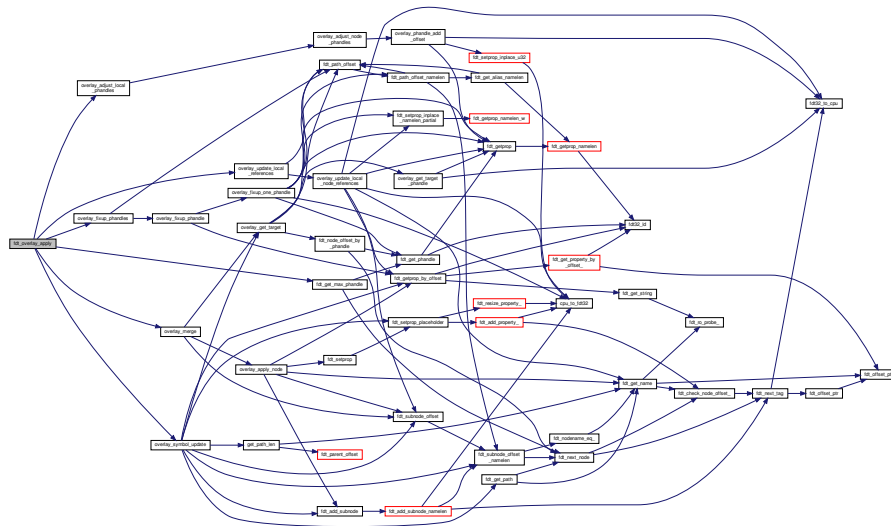
```
int fdt_overlay_apply (
    void * fdt,
    void * fdto )
```

fdt_overlay_apply - Applies a DT overlay on a base DT : pointer to the base device tree blob : 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.

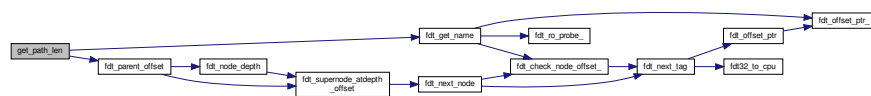
returns: 0, on success -FDT_ERR_NOSPACE, there's not enough space in the base device tree -FDT_ERR_↵
NOTFOUND, the overlay points to some inexistant nodes or properties in the base DT -FDT_ERR_BADPHAN↵
DLE, -FDT_ERR_BADOVERLAY, -FDT_ERR_NOPHANDLES, -FDT_ERR_INTERNAL, -FDT_ERR_BADLAYOUT,
-FDT_ERR_BADMAGIC, -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:



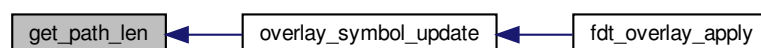
21.84.1.2 get_path_len()

```
static int get_path_len (
    const void * fdt,
    int nodeoffset ) [static]
```

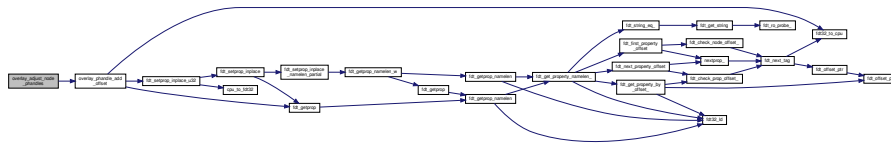
Here is the call graph for this function:



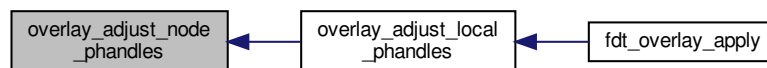
Here is the caller graph for this function:



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:



21.84.1.5 overlay_apply_node()

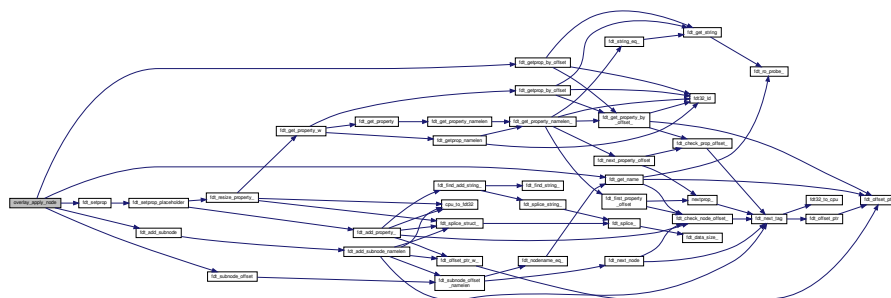
```
static int overlay_apply_node (
    void * fdt,
    int target,
    void * fdto,
    int node ) [static]
```

overlay_apply_node - Merges a node into the base device tree : Base Device Tree blob : Node offset in the base device tree to apply the fragment to : Device tree overlay blob : 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 handles 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:



```
graph RL; fdt_overlay_apply --> overlay_merge; overlay_merge --> overlay_apply_node;
```

The diagram illustrates the sequence of operations for applying an overlay. It consists of three rectangular boxes arranged horizontally from right to left. The rightmost box is labeled 'fdt_overlay_apply'. A blue arrow points from this box to the middle box, labeled 'overlay_merge'. Another blue arrow points from the middle box to the leftmost box, labeled 'overlay_apply_node'. The leftmost box has a gray background, while the other two are white.

Here is the caller graph for this function:



21.84.1.9 overlay_get_target()

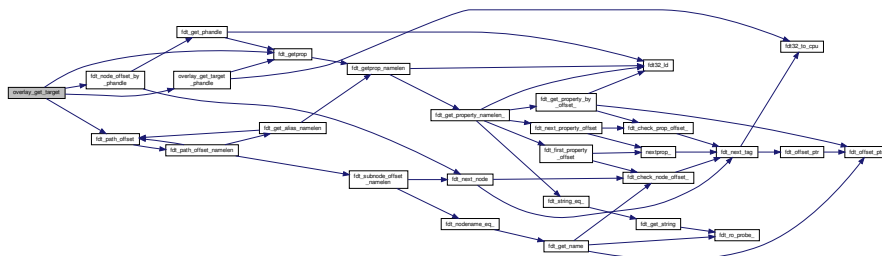
```

static int overlay_get_target (
    const void * fdt,
    const void * fdto,
    int fragment,
    char const ** pathp ) [static]
  
```

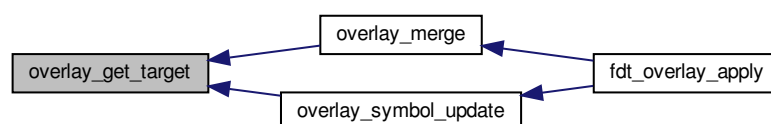
overlay_get_target - retrieves the offset of a fragment's target : Base device tree blob : Device tree overlay blob : node offset of the fragment in the overlay : pointer which receives the path of the target (or NULL)

[overlay_get_target\(\)](#) retrieves the target offset in the base device tree of a fragment, no matter how the actual targetting is done (through a phandle or a path)

returns: the targetted 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:



```
static uint32_t overlay_get_target_phandle (
    const void * fdto,
    int fragment ) [static]
```

`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).

```

graph LR
    Entry(( )) --> LdtGetPropName[ldt_get_prop_name]
    LdtGetPropName --> LdtGetPropByOffset[ldt_get_prop_by_offset]
    LdtGetPropByOffset --> LdtStringEq[ldt_string_eq]
    LdtStringEq --> LdtGetString[ldt_get_string]
    LdtGetString --> LdtNoProbe[ldt_no_probe]
    LdtGetPropByOffset --> LdtNextPropertyOffset[ldt_next_property_offset]
    LdtNextPropertyOffset --> LdtCheckPropOffset[ldt_check_prop_offset]
    LdtGetPropByOffset --> LdtFirstPropertyOffset[ldt_first_property_offset]
    LdtFirstPropertyOffset --> NextProp[ldt_next_prop]
    LdtFirstPropertyOffset --> LdtCheckNodeOffset[ldt_check_node_offset]
    LdtCheckPropOffset --> LdtOffsetPtr[ldt_offset_ptr]
    NextProp --> LdtNextTag[ldt_next_tag]
    LdtCheckNodeOffset --> LdtOffsetPtr
    LdtNextTag --> LdtOffsetPtr
    LdtCheckNodeOffset --> Ldt32ToCpu[ldt32_to_cpu]
    Ldt32ToCpu --> Exit(( ))
  
```

```

graph LR
    fdt_overlay_apply[fdt_overlay_apply] --> overlay_merge[overlay_merge]
    fdt_overlay_apply --> overlay_symbol_update[overlay_symbol_update]
    overlay_merge --> overlay_get_target[overlay_get_target]
    overlay_symbol_update --> overlay_get_target
    overlay_get_target --> overlay_get_target_phandle[overlay_get_target_phandle]

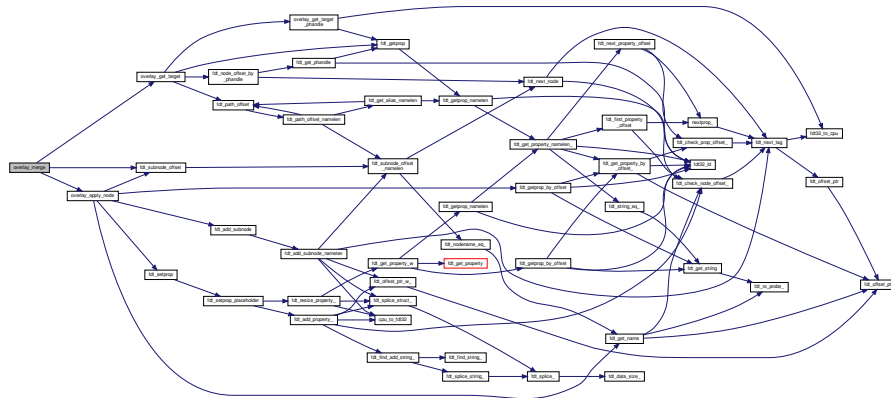
```

```
static int overlay_merge (
    void * fdt,
    void * fdto ) [static]
```

`overlay_merge()` merges an overlay into its base device tree.

Generated by Doxygen

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:

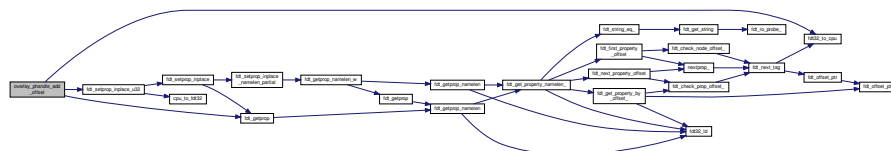


21.84.1.12 overlay_phandle_add_offset()

```

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:



21.84.1.13 overlay_symbol_update()

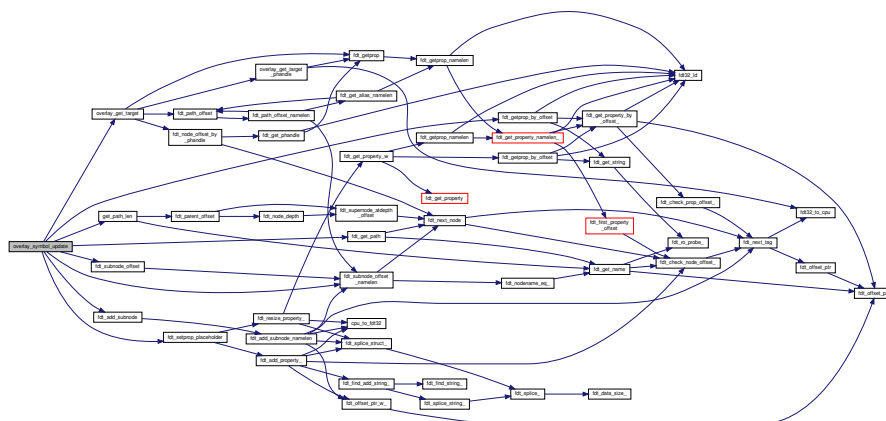
```
static int overlay_symbol_update (
    void * fdt,
    void * fdto ) [static]
```

overlay_symbol_update - Update the symbols of base tree after a merge : Base Device Tree blob : 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:

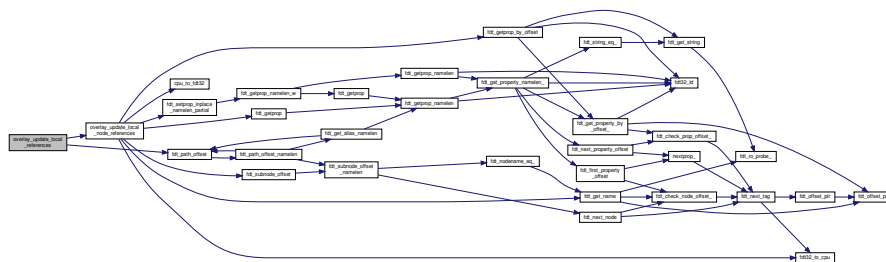


Here is the caller graph for this function:

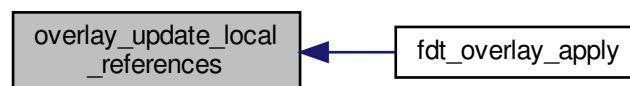


This is mainly used as part of a device tree application process, where you want the device tree overlays phandles to not conflict with the ones from the base device tree before merging them.

returns: 0 on success Negative error code on failure Here is the call graph for this function:



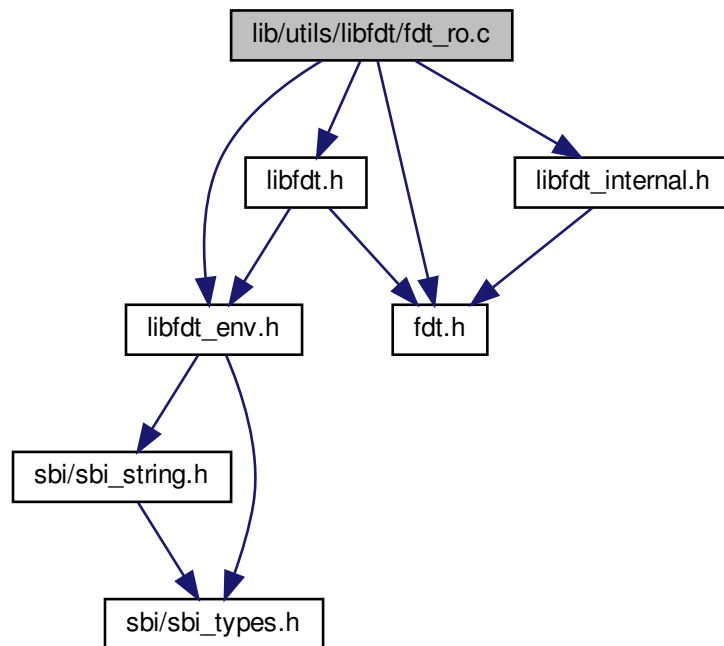
Here is the caller graph for this function:



21.85 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

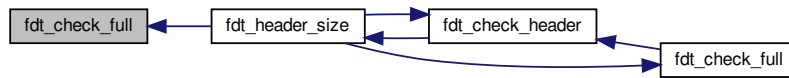
- static int `fdt_nodename_eq` (const void *fdt, int offset, const char *s, int len)
- const char * `fdt_get_string` (const void *fdt, int stroffset, int *lenp)
- const char * `fdt_string` (const void *fdt, int stroffset)
- static int `fdt_string_eq` (const void *fdt, int stroffset, const char *s, int len)
- uint32_t `fdt_get_max_phandle` (const void *fdt)
- 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 *lenp)
- 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)

- ### 21.85.1 Function Documentation

```
int fdt_check_full (
    const void * fdt,
    size_t bufsize )
```

[illegible]

Here is the caller graph for this function:



21.85.1.2 fdt_first_property_offset()

```

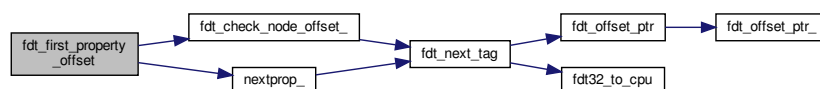
int fdt_first_property_offset (
    const void * fdt,
    int nodeoffset )

```

`fdt_first_property_offset` - find the offset of a node's first property : pointer to the device tree blob : structure block offset of a node

[fdt_first_property_offset\(\)](#) finds the first property of the node at the given structure block offset.

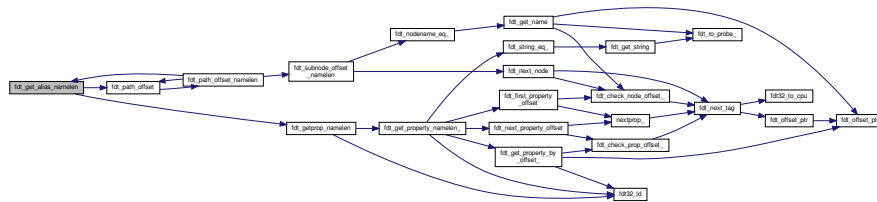
returns: structure block offset of the property (≥ 0), on success -FDT_ERR_NOTFOUND, if the requested node has no properties -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings. Here is the call graph for this function:



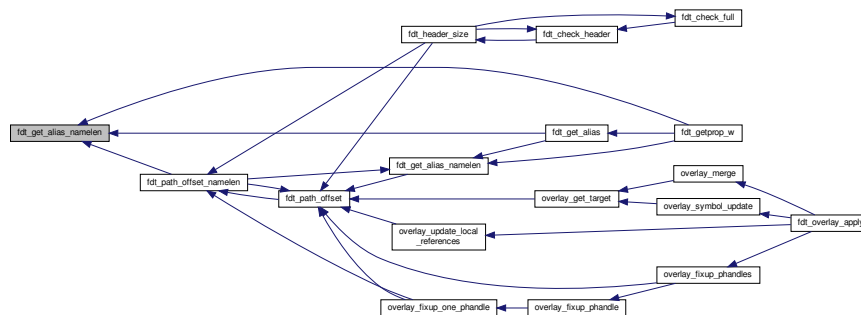
21.85.1.4 fdt_get_alias_namelen()

```
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:

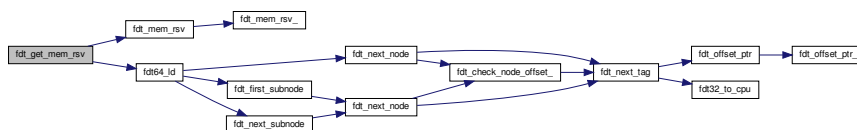
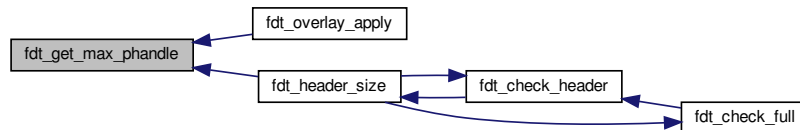
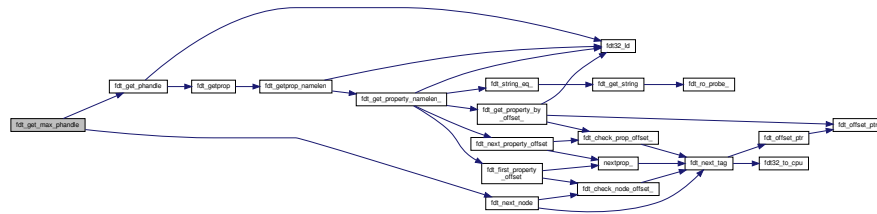


21.85.1.5 fdt_get_max_phandle()

```
uint32_t fdt_get_max_phandle (
    const void * fdt )
```

`fdt_get_max_phandle` - retrieves the highest phandle in a tree : 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.



```
const char* fdt_get_name (
    const void * fdt,
    int nodeoffset,
    int * lenp )
```

fdt_get_name() retrieves the name (including unit address) of the device tree node at structure block offset `nodeoffset`. If `lenp` is non-NULL, the length of this name is also returned, in the integer pointed to by `lenp`.

returns: pointer to the node's name, on success If lenp is non-NULL, *lenp contains the length of that name (>=0) NULL, on error if lenp is non-NULL *lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:

21.85.1.8 fdt_get_path()

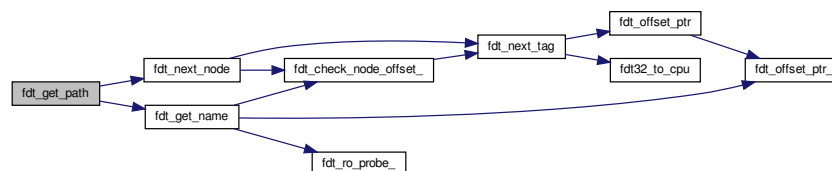
```
int fdt_get_path (
    const void * fdt,
    int nodeoffset,
    char * buf,
    int buflen )
```

fdt_get_path - determine the full path of a node : pointer to the device tree blob : offset of the node whose path to find : character buffer to contain the returned path (will be overwritten) : size of the character buffer at buf

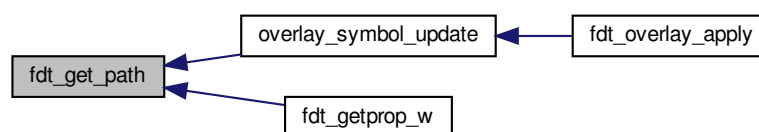
fdt_get_path() computes the full path of the node at offset nodeoffset, and records that path in the buffer at buf.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: 0, on success buf contains the absolute path of the node at nodeoffset, as a NUL-terminated string. -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOSPACE, the path of the given node is longer than (bufsize-1) characters and will not fit in the given buffer. -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



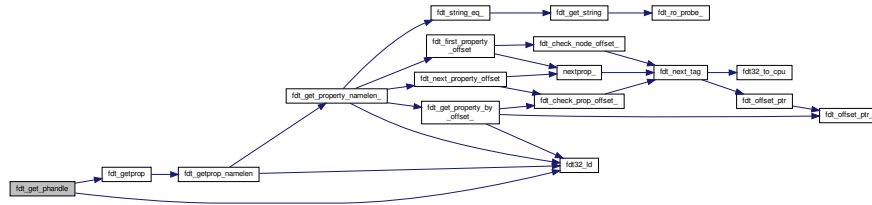
21.85.1.9 fdt_get_phandle()

```
uint32_t fdt_get_phandle (
    const void * fdt,
    int nodeoffset )
```

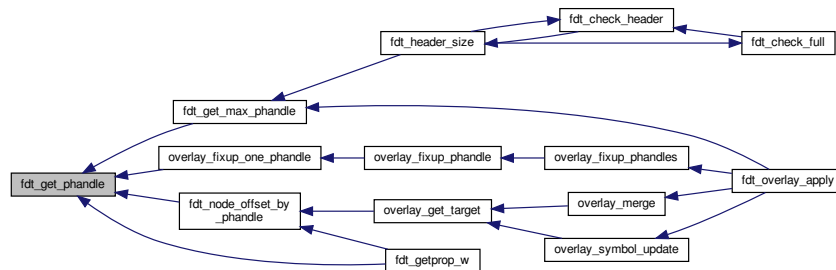
fdt_get_phandle - retrieve the phandle of a given node : pointer to the device tree blob : structure block offset of the node

[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:

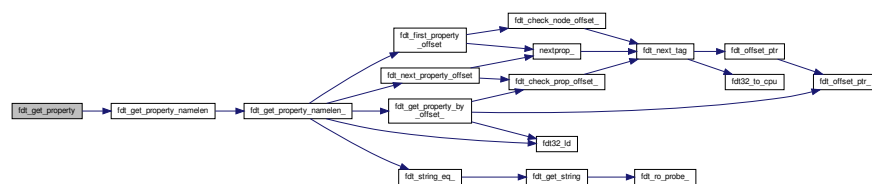


21.85.1.10 fdt_get_property()

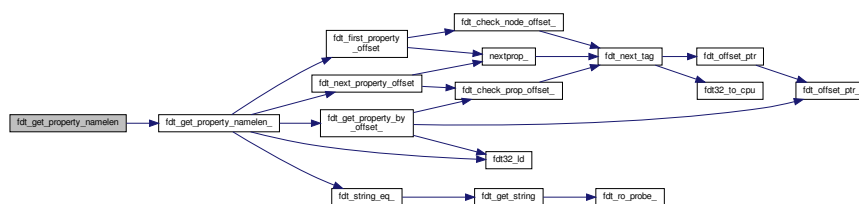
```

const struct fdt_property* fdt_get_property (
    const void * fdt,
    int nodeoffset,
    const char * name,
    int * lenp )
  
```

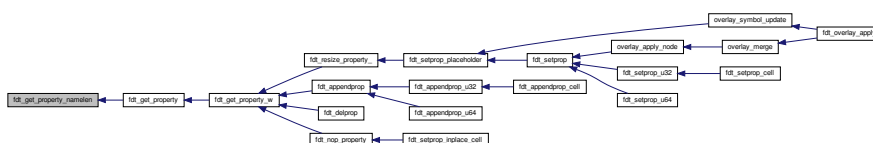
Here is the call graph for this function:



Here is the caller graph for this function:



Here is the caller graph for this function:



```
static const struct fdt\_property* fdt_get_property_namelen_ (
    const void * fdt,
    int offset,
    const char * name,
    int namelen,
    int * lenp,
    int * poffset ) [static]
```

[illegible]


```

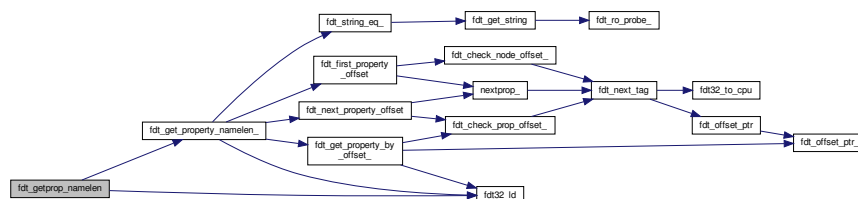
graph LR
    Entry[fdt_getprop_namelen] --> FdtGetStringEq[fdt_string_eq_]
    Entry --> FdtFirstPropertyOffset[fdt_first_property_offset_]
    Entry --> FdtNextPropertyOffset[fdt_next_property_offset_]
    Entry --> FdtGetPropertyByNameLen[fdt_get_property_namelen_]
    Entry --> FdtGetPropNamelen[fdt_getprop_namelen]
    Entry --> Fdt32Id[fdt32_id]

    FdtGetStringEq --> FdtGetString[fdt_get_string_]
    FdtGetString --> FdtRoProbe[fdt_ro_probe_]

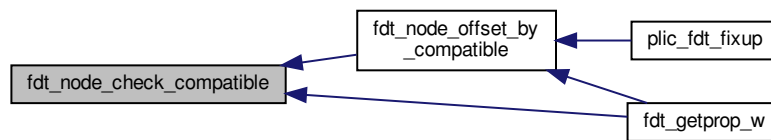
    FdtFirstPropertyOffset --> FdtCheckNodeOffset[fdt_check_node_offset_]
    FdtNextPropertyOffset --> FdtCheckNodeOffset
    FdtCheckNodeOffset --> NextProp[nextprop_]
    NextProp --> FdtNextTag[fdt_next_tag_]
    FdtNextTag --> Fdt32Cpu[fdt32_to_cpu]
    Fdt32Cpu --> FdtOffsetPtr[fdt_offset_ptr_]

    FdtGetPropertyByNameLen --> FdtCheckPropOffset[fdt_check_prop_offset_]
    FdtGetPropNamelen --> FdtCheckPropOffset
    FdtCheckPropOffset --> FdtOffsetPtr

    FdtOffsetPtr --> FdtOffsetPtrRet[fdt_offset_ptr_]
  
```



Here is the caller graph for this function:



21.85.1.22 fdt_node_depth()

```

int fdt_node_depth (
    const void * fdt,
    int nodeoffset )

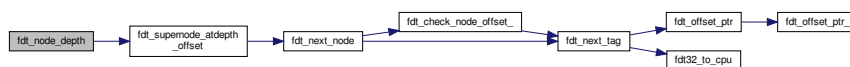
```

fdt_node_depth - find the depth of a given node : pointer to the device tree blob : offset of the node whose parent to find

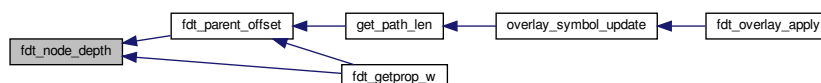
fdt_node_depth() finds the depth of a given node. The root node has depth 0, its immediate subnodes depth 1 and so forth.

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:



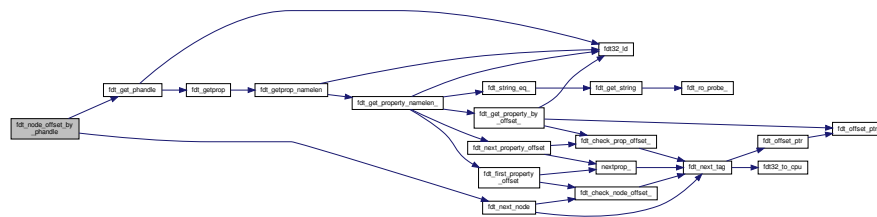
21.85.1.24 fdt_node_offset_by_phandle()

```
int fdt_node_offset_by_phandle (
    const void * fdt,
    uint32_t phandle )
```

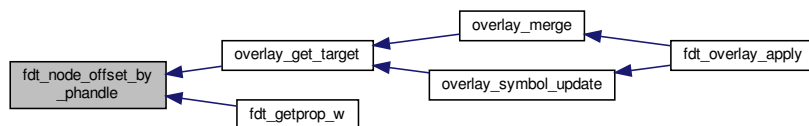
fdt_node_offset_by_phandle - find the node with a given phandle : pointer to the device tree blob : phandle value

`fdt_node_offset_by_phandle()` returns the offset of the node which has the given phandle value. If there is more than one node in the tree with the given phandle (an invalid tree), results are undefined.

returns: structure block offset of the located node (≥ 0), on success -FDT_ERR_NOTFOUND, no node with that phandle exists -FDT_ERR_BADPHANDLE, given phandle value was invalid (0 or -1) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



21.85.1.25 fdt_node_offset_by_prop_value()

```
int fdt_node_offset_by_prop_value (
    const void * fdt,
    int startoffset,
    const char * propname,
    const void * propval,
    int proplen )
```

fdt_node_offset_by_prop_value - find nodes with a given property value : pointer to the device tree blob : only find nodes after this offset : property name to check : property value to search for : 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 `proprname` 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.


```
int fdt_num_mem_rsv (
    const void * fdt )
```

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.

```

graph LR
    fdt_num_mem_rsv[fdt_num_mem_rsv] --> fdt_mem_rsv[fdt_mem_rsv]
    fdt_num_mem_rsv --> fdt64_id[fdt64_id]
    fdt_mem_rsv --> fdt_mem_rsv_2[fdt_mem_rsv_2]
    fdt64_id --> fdt_next_node_1[fdt_next_node]
    fdt64_id --> fdt_first_subnode[fdt_first_subnode]
    fdt64_id --> fdt_next_subnode[fdt_next_subnode]
    fdt_next_node_1 --> fdt_check_node_offset[fdt_check_node_offset]
    fdt_first_subnode --> fdt_next_node_2[fdt_next_node]
    fdt_next_subnode --> fdt_next_node_2
    fdt_check_node_offset --> fdt_next_tag[fdt_next_tag]
    fdt_next_node_2 --> fdt_next_tag
    fdt_next_tag --> fdt_offset_ptr[fdt_offset_ptr]
    fdt_next_tag --> fdt32_to_cpu[fdt32_to_cpu]
    fdt_offset_ptr --> fdt_offset_ptr_2[fdt_offset_ptr_2]
  
```

The flowchart illustrates the logic of the `fdt_next_node` function. It starts with `fdt_num_mem_rsv`, which branches to `fdt_mem_rsv` and `fdt64_id`. `fdt_mem_rsv` points to `fdt_mem_rsv_2`. `fdt64_id` branches to `fdt_next_node`, `fdt_first_subnode`, and `fdt_next_subnode`. `fdt_next_node` (from `fdt64_id`) points to `fdt_check_node_offset`. `fdt_first_subnode` and `fdt_next_subnode` both point to `fdt_next_node`. `fdt_check_node_offset` and `fdt_next_node` (from `fdt_first_subnode` and `fdt_next_subnode`) both point to `fdt_next_tag`. `fdt_next_tag` branches to `fdt_offset_ptr` and `fdt32_to_cpu`. `fdt_offset_ptr` points to `fdt_offset_ptr_2`.

```

graph TD
    fdt_header_size[fdt_header_size] <--> fdt_check_full[fdt_check_full]
    fdt_header_size --> fdt_num_mem_rsv[fdt_num_mem_rsv]
    fdt_check_full --> fdt_num_mem_rsv
    fdt_add_mem_rsv[fdt_add_mem_rsv] --> fdt_num_mem_rsv
    fdt_del_mem_rsv[fdt_del_mem_rsv] --> fdt_num_mem_rsv
    fdt_create_empty_tree[fdt_create_empty_tree] --> fdt_open_into[fdt_open_into]
    fdt_open_into --> fdt_num_mem_rsv
    fdt_pack[fdt_pack] --> fdt_num_mem_rsv
  
```

21.85.1.28 fdt_parent_offset()

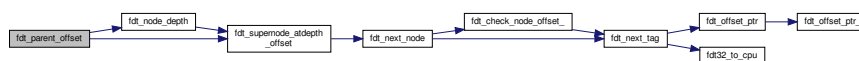
```
int fdt_parent_offset (
    const void * fdt,
    int nodeoffset )
```

fdt_parent_offset - find the parent of a given node : pointer to the device tree blob : offset of the node whose parent to find

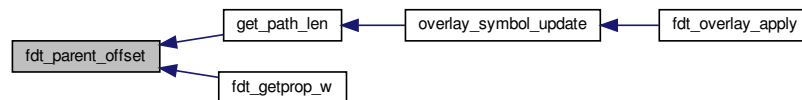
fdt_parent_offset() locates the parent node of a given node (that is, it finds the offset of the node which contains the node at nodeoffset as a subnode).

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset, *twice*.

returns: structure block offset of the parent of the node at nodeoffset (≥ 0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



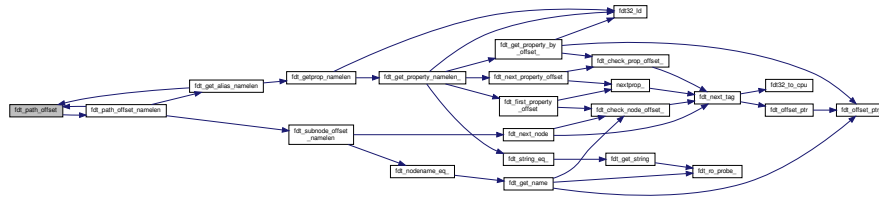
21.85.1.29 fdt_path_offset()

```
int fdt_path_offset (
    const void * fdt,
    const char * path )
```

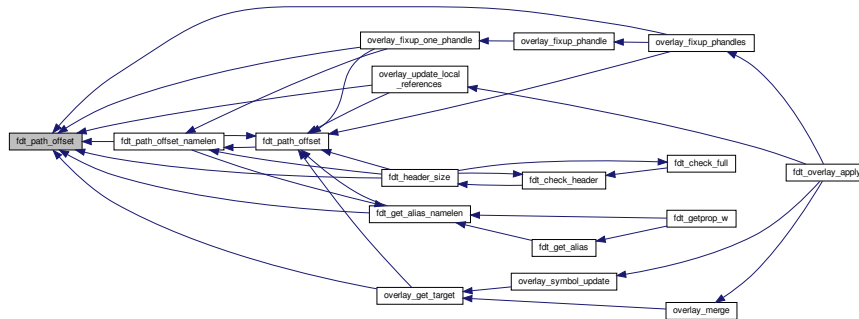
fdt_path_offset - find a tree node by its full path : pointer to the device tree blob : full path of the node to locate

fdt_path_offset() finds a node of a given path in the device tree. Each path component may omit the unit address portion, but the results of this are undefined if any such path component is ambiguous (that is if there are multiple nodes at the relevant level matching the given component, differentiated only by unit address).

returns: structure block offset of the node with the requested path (≥ 0), on success -FDT_ERR_BADPATH, given path does not begin with '/' or is invalid -FDT_ERR_NOTFOUND, if the requested node does not exist -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings. Here is the call graph for this function:



Here is the caller graph for this function:

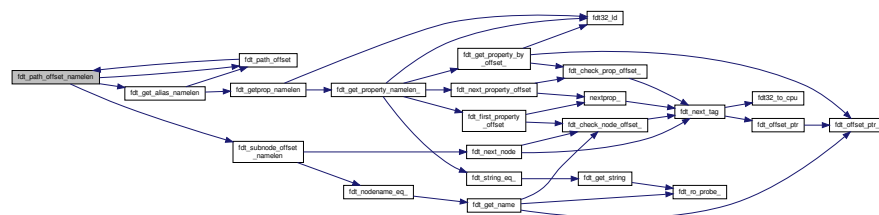


21.85.1.30 fdt_path_offset_namelen()

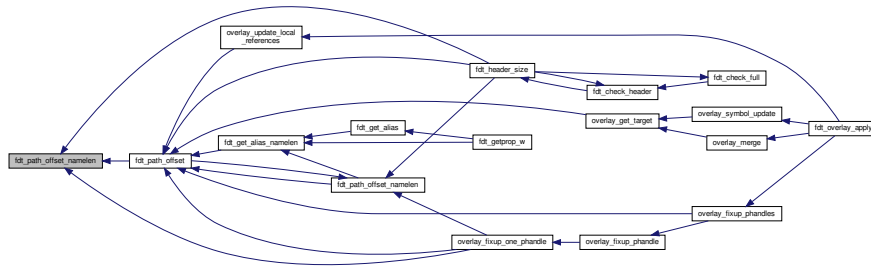
```
int fdt_path_offset_namelen (
    const void * fdt,
    const char * path,
    int namelen )
```

fdt_path_offset_namelen - find a tree node by its full path : pointer to the device tree blob : full path of the node to locate : 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:



21.85.1.31 fdt_string()

```

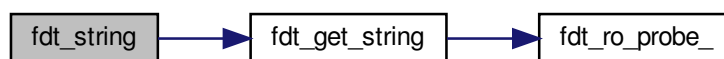
const char* fdt_string (
    const void * fdt,
    int stroffset )

```

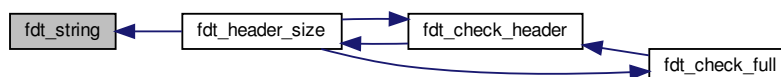
fdt_string - retrieve a string from the strings block of a device tree : pointer to the device tree blob : offset of the string within the strings block (native endian)

fdt_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt.

returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here is the call graph for this function:



Here is the caller graph for this function:



```
static int fdt_string_eq (
    const void * fdt,
    int stroffset,
    const char * s,
    int len ) [static]
```

```
graph LR; fdt_string_eq_ --> fdt_get_string; fdt_get_string --> fdt_ro_probe_;
```

[illegible]

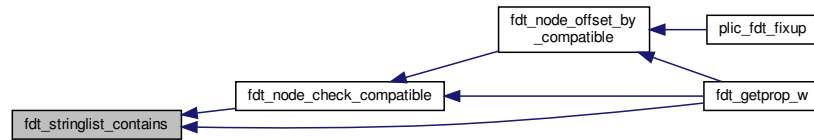
```
int fdt_stringlist_contains (
    const char * strlist,
    int listlen,
    const char * str )
```

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

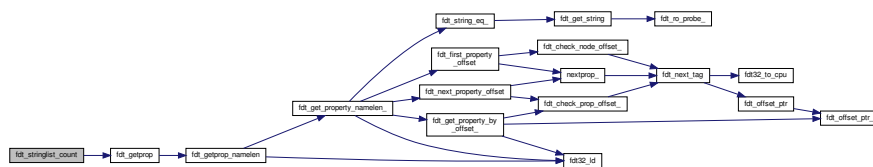
Here is the caller graph for this function:

**21.85.1.34 fdt_stringlist_count()**

```

int fdt_stringlist_count (
    const void * fdt,
    int nodeoffset,
    const char * property )
  
```

Here is the call graph for this function:



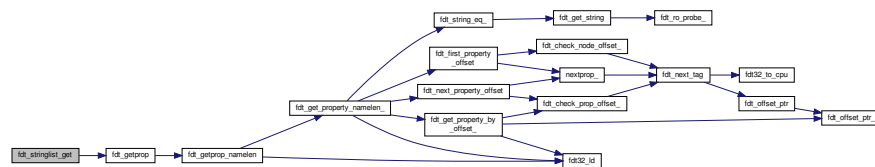
Here is the caller graph for this function:



21.85.1.35 fdt_stringlist_get()

```
const char* fdt_stringlist_get (
    const void * fdt,
    int nodeoffset,
    const char * property,
    int idx,
    int * lenp )
```

Here is the call graph for this function:



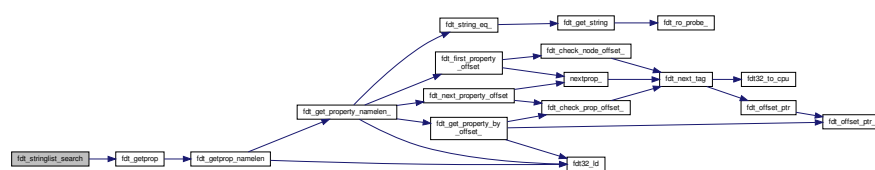
Here is the caller graph for this function:



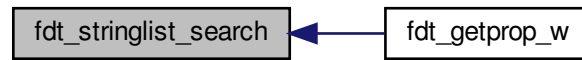
21.85.1.36 fdt_stringlist_search()

```
int fdt_stringlist_search (
    const void * fdt,
    int nodeoffset,
    const char * property,
    const char * string )
```

Here is the call graph for this function:



Here is the caller graph for this function:



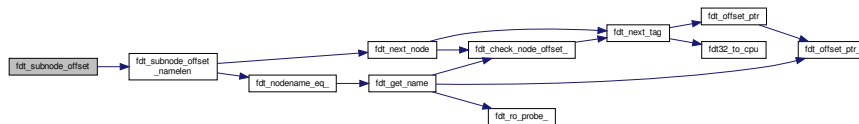
21.85.1.37 fdt_subnode_offset()

```

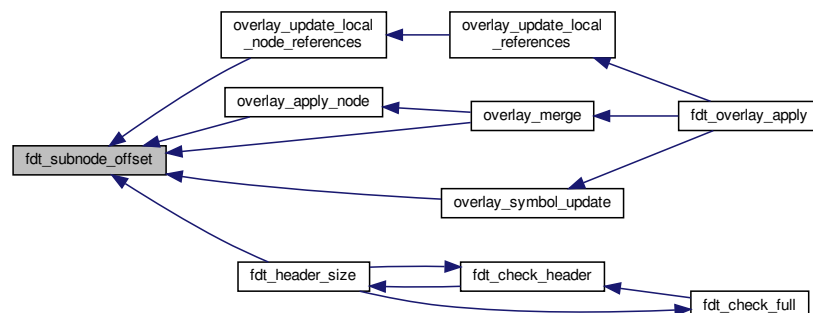
int fdt_subnode_offset (
    const void * fdt,
    int parentoffset,
    const char * name )

```

Here is the call graph for this function:



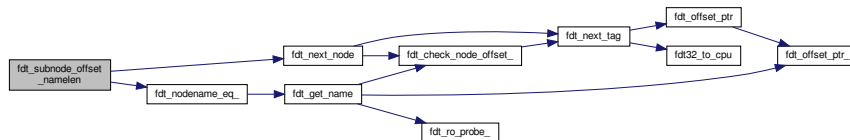
Here is the caller graph for this function:



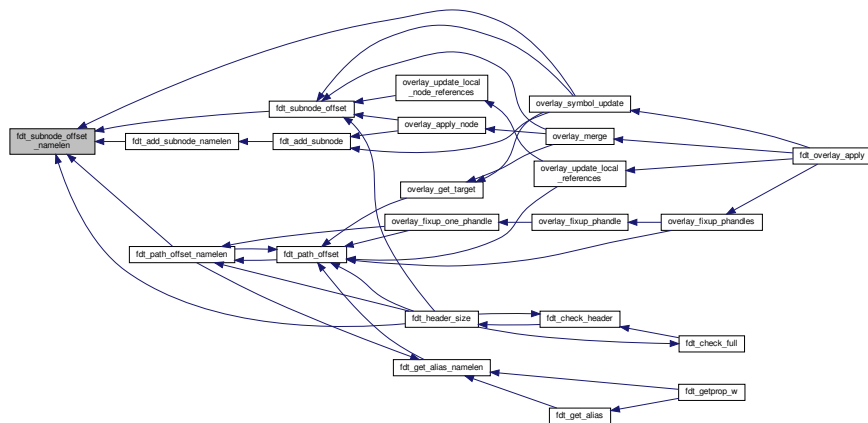
21.85.1.38 fdt_subnode_offset_namelen()

```
int fdt_subnode_offset_namelen (
    const void * fdt,
    int offset,
    const char * name,
    int namelen )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.85.1.39 fdt_supernode_atdepth_offset()

```
int fdt_supernode_atdepth_offset (
    const void * fdt,
    int nodeoffset,
    int supernodedepth,
    int * nodedepth )
```

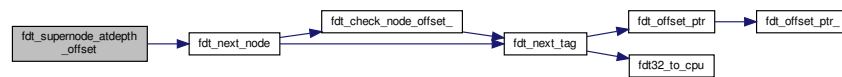
fdt_supernode_atdepth_offset - find a specific ancestor of a node : pointer to the device tree blob : offset of the node whose parent to find : depth of the ancestor to find : pointer to an integer variable (will be overwritten) or NULL

`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,`

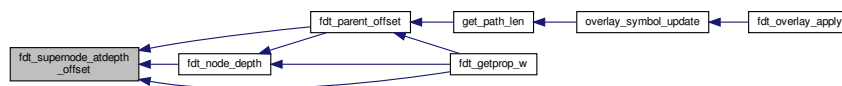
0, NULL); will always return 0, the offset of the root node. If the node at nodeoffset has depth D, then: fdt_↵
supernode_atdepth_offset(fdt, nodeoffset, D, NULL); will return nodeoffset itself.

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, supernod-
edepth 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:

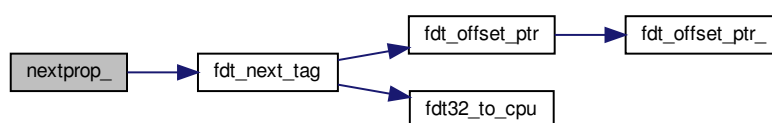


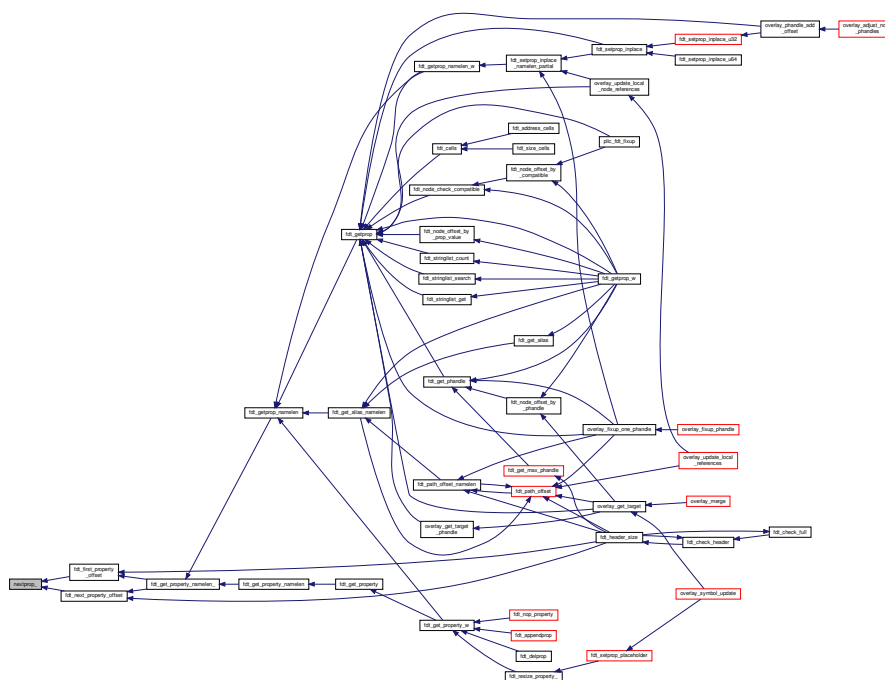
21.85.1.40 nextprop_()

```

static int nextprop_ (
    const void * fdt,
    int offset ) [static]
  
```

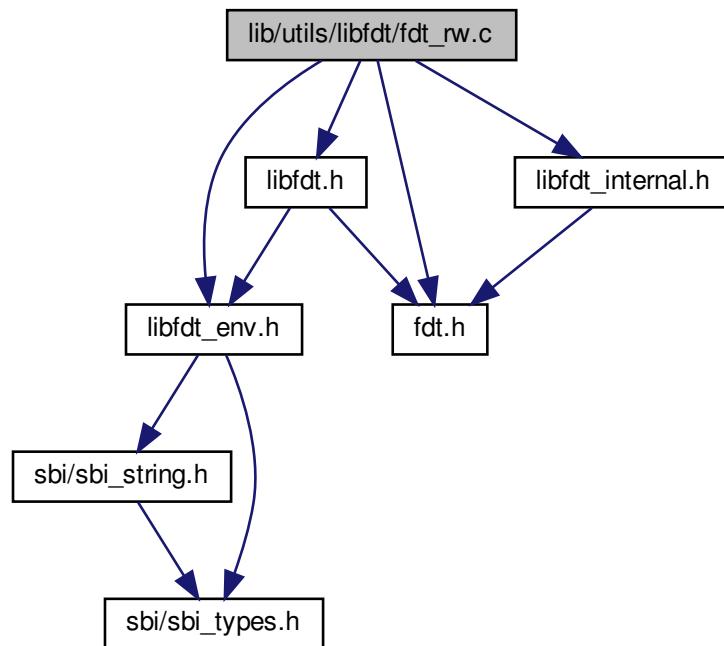
Here is the call graph for this function:





Generated by Doxygen

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 int `fdt_splice_string_` (void *fdt, int newlen)
- static int `fdt_find_add_string_` (void *fdt, const char *s)
- 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)

21.86.1 Macro Definition Documentation

21.86.1.1 FDT_RW_PROBE

```
#define FDT_RW_PROBE(  
    fdt )
```

Value:

```
{ \
    int err_; \
    if ((err_ = fdt\_rw\_probe_(fdt)) != 0) \
        return err_; \
}
```

21.86.2 Function Documentation

21.86.2.1 fdt_add_mem_rsv()

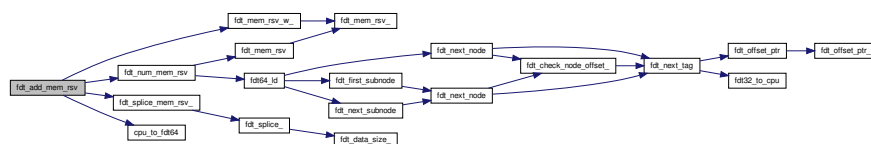
```
int fdt_add_mem_rsv (  
    void * fdt,  
    uint64_t address,  
    uint64_t size )
```

[fdt_add_mem_rsv](#) - add one memory reserve map entry : pointer to the device tree blob , : 64-bit values (native endian)

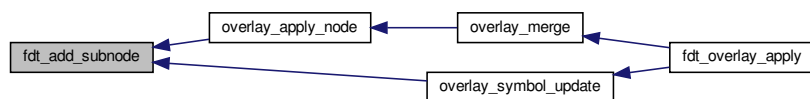
Adds a reserve map entry to the given blob reserving a region at address address of length size.

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:



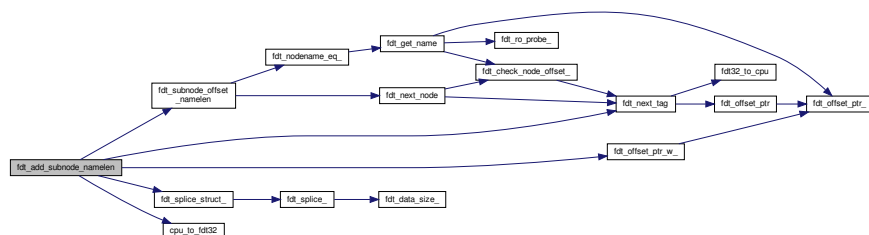
Here is the caller graph for this function:



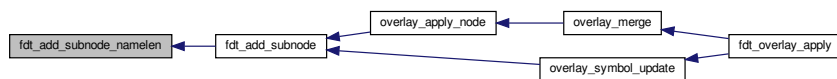
21.86.2.4 fdt_add_subnode_namelen()

```
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:



21.86.2.5 fdt_appendprop()

```
int fdt_appendprop (
    void * fdt,
    int nodeoffset,
    const char * name,
```



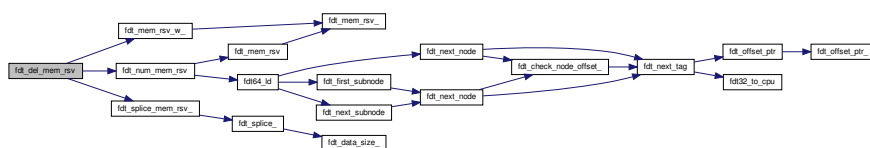
```
static int fdt_data_size_ (
    void * fdt ) [inline], [static]
```

[illegible]

```
int fdt_del_mem_rsv (
    void * fdt,
    int n )
```

`fdt_del_mem_rsv()` removes the n-th memory reserve map entry from the blob.

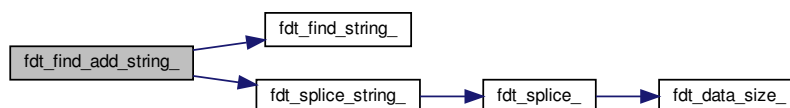
returns: 0, on success -FDT_ERR_NOTFOUND, there is no entry of the given index (i.e. there are less than n+1 reserve map entries) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



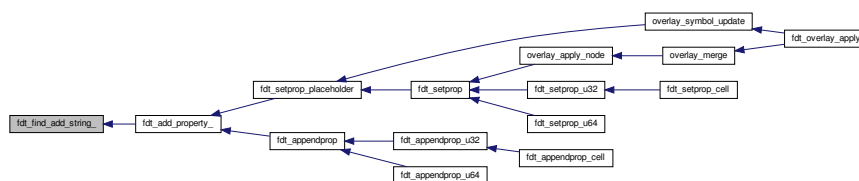
21.86.2.11 fdt_find_add_string()

```
static int fdt_find_add_string_ (
    void * fdt,
    const char * s ) [static]
```

Here is the call graph for this function:



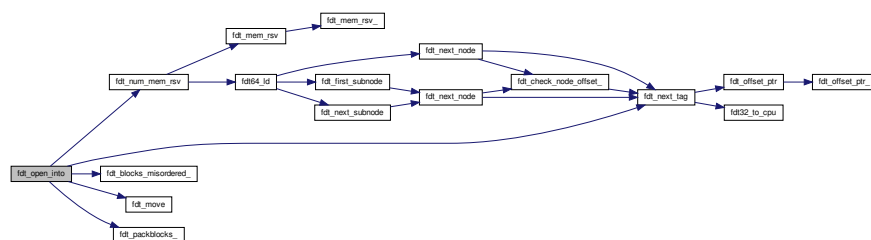
Here is the caller graph for this function:



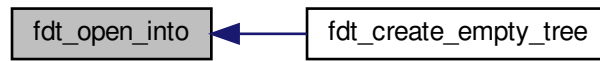
21.86.2.12 fdt_open_into()

```
int fdt_open_into (
    const void * fdt,
    void * buf,
    int bufsize )
```

Here is the call graph for this function:



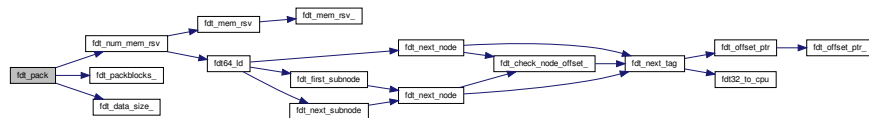
Here is the caller graph for this function:



21.86.2.13 fdt_pack()

```
int fdt_pack (
    void * fdt )
```

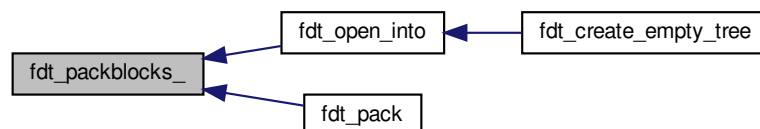
Here is the call graph for this function:



21.86.2.14 fdt_packblocks_()

```
static void fdt_packblocks_ (
    const char * old,
    char * new,
    int mem_rsv_size,
    int struct_size ) [static]
```

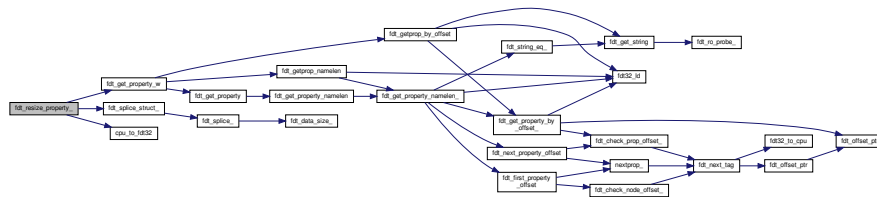
Here is the caller graph for this function:



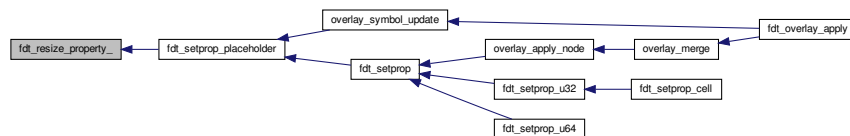
21.86.2.15 fdt_resize_property()

```
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:



21.86.2.16 fdt_rw_probe()

```
static int fdt_rw_probe_ (
    void * fdt ) [static]
```

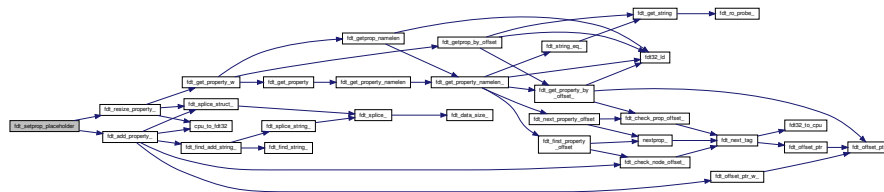
Here is the call graph for this function:



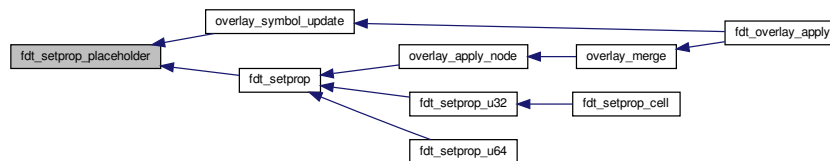
21.86.2.19 fdt_setprop_placeholder()

```
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:



21.86.2.20 fdt_splice_()

```
static int fdt_splice_ (
    void * fdt,
    void * splicepoint,
    int oldlen,
    int newlen ) [static]
```

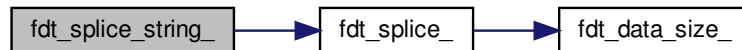
Here is the call graph for this function:



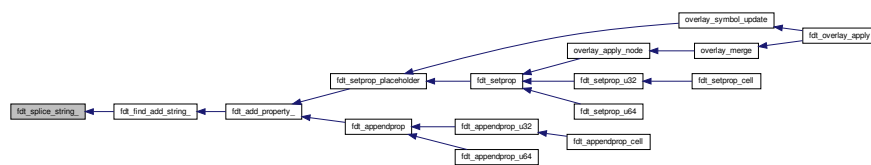
21.86.2.22 fdt_splice_string_()

```
static int fdt_splice_string_ (
    void * fdt,
    int newlen ) [static]
```

Here is the call graph for this function:



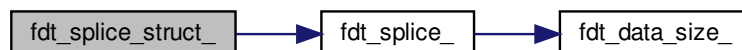
Here is the caller graph for this function:



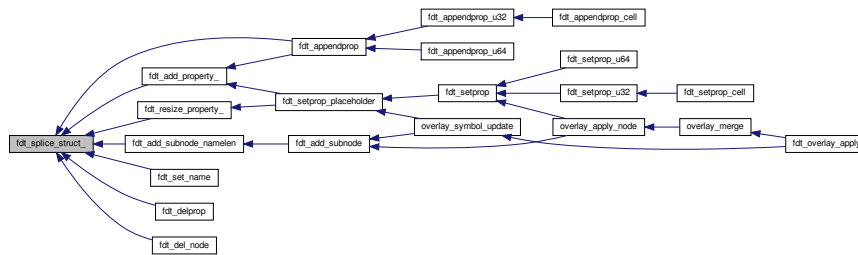
21.86.2.23 fdt_splice_struct_()

```
static int fdt_splice_struct_ (
    void * fdt,
    void * p,
    int oldlen,
    int newlen ) [static]
```

Here is the call graph for this function:

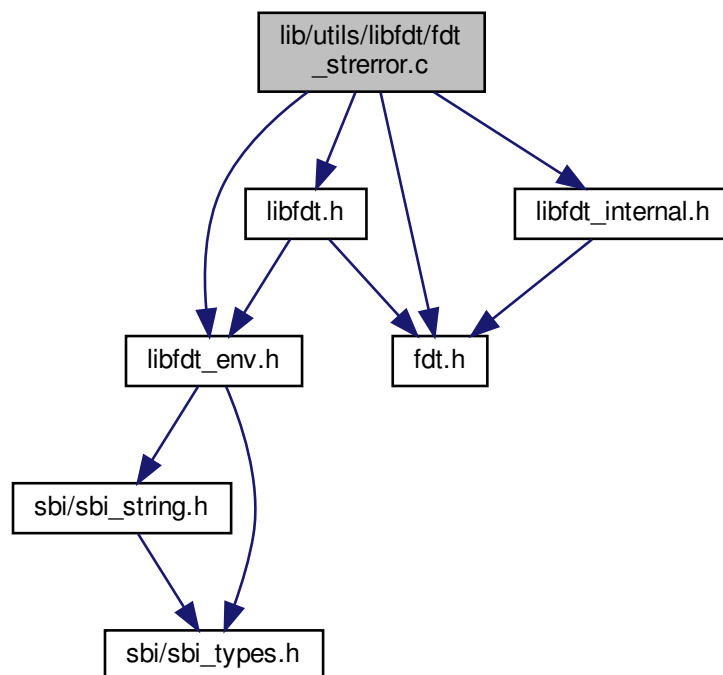


Here is the caller graph for this function:



21.87 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_ERRTABSIZ (sizeof(fdt_errtable) / sizeof(fdt_errtable[0]))`

Functions

- const char * [fdt_strerror](#) (int errval)

Variables

- static struct fdt_errtabent [fdt_errtable](#) []

21.87.1 Macro Definition Documentation

21.87.1.1 FDT_ERRTABENT

```
#define FDT_ERRTABENT(  
    val ) [(val)] = { .str = #val, }
```

21.87.1.2 FDT_ERRTABSIZE

```
#define FDT_ERRTABSIZE (sizeof(fdt\_errtable) / sizeof(fdt\_errtable[0]))
```

21.87.2 Function Documentation

21.87.2.1 fdt_strerror()

```
const char* fdt_strerror (  
    int errval )
```

21.87.3 Variable Documentation

21.87.3.1 fdt_errtable

```
struct fdt_errtabent fdt_errtable[] [static]
```

Initial value:

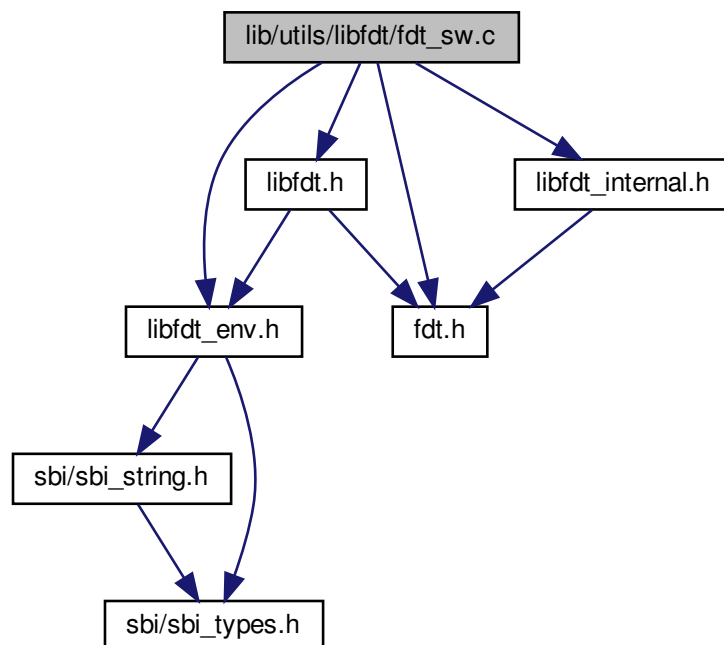
```
= {
    FDT_ERRTABENT(FDT_ERR_NOTFOUND),
    FDT_ERRTABENT(FDT_ERR_EXISTS),
    FDT_ERRTABENT(FDT_ERR_NOSPACE),

    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),
}
```

21.88 lib/utls/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 void * `fdt_grab_space_` (void *fdt, `size_t` len)
- int `fdt_create` (void *buf, int bufsz)
- int `fdt_resize` (void *fdt, void *buf, int bufsz)
- 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_find_add_string_` (void *fdt, const char *s)
- 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)

21.88.1 Macro Definition Documentation

21.88.1.1 FDT_SW_PROBE

```
#define FDT_SW_PROBE(  
    fdt )
```

Value:

```
{ \
    int err; \
    if ((err = fdt_sw_probe_(fdt)) != 0) \
        return err; \
}
```

21.88.1.2 FDT_SW_PROBE_MEMRSV

```
#define FDT_SW_PROBE_MEMRSV(  
    fdt )
```

Value:

```
{ \
    int err; \
    if ((err = fdt_sw_probe_memrsv_(fdt)) != 0) \
        return err; \
}
```

21.88.1.3 FDT_SW_PROBE_STRUCT

```
#define FDT_SW_PROBE_STRUCT(  
    fdt )
```

Value:

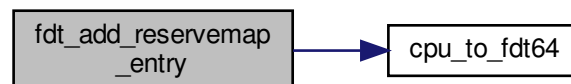
```
{ \n    int err; \n    if ((err = fdt_sw_probe_struct_(fdt)) != 0) \n        return err; \n}
```

21.88.2 Function Documentation

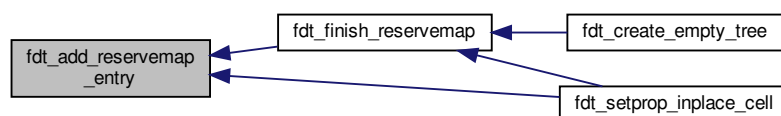
21.88.2.1 fdt_add_reservemap_entry()

```
int fdt_add_reservemap_entry (  
    void * fdt,  
    uint64_t addr,  
    uint64_t size )
```

Here is the call graph for this function:



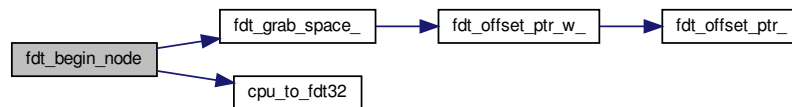
Here is the caller graph for this function:



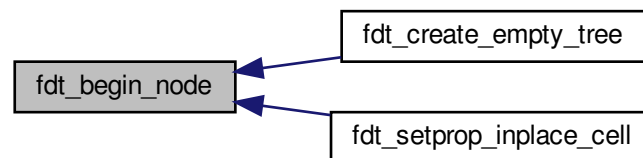
21.88.2.2 fdt_begin_node()

```
int fdt_begin_node (
    void * fdt,
    const char * name )
```

Here is the call graph for this function:



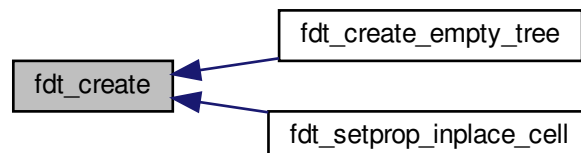
Here is the caller graph for this function:



21.88.2.3 fdt_create()

```
int fdt_create (
    void * buf,
    int bufsize )
```

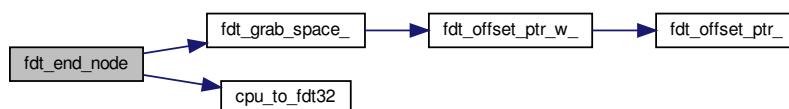
Here is the caller graph for this function:



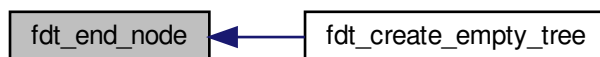
21.88.2.4 fdt_end_node()

```
int fdt_end_node (
    void * fdt )
```

Here is the call graph for this function:



Here is the caller graph for this function:



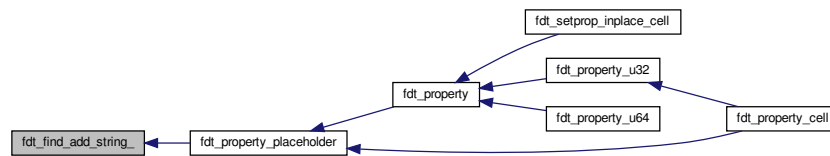
21.88.2.5 fdt_find_add_string_()

```
static int fdt_find_add_string_ (
    void * fdt,
    const char * s ) [static]
```

Here is the call graph for this function:



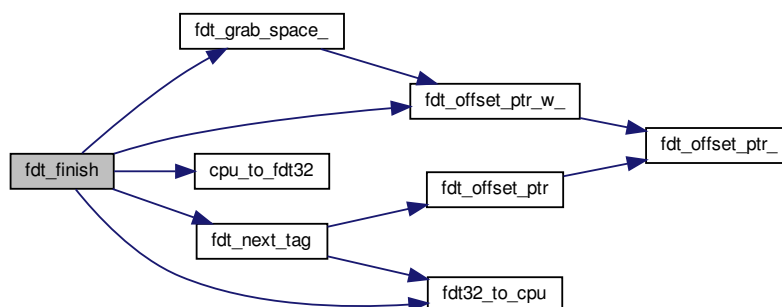
Here is the caller graph for this function:



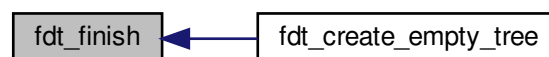
21.88.2.6 fdt_finish()

```
int fdt_finish (
    void * fdt )
```

Here is the call graph for this function:



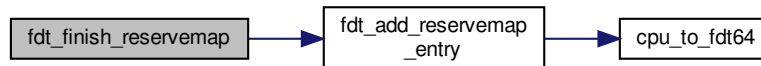
Here is the caller graph for this function:



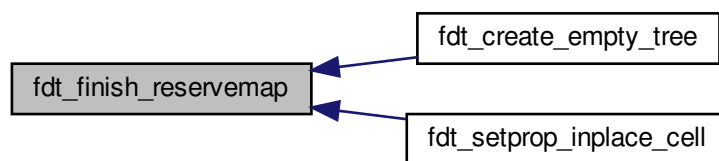
21.88.2.7 fdt_finish_reservemap()

```
int fdt_finish_reservemap (  
    void * fdt )
```

Here is the call graph for this function:



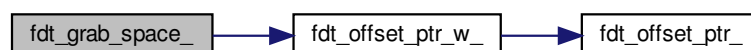
Here is the caller graph for this function:



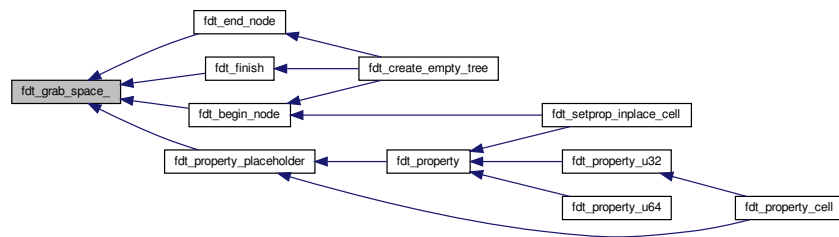
21.88.2.8 fdt_grab_space_()

```
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:



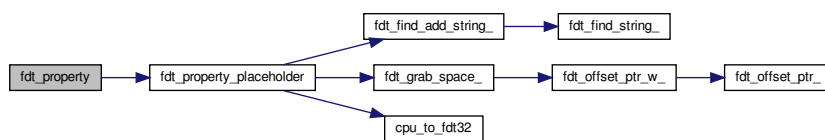
21.88.2.9 fdt_property()

```

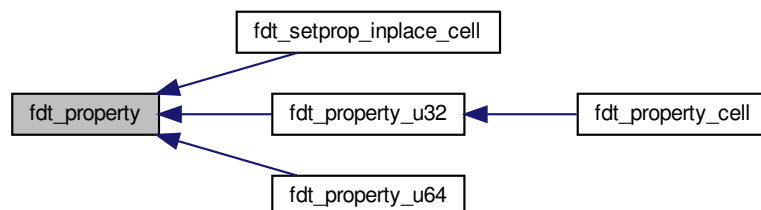
int fdt_property (
    void * fdt,
    const char * name,
    const void * val,
    int len )

```

Here is the call graph for this function:



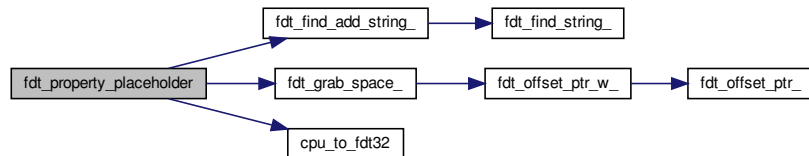
Here is the caller graph for this function:



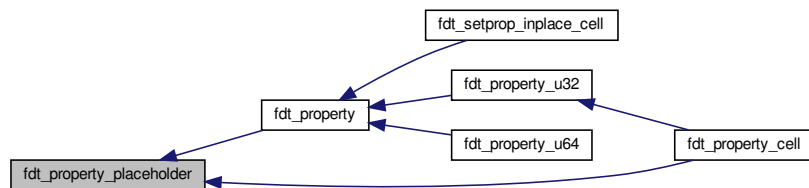
21.88.2.10 fdt_property_placeholder()

```
int fdt_property_placeholder (
    void * fdt,
    const char * name,
    int len,
    void ** valp )
```

Here is the call graph for this function:



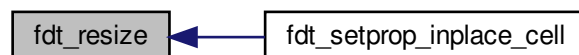
Here is the caller graph for this function:



21.88.2.11 fdt_resize()

```
int fdt_resize (
    void * fdt,
    void * buf,
    int bufsize )
```

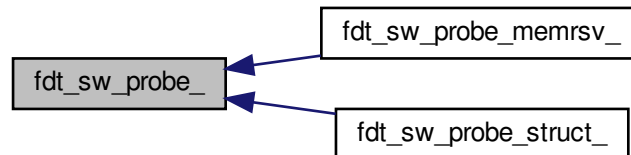
Here is the caller graph for this function:



21.88.2.12 fdt_sw_probe_()

```
static int fdt_sw_probe_ (  
    void * fdt ) [static]
```

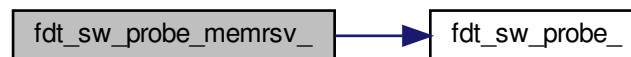
Here is the caller graph for this function:



21.88.2.13 fdt_sw_probe_memrsv_()

```
static int fdt_sw_probe_memrsv_ (  
    void * fdt ) [static]
```

Here is the call graph for this function:



21.88.2.14 fdt_sw_probe_struct_()

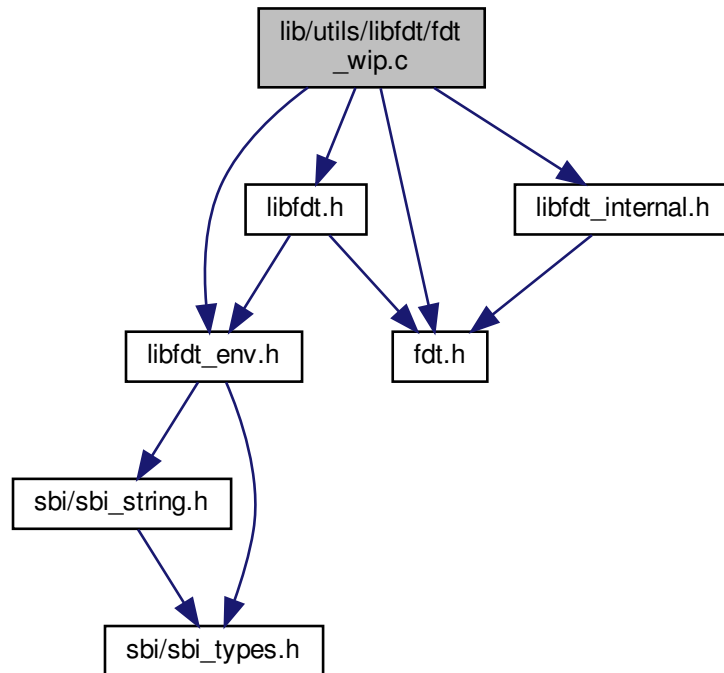
```
static int fdt_sw_probe_struct_ (  
    void * fdt ) [static]
```

Here is the call graph for this function:



21.89 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

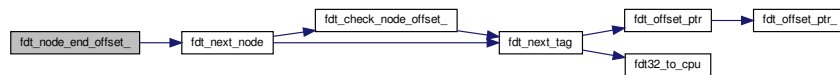
- `int fdt_setprop_inplace_namelen_partial` (void *fdt, int nodeoffset, const char *name, int namelen, [uint32_t](#) idx, const void *val, int len)
- `int fdt_setprop_inplace` (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- `static void fdt_nop_region_` (void *start, int len)
- `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)

21.89.1 Function Documentation

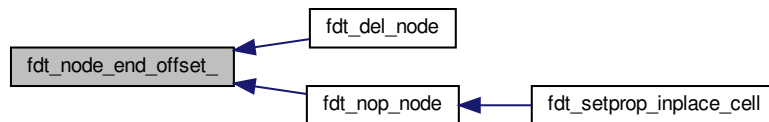
21.89.1.1 fdt_node_end_offset_()

```
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:



21.89.1.2 fdt_nop_node()

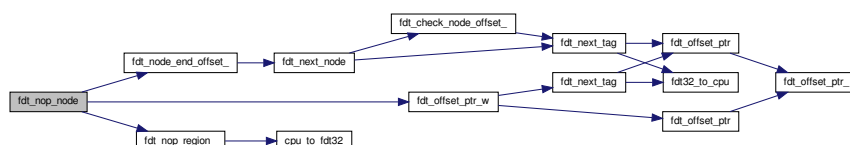
```
int fdt_nop_node (
    void * fdt,
    int nodeoffset )
```

fdt_nop_node - replace a node (subtree) with nop tags : pointer to the device tree blob : 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:




```
static void fdt_nop_region_ (
    void * start,
    int len ) [static]
```

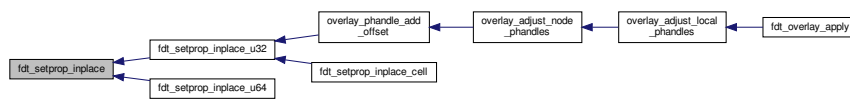
```
graph LR; fdt_nop_region_ --> cpu_to_fdt32
```

```
graph RL
    fdt_setprop_inplace_cell --> fdt_nop_property
    fdt_setprop_inplace_cell --> fdt_nop_node
    fdt_nop_property --> fdt_nop_region_
    fdt_nop_node --> fdt_nop_region_
    style fdt_nop_region_ fill:#ccc,stroke:#333,stroke-width:1px
```

```
int fdt_setprop_inplace (
    void * fdt,
    int nodeoffset,
    const char * name,
    const void * val,
    int len )
```

[illegible]

Here is the caller graph for this function:



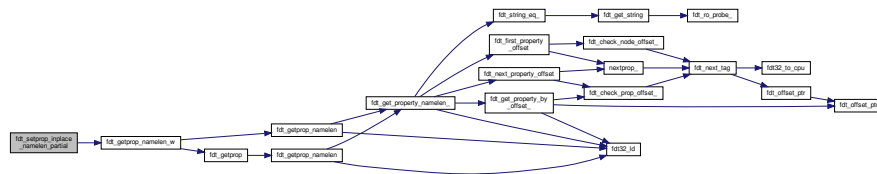
21.89.1.6 fdt_setprop_inplace_namelen_partial()

```

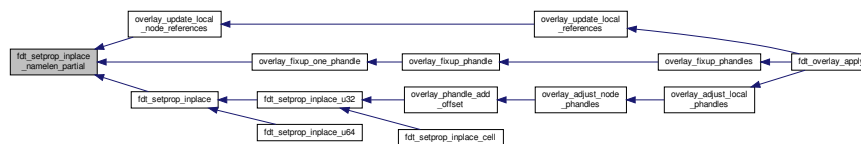
int fdt_setprop_inplace_namelen_partial (
    void * fdt,
    int nodeoffset,
    const char * name,
    int namelen,
    uint32_t idx,
    const void * val,
    int len )

```

Here is the call graph for this function:



Here is the caller graph for this function:



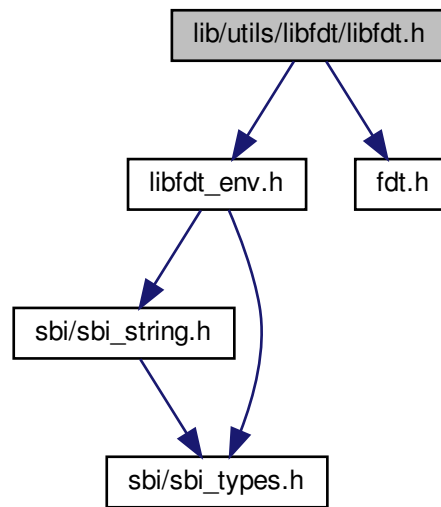
21.90 lib/utils/libfdt/libfdt.h File Reference

```

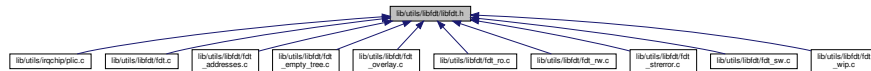
#include <libfdt_env.h>
#include <fdt.h>

```

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_BADNCCELLS 14`
- `#define FDT_ERR_BADVALUE 15`
- `#define FDT_ERR_BADOVERLAY 16`

- `#define FDT_ERR_NOPHANDLES 17`
- `#define FDT_ERR_MAX 17`
- `#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 uint64_t fdt64_ld (const fdt64_t *p)`
- `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)`
- `uint32_t fdt_get_max_phandle (const void *fdt)`
- `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 : pointer to the device tree blob : offset of the node whose property to find

: 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 of the property to nop

fdt_delprop - delete a property : pointer to the device tree blob : offset of the node whose property to nop

fdt_del_property() will delete the given property.

This function will delete data from the blob, and will therefore change the offsets of some existing nodes.

returns: 0, on success -FDT_ERR_NOTFOUND, node does not have the named property -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings

- int fdt_nop_property (void *fdt, int nodeoffset, const char *name)
- int fdt_nop_node (void *fdt, int nodeoffset)
- 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 to give the node

fdt_set_name - change the name of a given node : pointer to the device tree blob : 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 : pointer to the device tree blob : offset of the node whose property to change

: pointer to data to append to the property value : 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 the subnode to locate

fdt_add_subnode - creates a new node : pointer to the device tree blob : 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 node requested subnode (≥ 0), on success -FDT_ERR_NOTFOUND, if the requested subnode does not exist -FDT_ERR_BADOFFSET, if parentoffset did not point to an FDT_BEGIN_NODE tag -FDT_ERR_EXISTS, if the node at parentoffset already has a subnode of the given name -FDT_ERR_NOSPACE, if there is insufficient free space in the blob to contain the new node -FDT_ERR_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 to look up

`fdt_get_alias` - retrieve the path referenced by a given alias : 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 : pointer to the device tree blob : offset of the node whose property to change

: string value to append to the property

`fdt_appendprop_string()` appends the given string to the value of the named property in the given node, or creates a new property with that value if it does not already exist.

This function may insert data into the blob, and will therefore change the offsets of some existing nodes.

returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new property value -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADLAYOUT, -↵ FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -↵ FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings

- `#define fdt_setprop_string(fdt, nodeoffset, name, str) fdt_setprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)`
- `#define fdt_setprop_empty(fdt, nodeoffset, name) fdt_setprop((fdt), (nodeoffset), (name), NULL, 0)`
- `#define fdt_appendprop_string(fdt, nodeoffset, name, str) fdt_appendprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)`
- `int fdt_setprop_inplace_namelen_partial` (void *fdt, int nodeoffset, const char *name, int namelen, uint32_t idx, const void *val, int len)
- `int fdt_setprop_inplace` (void *fdt, int nodeoffset, const char *name, const void *val, int len)

- static int `fdt_setprop_inplace_u32` (void *fdt, int nodeoffset, const char *name, `uint32_t` val)
- static int `fdt_setprop_inplace_u64` (void *fdt, int nodeoffset, const char *name, `uint64_t` val)
- 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 property to add

`fdt_property_placeholder` - add a new property and return a ptr to its value

: pointer to the device tree blob

: length of property value in bytes : returns a pointer to 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)

21.90.1 Macro Definition Documentation

21.90.1.1 `fdt_appendprop_string`

```
#define fdt_appendprop_string(  
    fdt,  
    nodeoffset,  
    name,  
    str ) fdt_appendprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
```

21.90.1.2 `fdt_boot_cpuid_phys`

```
#define fdt_boot_cpuid_phys(  
    fdt ) (fdt_get_header(fdt, boot_cpuid_phys))
```


21.90.1.3 FDT_ERR_BADLAYOUT

```
#define FDT_ERR_BADLAYOUT 12
```

21.90.1.4 FDT_ERR_BADMAGIC

```
#define FDT_ERR_BADMAGIC 9
```

21.90.1.5 FDT_ERR_BADNCELLS

```
#define FDT_ERR_BADNCELLS 14
```

21.90.1.6 FDT_ERR_BADOFFSET

```
#define FDT_ERR_BADOFFSET 4
```

21.90.1.7 FDT_ERR_BADOVERLAY

```
#define FDT_ERR_BADOVERLAY 16
```

21.90.1.8 FDT_ERR_BADPATH

```
#define FDT_ERR_BADPATH 5
```

21.90.1.9 FDT_ERR_BADPHANDLE

```
#define FDT_ERR_BADPHANDLE 6
```

21.90.1.10 FDT_ERR_BADSTATE

```
#define FDT_ERR_BADSTATE 7
```

21.90.1.11 FDT_ERR_BADSTRUCTURE

```
#define FDT_ERR_BADSTRUCTURE 11
```

21.90.1.12 FDT_ERR_BADVALUE

```
#define FDT_ERR_BADVALUE 15
```

21.90.1.13 FDT_ERR_BADVERSION

```
#define FDT_ERR_BADVERSION 10
```

21.90.1.14 FDT_ERR_EXISTS

```
#define FDT_ERR_EXISTS 2
```

21.90.1.15 FDT_ERR_INTERNAL

```
#define FDT_ERR_INTERNAL 13
```

21.90.1.16 FDT_ERR_MAX

```
#define FDT_ERR_MAX 17
```

21.90.1.17 FDT_ERR_NOPHANDLES

```
#define FDT_ERR_NOPHANDLES 17
```

21.90.1.18 FDT_ERR_NOSPACE

```
#define FDT_ERR_NOSPACE 3
```

21.90.1.19 FDT_ERR_NOTFOUND

```
#define FDT_ERR_NOTFOUND 1
```

21.90.1.20 FDT_ERR_TRUNCATED

```
#define FDT_ERR_TRUNCATED 8
```

21.90.1.21 FDT_FIRST_SUPPORTED_VERSION

```
#define FDT_FIRST_SUPPORTED_VERSION 0x02
```

21.90.1.22 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 (int, lvalue) : FDT blob (const void *) : 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.

21.90.1.23 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

: child node (int, lvalue) : FDT blob (const void *) : 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 is used as iterator in the loop. The parent variable be constant or even a literal.

21.90.1.24 fdt_get_header

```
#define fdt_get_header(
    fdt,
    field ) (fdt32_ld(&((const struct fdt_header *) (fdt))->field))
```

21.90.1.25 fdt_last_comp_version

```
#define fdt_last_comp_version(
    fdt ) (fdt_get_header(fdt, last_comp_version))
```

21.90.1.26 FDT_LAST_SUPPORTED_VERSION

```
#define FDT_LAST_SUPPORTED_VERSION 0x11
```

21.90.1.27 fdt_magic

```
#define fdt_magic(
    fdt ) (fdt_get_header(fdt, magic))
```

21.90.1.28 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.

21.90.1.29 fdt_off_dt_strings

```
#define fdt_off_dt_strings(  
    fdt ) (fdt_get_header(fdt, off_dt_strings))
```

21.90.1.30 fdt_off_dt_struct

```
#define fdt_off_dt_struct(  
    fdt ) (fdt_get_header(fdt, off_dt_struct))
```

21.90.1.31 fdt_off_mem_rsvmap

```
#define fdt_off_mem_rsvmap(  
    fdt ) (fdt_get_header(fdt, off_mem_rsvmap))
```

21.90.1.32 fdt_property_string

```
#define fdt_property_string(  
    fdt,  
    name,  
    str ) fdt_property(fdt, name, str, strlen(str)+1)
```

21.90.1.33 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); \
}
```

21.90.1.34 fdt_setprop_empty

```
#define fdt_setprop_empty(  
    fdt,  
    nodeoffset,  
    name ) fdt_setprop((fdt), (nodeoffset), (name), NULL, 0)
```

21.90.1.35 fdt_setprop_string

```
#define fdt_setprop_string(  
    fdt,  
    nodeoffset,  
    name,  
    str ) fdt_setprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
```

21.90.1.36 fdt_size_dt_strings

```
#define fdt_size_dt_strings(  
    fdt ) (fdt_get_header(fdt, size_dt_strings))
```

21.90.1.37 fdt_size_dt_struct

```
#define fdt_size_dt_struct(  
    fdt ) (fdt_get_header(fdt, size_dt_struct))
```

21.90.1.38 fdt_totalsize

```
#define fdt_totalsize(  
    fdt ) (fdt_get_header(fdt, totalsize))
```

21.90.1.39 fdt_version

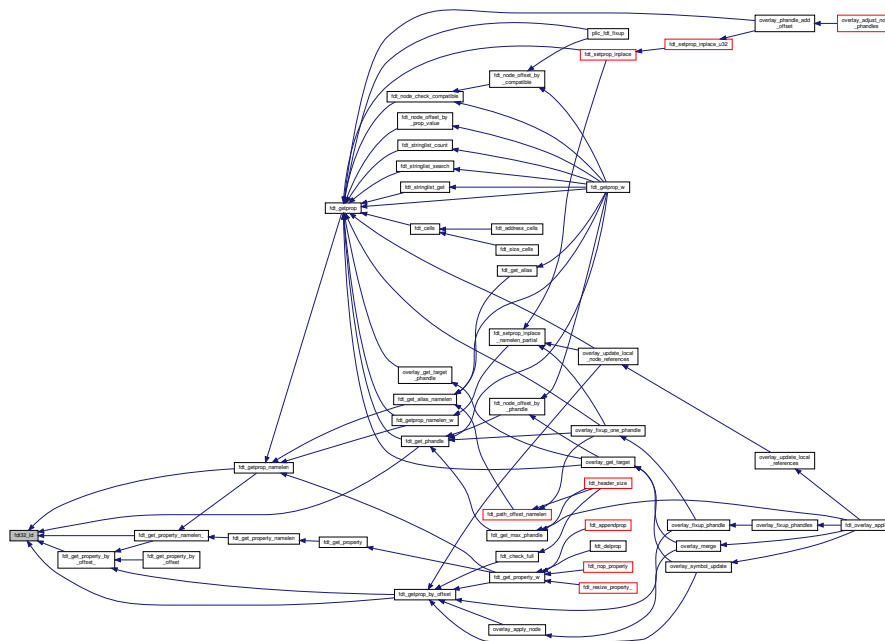
```
#define fdt_version(  
    fdt ) (fdt_get_header(fdt, version))
```

21.90.2 Function Documentation

21.90.2.1 fdt32_ld()

```
static uint32_t fdt32_ld (
    const fdt32_t * p ) [inline], [static]
```

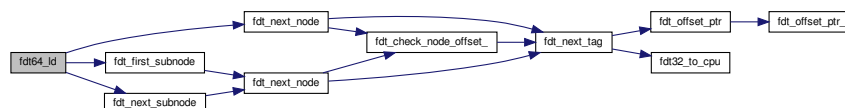
Here is the caller graph for this function:



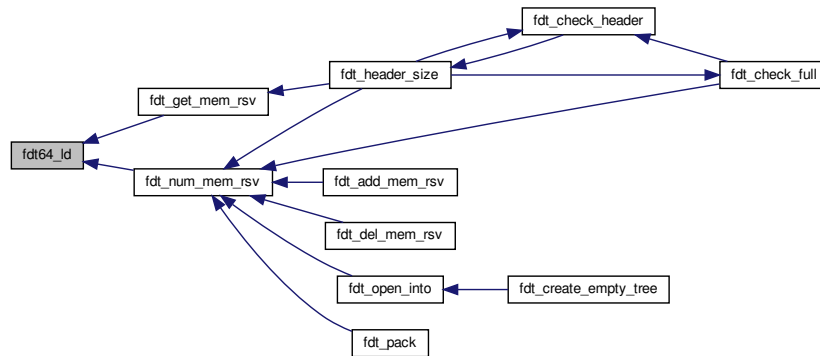
21.90.2.2 fdt64_ld()

```
static uint64_t fdt64_ld (
    const fdt64_t * p ) [inline], [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.90.2.3 fdt_add_mem_rsv()

```

int fdt_add_mem_rsv (
    void * fdt,
    uint64_t address,
    uint64_t size )

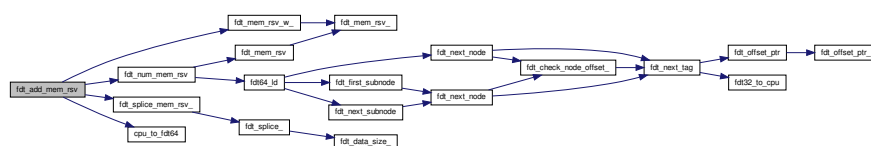
```

fdt_add_mem_rsv - add one memory reserve map entry : pointer to the device tree blob , : 64-bit values (native endian)

Adds a reserve map entry to the given blob reserving a region at address address of length size.

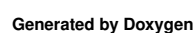
This function will insert data into the reserve map and will therefore change the indexes of some entries in the table.

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:

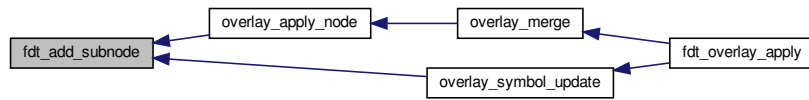



```
int fdt_add_reservemap_entry (
    void * fdt,
    uint64_t addr,
    uint64_t size )
```

```
int fdt_add_subnode (
    void * fdt,
    int parentoffset,
    const char * name )
```



Here is the caller graph for this function:



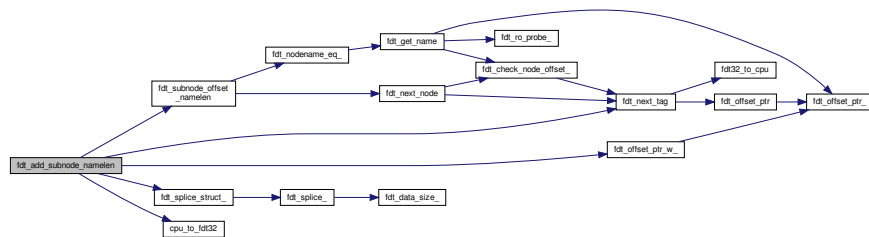
21.90.2.6 fdt_add_subnode_namelen()

```

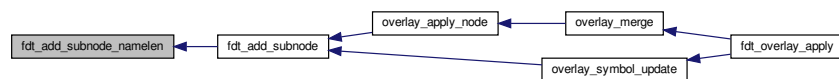
int fdt_add_subnode_namelen (
    void * fdt,
    int parentoffset,
    const char * name,
    int namelen )

```

Here is the call graph for this function:



Here is the caller graph for this function:



21.90.2.7 fdt_address_cells()

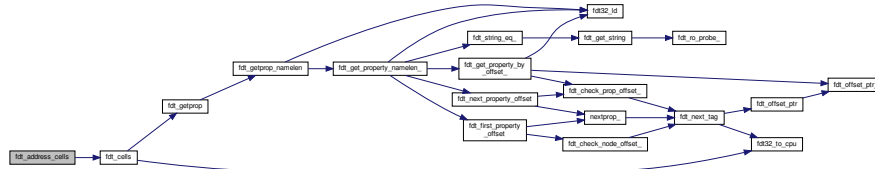
```

int fdt_address_cells (
    const void * fdt,
    int nodeoffset )

```

When the node has a valid `#address-cells` property, returns its value.

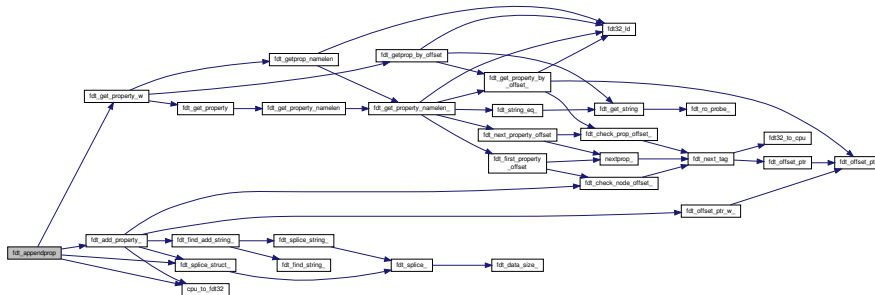
meanings Here is the call graph for this function:



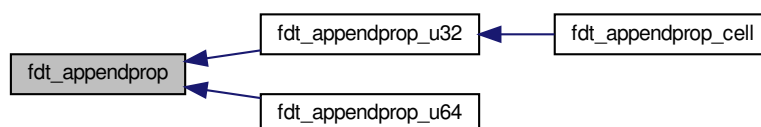
21.90.2.8 fdt_appendprop()

```
int fdt_appendprop (
    void * fdt,
    int nodeoffset,
    const char * name,
    const void * val,
    int len )
```

Here is the call graph for this function:



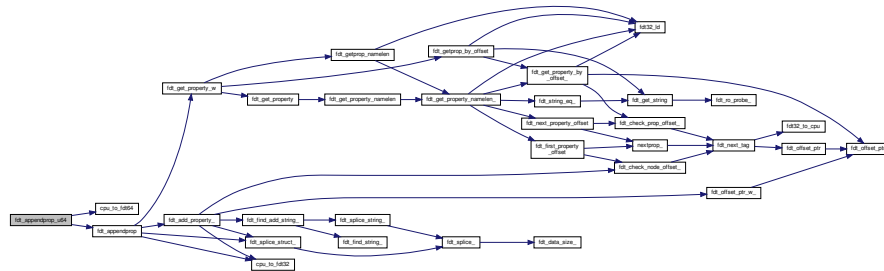
Here is the caller graph for this function:



21.90.2.11 fdt_appendprop_u64()

```
static int fdt_appendprop_u64 (
    void * fdt,
    int nodeoffset,
    const char * name,
    uint64_t val ) [inline], [static]
```

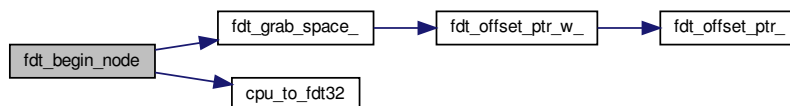
Here is the call graph for this function:



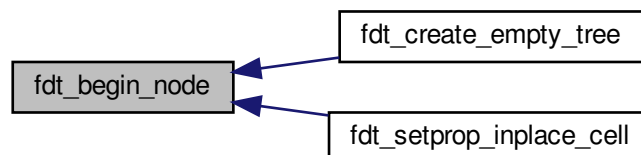
21.90.2.12 fdt_begin_node()

```
int fdt_begin_node (
    void * fdt,
    const char * name )
```

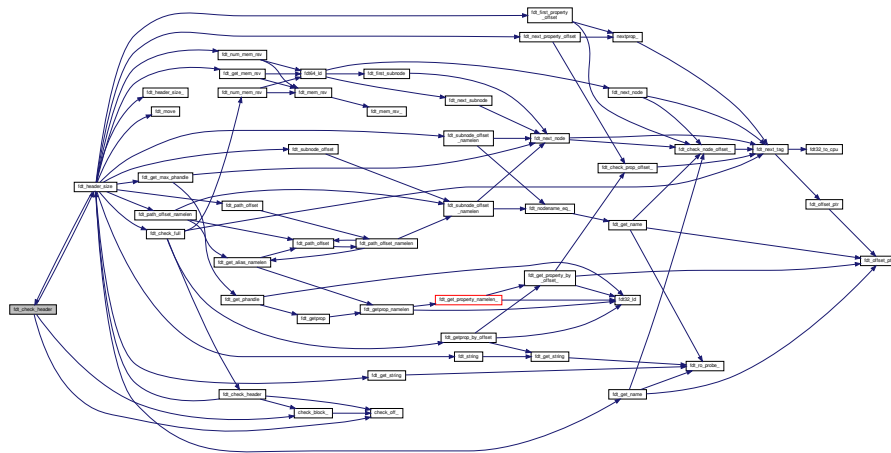
Here is the call graph for this function:



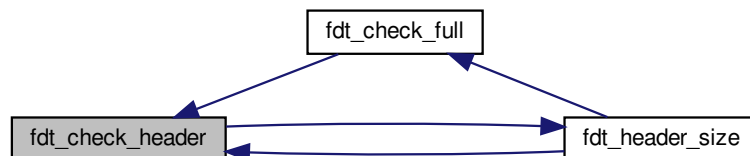
Here is the caller graph for this function:



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:



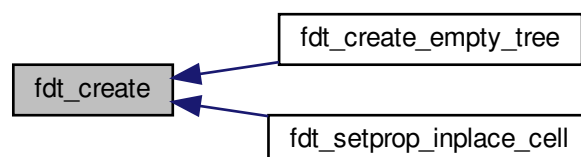
Here is the caller graph for this function:



21.90.2.15 fdt_create()

```
int fdt_create (
    void * buf,
    int bufsize )
```

Here is the caller graph for this function:



21.90.2.18 fdt_del_node()

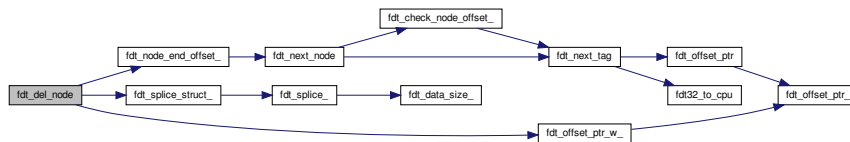
```
int fdt_del_node (
    void * fdt,
    int nodeoffset )
```

fdt_del_node - delete a node (subtree) : pointer to the device tree blob : offset of the node to nop

`fdt_del_node()` will remove the given node, including all its subnodes if any, from the blob.

This function will delete data from the blob, and will therefore change the offsets of some existing nodes.

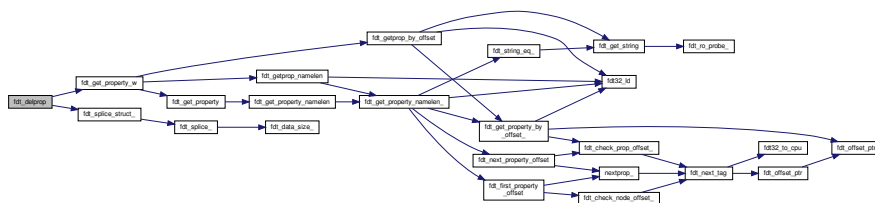
returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



21.90.2.19 fdt_delprop()

```
int fdt_delprop (
    void * fdt,
    int nodeoffset,
    const char * name )
```

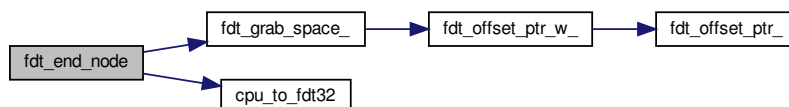
Here is the call graph for this function:



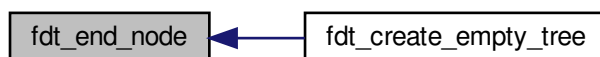
21.90.2.20 fdt_end_node()

```
int fdt_end_node (
    void * fdt )
```

Here is the call graph for this function:



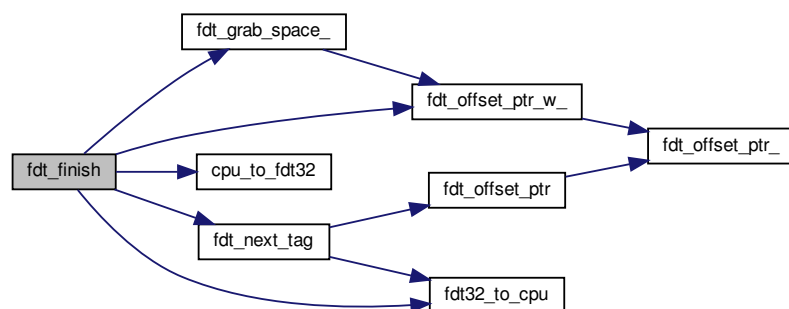
Here is the caller graph for this function:



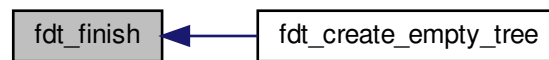
21.90.2.21 fdt_finish()

```
int fdt_finish (
    void * fdt )
```

Here is the call graph for this function:



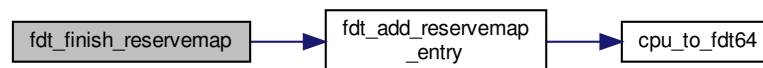
Here is the caller graph for this function:



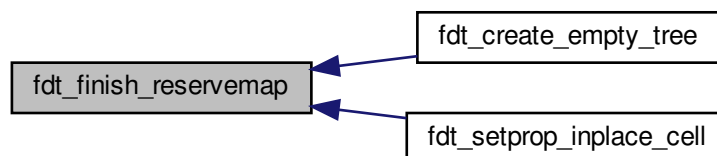
21.90.2.22 fdt_finish_reservemap()

```
int fdt_finish_reservemap (  
    void * fdt )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.90.2.24 fdt_first_subnode()

```
int fdt_first_subnode (
    const void * fdt,
    int offset )
```

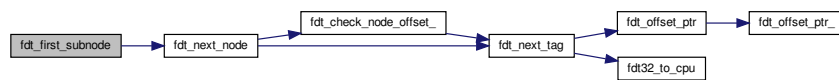
`fdt_first_subnode()` - get offset of first direct subnode

: FDT blob : Offset of node to check

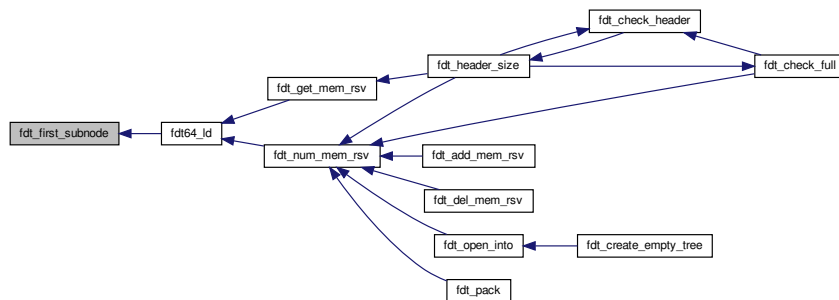
Returns

offset of first subnode, or -FDT_ERR_NOTFOUND if there is none

Here is the call graph for this function:



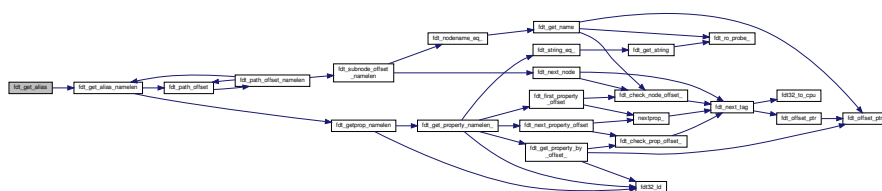
Here is the caller graph for this function:



21.90.2.25 fdt_get_alias()

```
const char* fdt_get_alias (
    const void * fdt,
    const char * name )
```

Here is the call graph for this function:



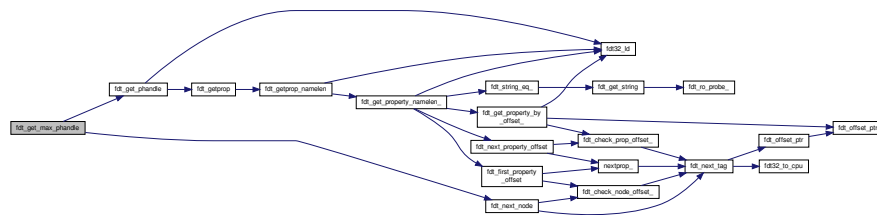
21.90.2.27 fdt_get_max_phandle()

```
uint32_t fdt_get_max_phandle (
    const void * fdt )
```

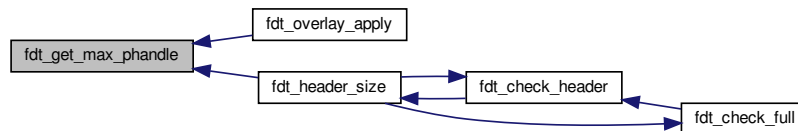
`fdt_get_max_phandle` - retrieves the highest phandle in a tree : pointer to the device tree blob

`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.

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:



Here is the caller graph for this function:



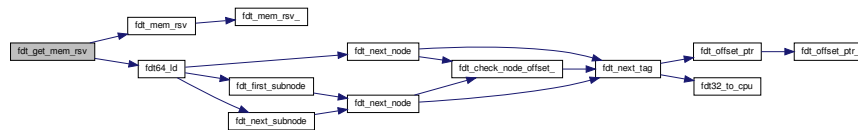
21.90.2.28 fdt_get_mem_rsv()

```
int fdt_get_mem_rsv (
    const void * fdt,
    int n,
    uint64_t * address,
    uint64_t * size )
```

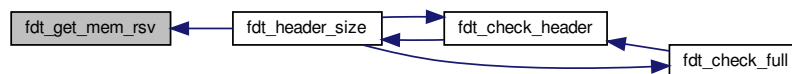
fdt_get_mem_rsv - retrieve one memory reserve map entry : pointer to the device tree blob , : pointers to 64-bit variables

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:



Here is the caller graph for this function:



21.90.2.29 fdt_get_name()

```

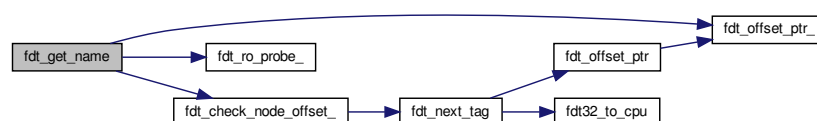
const char* fdt_get_name (
    const void * fdt,
    int nodeoffset,
    int * lenp )

```

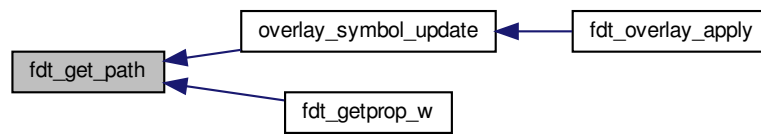
fdt_get_name - retrieve the name of a given node : pointer to the device tree blob : structure block offset of the starting node : pointer to an integer variable (will be overwritten) or NULL

fdt_get_name() retrieves the name (including unit address) of the device tree node at structure block offset nodeoffset. If lenp is non-NULL, the length of this name is also returned, in the integer pointed to by lenp.

returns: pointer to the node's name, on success If lenp is non-NULL, *lenp contains the length of that name (≥ 0) NULL, on error if lenp is non-NULL *lenp contains an error code (< 0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



21.90.2.31 fdt_get_phandle()

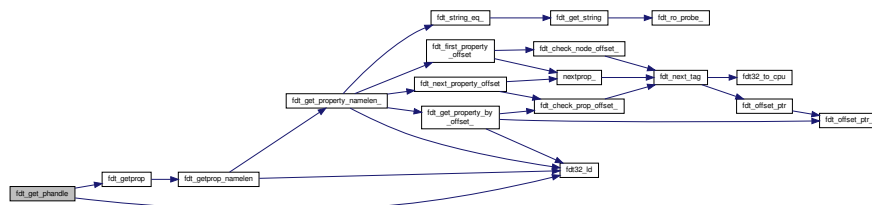
```

uint32_t fdt_get_phandle (
    const void * fdt,
    int nodeoffset )
  
```

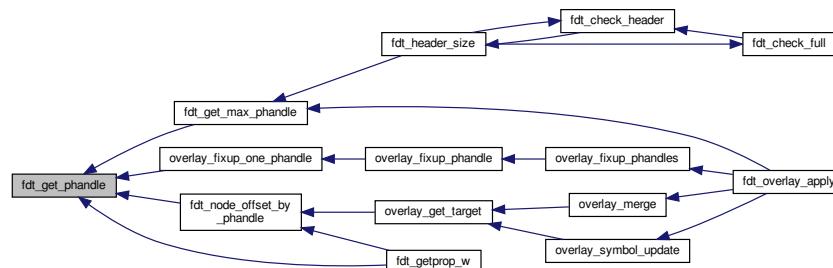
fdt_get_phandle - retrieve the phandle of a given node : pointer to the device tree blob : structure block offset of the node

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:



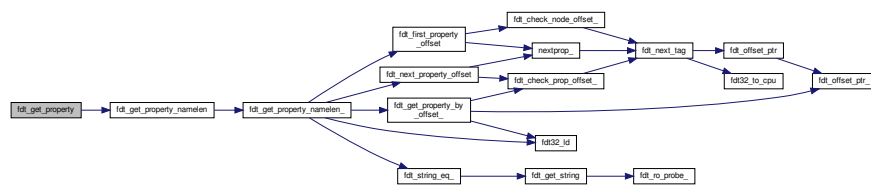
Here is the caller graph for this function:



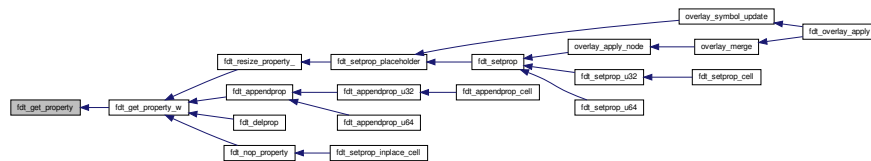
21.90.2.32 fdt_get_property()

```
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:



21.90.2.33 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 : pointer to the device tree blob : offset of the property to retrieve : pointer to an integer variable (will be overwritten) or NULL

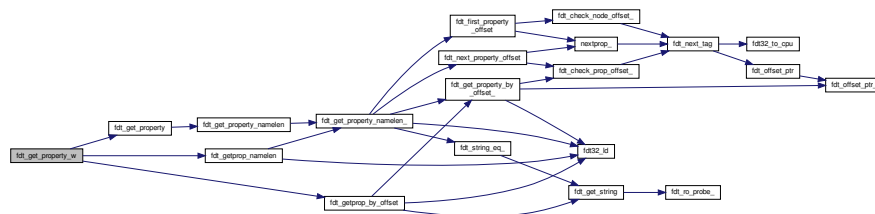
`fdt_get_property_by_offset()` retrieves a pointer to the `fdt_property` structure within the device tree blob at the given offset. If `lenp` is non-NULL, the length of the property value is also returned, in the integer pointed to by `lenp`.

Note that this code only works on device tree versions ≥ 16 . `fdt_getprop()` works on all versions.

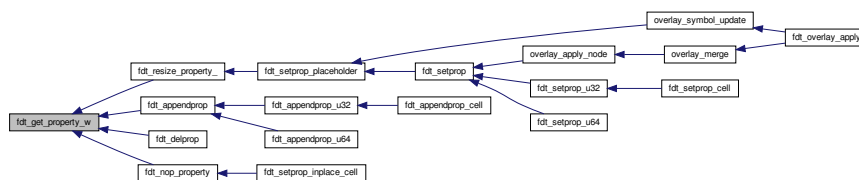
21.90.2.35 fdt_get_property_w()

```
static struct fdt_property* fdt_get_property_w (
    void * fdt,
    int nodeoffset,
    const char * name,
    int * lenp ) [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.90.2.36 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 : pointer to the device tree blob : offset of the string within the strings block (native endian) : 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:



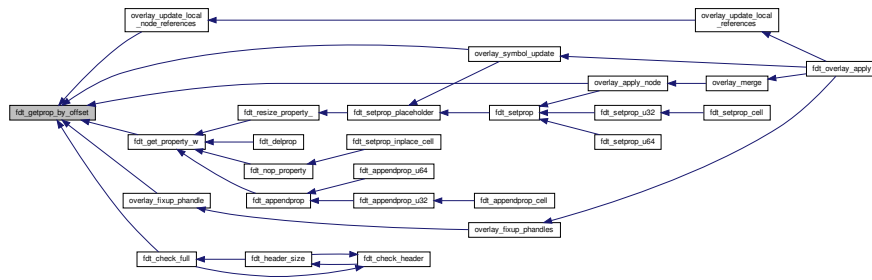

```
const void* fdt_getprop_by_offset (
    const void * fdt,
    int offset,
    const char ** namep,
    int * lenp )
```

`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 name 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).

```

graph LR
    Entry[fdt_getprop_by_offset] --> GetProp[fdt_get_property_by_offset]
    Entry --> GetId[fdt32_id]
    Entry --> GetString[fdt_get_string]
    GetProp --> CheckProp[fdt_check_prop_offset_]
    GetProp --> NextTag[fdt_next_tag]
    GetProp --> ToCpu[fdt32_to_cpu]
    CheckProp --> NextTag
    NextTag --> ToCpu
    NextTag --> OffsetPtr1[fdt_offset_ptr]
    ToCpu --> OffsetPtr2[fdt_offset_ptr]
    OffsetPtr1 --> Exit[fdt_offset_ptr]
    OffsetPtr2 --> Exit
    GetString --> Probe[fdt_ro_probe_]
  
```

Here is the caller graph for this function:



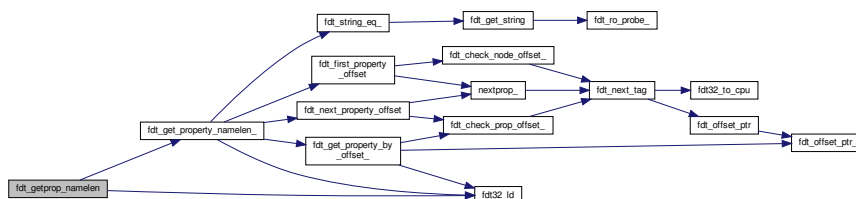
21.90.2.39 fdt_getprop_namelen()

```

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:



[illegible]

```
static void* fdt_getprop_namelen_w (
    void * fdt,
    int nodeoffset,
    const char * name,
    int namelen,
    int * lenp ) [inline], [static]
```

```

graph LR
    fdt_setprop[fdt_setprop] --> fdt_setprop_inplace_namestr_partial[fdt_setprop_inplace_namestr_partial]
    fdt_setprop --> fdt_setprop_inplace[fdt_setprop_inplace]
    fdt_setprop_inplace_namestr_partial --> overlay_update_local_node_references[overlay_update_local_node_references]
    overlay_update_local_node_references --> overlay_fixup_one_phandle[overlay_fixup_one_phandle]
    fdt_setprop_inplace --> fdt_setprop_inplace_u32[fdt_setprop_inplace_u32]
    fdt_setprop_inplace_u32 --> fdt_setprop_inplace_u64[fdt_setprop_inplace_u64]
    fdt_setprop_inplace_u64 --> fdt_setprop_inplace_cell[fdt_setprop_inplace_cell]
    fdt_setprop_inplace_cell --> overlay_phandle_add_offset[overlay_phandle_add_offset]
    overlay_phandle_add_offset --> overlay_adjust_node_phandles[overlay_adjust_node_phandles]
    overlay_adjust_node_phandles --> overlay_fixup_phandles[overlay_fixup_phandles]
    overlay_fixup_phandles --> fdt_overlay_apply[fdt_overlay_apply]
    fdt_overlay_apply --> overlay_fixup_phandles
    overlay_fixup_phandles --> overlay_update_local_references
    overlay_update_local_references --> fdt_setprop_namestr_w[fdt_setprop_namestr_w]
  
```

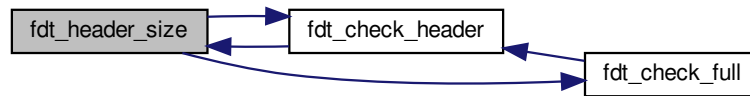
```
static void* fdt_getprop_w (
    void * fdt,
    int nodeoffset,
    const char * name,
    int * lenp ) [inline], [static]
```

The graph illustrates a complex network of relationships between various nodes. The nodes are labeled with identifiers and names, such as '100 get_offset', '100 node', '100 offset', and '100 in'. The edges represent directed connections between these nodes, forming a dense web of dependencies or relationships. The graph is organized into several layers, with nodes on the left side (e.g., '100 get_offset') acting as sources and nodes on the right side (e.g., '100 in') acting as sinks. The central part of the graph contains a large number of intermediate nodes, each representing a specific concept or entity. The edges are directed, indicating the flow of information or the direction of the relationship between the nodes. The overall structure of the graph suggests a hierarchical or sequential process, with nodes at the top representing higher-level concepts and nodes at the bottom representing more specific details or actions.

```
static size_t fdt_header_size (
    const void * fdt ) [inline], [static]
```

[illegible]

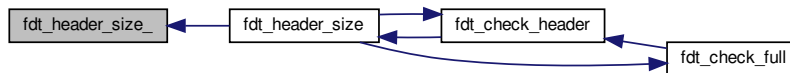
Here is the caller graph for this function:



21.90.2.43 fdt_header_size_()

```
size_t fdt_header_size_ (
    uint32_t version )
```

`fdt_header_size` - return the size of the tree's header : pointer to a flattened device tree Here is the caller graph for this function:



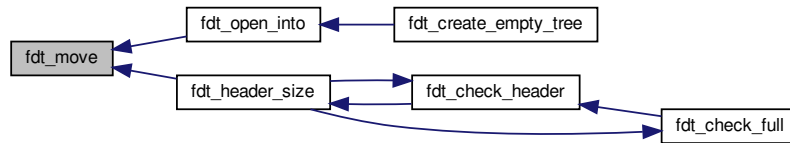
21.90.2.44 fdt_move()

```
int fdt_move (
    const void * fdt,
    void * buf,
    int bufsize )
```

`fdt_move` - move a device tree around in memory : pointer to the device tree to move : pointer to memory where the device is to be moved : size of the memory space at buf

`fdt_move()` relocates, if possible, the device tree blob located at `fdt` to the buffer at `buf` of size `bufsize`. The buffer may overlap with the existing device tree blob at `fdt`. Therefore, `fdt_move(fdt, fdt, fdt_totalsize(fdt))` should always succeed.

returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient to contain the device tree -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the caller graph for this function:



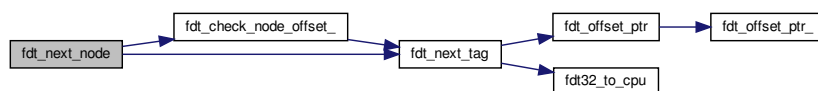
21.90.2.45 fdt_next_node()

```

int fdt_next_node (
    const void * fdt,
    int offset,
    int * depth )

```

Here is the call graph for this function:



[illegible]

```
int fdt_next_property_offset (
    const void * fdt,
    int offset )
```

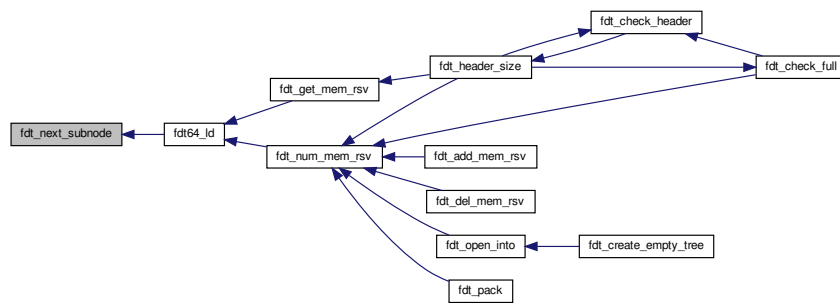
`fdt_next_property_offset()` finds the property immediately after the one at the given structure block offset. This will be a property of the same node as the given property.

```

graph LR
    A[fdt_next_property__offset] --> B[fdt_check_prop_offset_]
    A --> C[nextprop_]
    B --> D[fdt_next_tag]
    C --> D
    D --> E[fdt_offset_ptr]
    D --> F[fdt32_to_cpu]
    E --> G[fdt_offset_ptr_]

```


Here is the caller graph for this function:



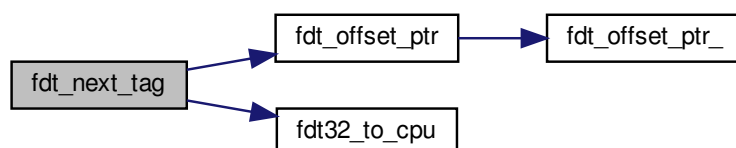
21.90.2.48 fdt_next_tag()

```

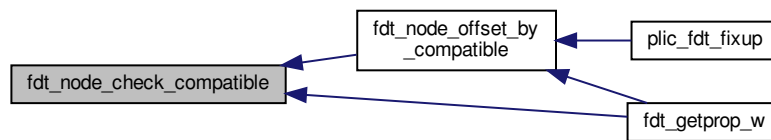
uint32_t fdt_next_tag (
    const void * fdt,
    int offset,
    int * nextoffset )

```

Here is the call graph for this function:



Here is the caller graph for this function:



21.90.2.50 fdt_node_depth()

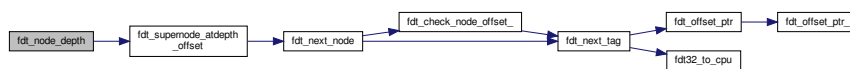
```
int fdt_node_depth (
    const void * fdt,
    int nodeoffset )
```

fdt_node_depth - find the depth of a given node : pointer to the device tree blob : offset of the node whose parent to find

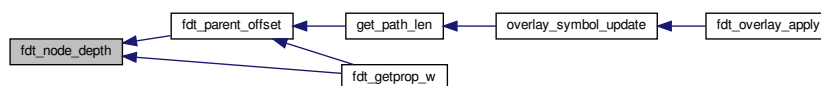
fdt_node_depth() finds the depth of a given node. The root node has depth 0, its immediate subnodes depth 1 and so forth.

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:




```
int fdt_node_offset_by_phandle (
    const void * fdt,
    uint32_t phandle )
```

`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.

[illegible]

```

graph LR
    fdt_node_offset_by_phandle[fdt_node_offset_by_phandle]
    overlay_get_target[overlay_get_target]
    overlay_merge[overlay_merge]
    overlay_symbol_update[overlay_symbol_update]
    fdt_overlay_apply[fdt_overlay_apply]
    fdt_getprop_w[fdt_getprop_w]

    fdt_node_offset_by_phandle --> overlay_get_target
    overlay_get_target --> overlay_merge
    overlay_get_target --> overlay_symbol_update
    overlay_merge --> fdt_overlay_apply
    overlay_symbol_update --> fdt_overlay_apply
    fdt_getprop_w --> fdt_node_offset_by_phandle

```

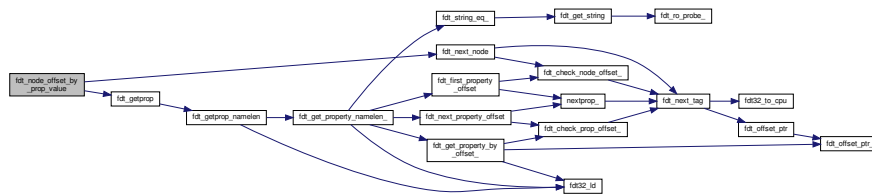
```
int fdt_node_offset_by_prop_value (
    const void * fdt,
    int startoffset,
    const char * propname,
    const void * propval,
    int proplen )
```

`find_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.

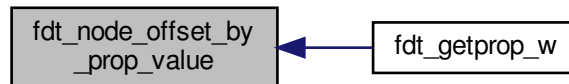
To iterate through all nodes matching the criterion, the following idiom can be used: `offset = fdt_node_offset_by_prop_value(fdt, -1, propname, propval, proplen); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_offset_by_prop_value(fdt, offset, propname, propval, proplen); }`

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 , $> \text{startoffset}$), 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:



21.90.2.54 fdt_nop_node()

```

int fdt_nop_node (
    void * fdt,
    int nodeoffset )

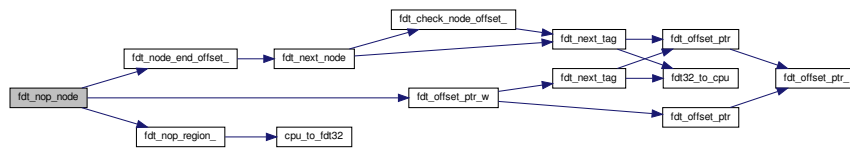
```

`fdt_nop_node` - replace a node (subtree) with nop tags : pointer to the device tree blob : offset of the node to nop

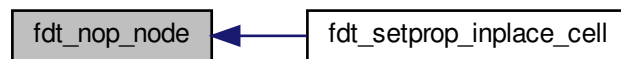
`fdt_nop_node()` will replace a given node's representation in the blob, including all its subnodes, if any, with FDT_NOP tags, effectively removing it from the tree.

This function will alter only the bytes in the blob which contain the node and its properties and subnodes, and will not alter or move any other part of the tree.

returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



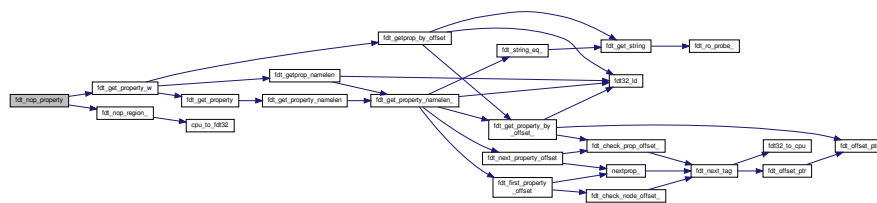
Here is the caller graph for this function:



21.90.2.55 fdt_nop_property()

```
int fdt_nop_property (
    void * fdt,
    int nodeoffset,
    const char * name )
```

Here is the call graph for this function:



Here is the caller graph for this function:



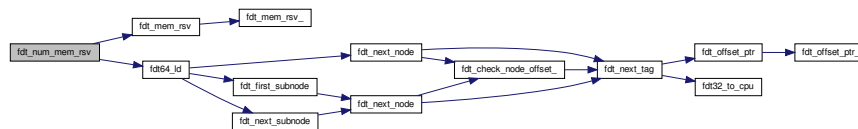
21.90.2.56 fdt_num_mem_rsv()

```
int fdt_num_mem_rsv (
    const void * fdt )
```

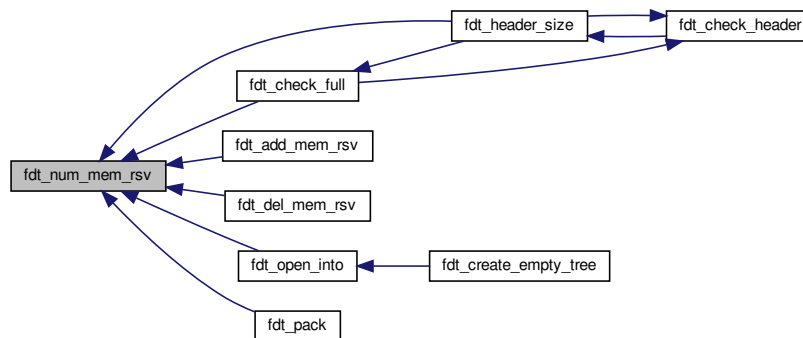
fdt_num_mem_rsv - retrieve the number of memory reserve map entries : pointer to the device tree blob

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:



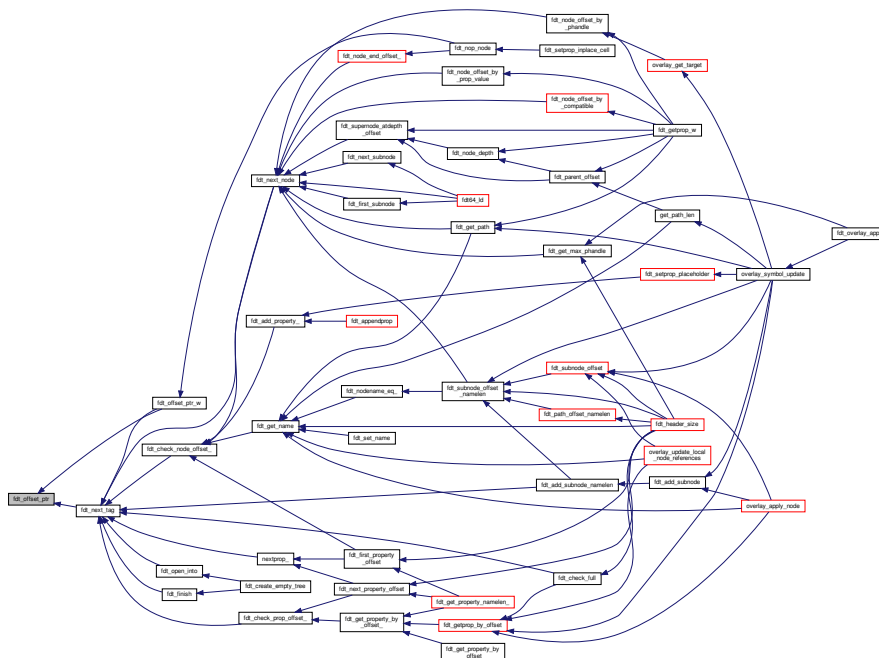
Here is the caller graph for this function:



21.90.2.57 fdt_offset_ptr()

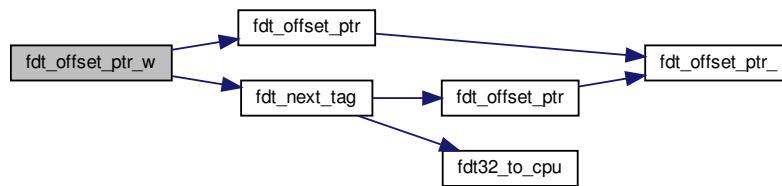
```
const void* fdt_offset_ptr (
    const void * fdt,
    int offset,
    unsigned int checklen )
```

Here is the caller graph for this function:

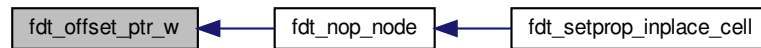


```
static void* fdt_offset_ptr_w (
    void * fdt,
    int offset,
    int checklen ) [inline], [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



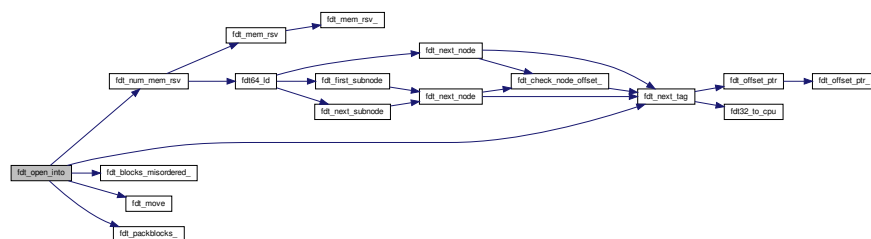
21.90.2.59 fdt_open_into()

```

int fdt_open_into (
    const void * fdt,
    void * buf,
    int bufsize )

```

Here is the call graph for this function:



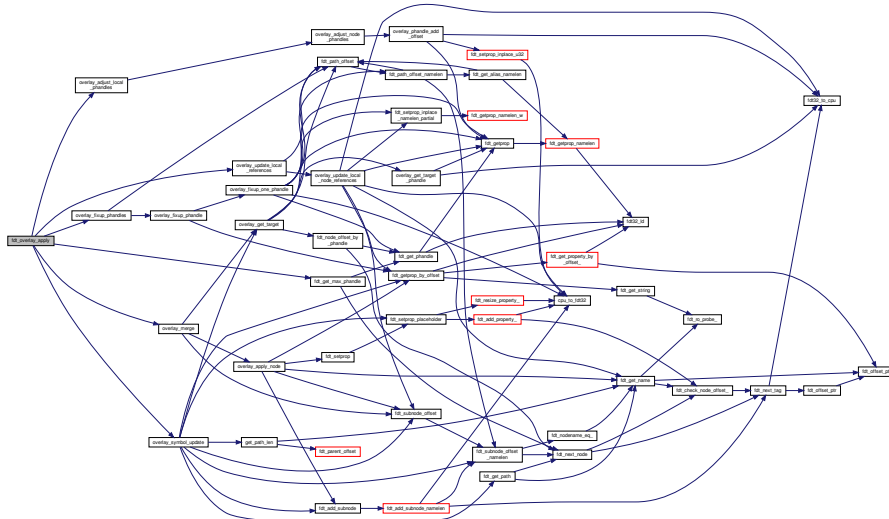
Here is the caller graph for this function:




```
int fdt_overlay_apply (
    void * fdt,
    void * fdto )
```

`fdt_overlay_apply()` will apply the given device tree overlay on the given base device tree.

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:



```
int fdt_pack (
    void * fdt )
```

[illegible]

21.90.2.62 fdt_parent_offset()

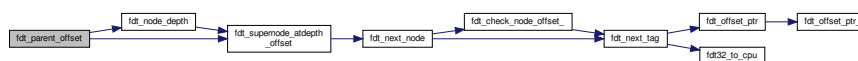
```
int fdt_parent_offset (
    const void * fdt,
    int nodeoffset )
```

fdt_parent_offset - find the parent of a given node : pointer to the device tree blob : offset of the node whose parent to find

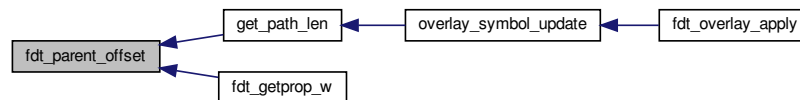
fdt_parent_offset() locates the parent node of a given node (that is, it finds the offset of the node which contains the node at nodeoffset as a subnode).

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset, *twice*.

returns: structure block offset of the parent of the node at nodeoffset (≥ 0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:

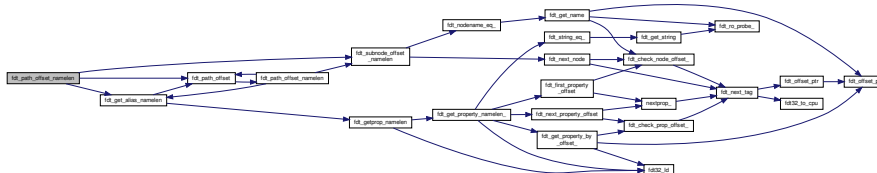
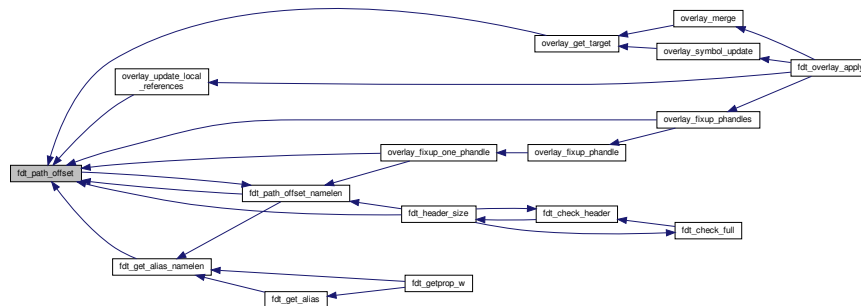
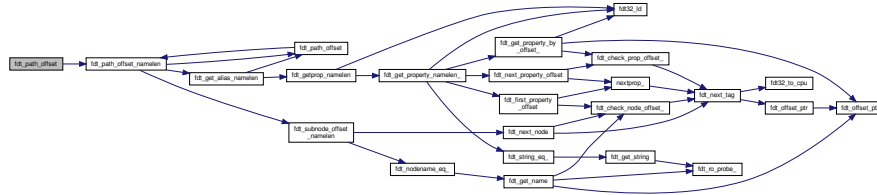


21.90.2.63 fdt_path_offset()

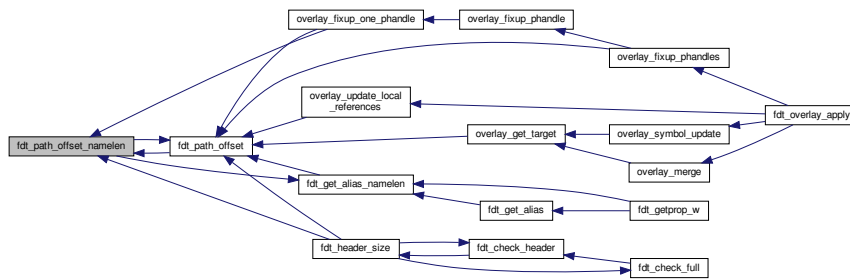
```
int fdt_path_offset (
    const void * fdt,
    const char * path )
```

fdt_path_offset - find a tree node by its full path : pointer to the device tree blob : 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).



Here is the caller graph for this function:



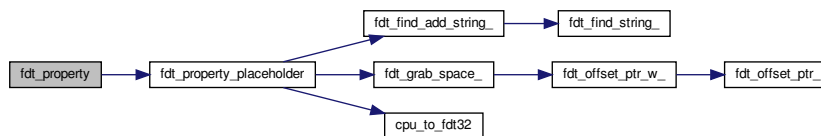
21.90.2.65 fdt_property()

```

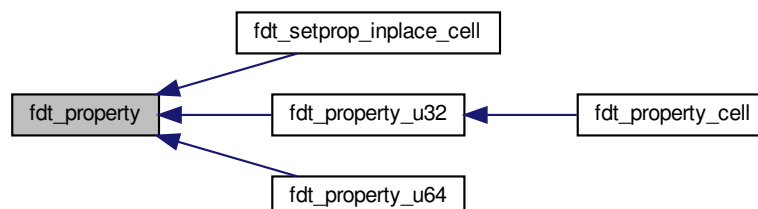
int fdt_property (
    void * fdt,
    const char * name,
    const void * val,
    int len )

```

Here is the call graph for this function:



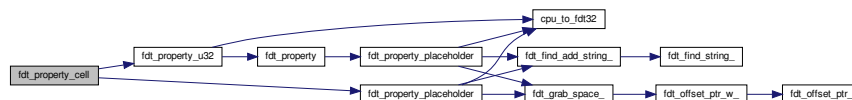
Here is the caller graph for this function:



21.90.2.66 fdt_property_cell()

```
static int fdt_property_cell (
    void * fdt,
    const char * name,
    uint32_t val ) [inline], [static]
```

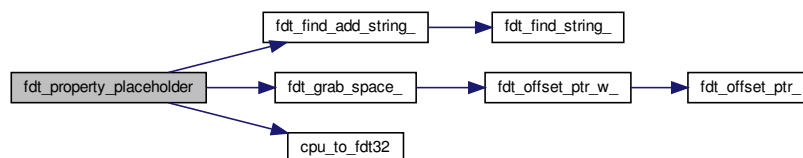
Here is the call graph for this function:



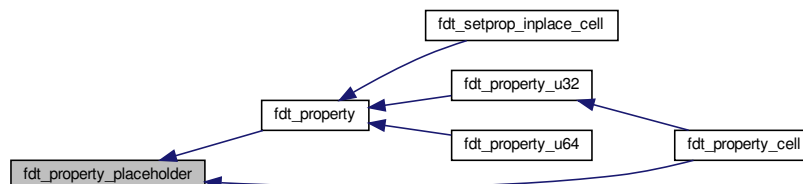
21.90.2.67 fdt_property_placeholder()

```
int fdt_property_placeholder (
    void * fdt,
    const char * name,
    int len,
    void ** valp )
```

Here is the call graph for this function:



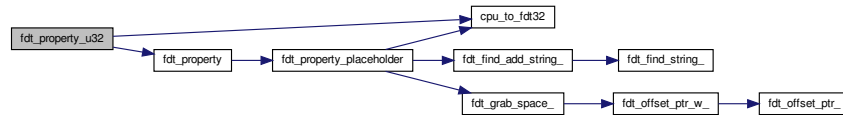
Here is the caller graph for this function:



21.90.2.68 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:



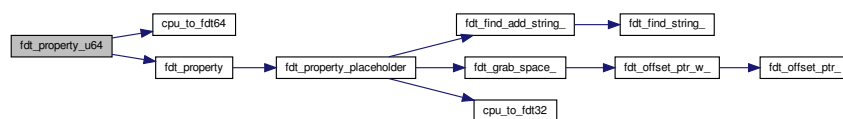
Here is the caller graph for this function:



21.90.2.69 fdt_property_u64()

```
static int fdt_property_u64 (
    void * fdt,
    const char * name,
    uint64_t val ) [inline], [static]
```

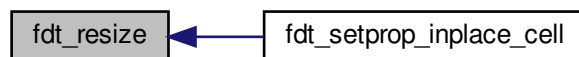
Here is the call graph for this function:



21.90.2.70 fdt_resize()

```
int fdt_resize (
    void * fdt,
    void * buf,
    int bufsize )
```

Here is the caller graph for this function:



21.90.2.71 fdt_set_hdr_() [1/10]

```
fdt_set_hdr_ (
    magic )
```

21.90.2.72 fdt_set_hdr_() [2/10]

```
fdt_set_hdr_ (
    totalsize )
```

21.90.2.73 fdt_set_hdr_() [3/10]

```
fdt_set_hdr_ (
    off_dt_struct )
```

21.90.2.74 fdt_set_hdr_() [4/10]

```
fdt_set_hdr_ (
    off_dt_strings )
```

21.90.2.75 fdt_set_hdr_() [5/10]

```
fdt_set_hdr_ (
    off_mem_rsvmap )
```

21.90.2.76 fdt_set_hdr_() [6/10]

```
fdt_set_hdr_ (
    version )
```

21.90.2.77 fdt_set_hdr_() [7/10]

```
fdt_set_hdr_ (
    last_comp_version )
```

21.90.2.78 fdt_set_hdr_() [8/10]

```
fdt_set_hdr_ (
    boot_cpuid_phys )
```

21.90.2.79 fdt_set_hdr_() [9/10]

```
fdt_set_hdr_ (
    size_dt_strings )
```

21.90.2.80 fdt_set_hdr_() [10/10]

```
fdt_set_hdr_ (
    size_dt_struct )
```



```
int fdt_set_name (
    void * fdt,
    int nodeoffset,
    const char * name )
```

```

graph LR
    fdt_set_name[fdt_set_name] --> fdt_get_name[fdt_get_name]
    fdt_set_name --> fdt_splice_struct[fdt_splice_struct_]
    fdt_get_name --> fdt_probe[fdt_ro_probe_]
    fdt_get_name --> fdt_check_node_offset[fdt_check_node_offset_]
    fdt_get_name --> fdt_offset_ptr[fdt_offset_ptr_]
    fdt_splice_struct --> fdt_splice[fdt_splice_]
    fdt_splice_struct --> fdt_offset_ptr
    fdt_splice --> fdt_data_size[fdt_data_size_]
    fdt_check_node_offset --> fdt_next_tag[fdt_next_tag]
    fdt_next_tag --> fdt32_to_cpu[fdt32_to_cpu]
    fdt32_to_cpu --> fdt_offset_ptr
    fdt_offset_ptr --> fdt_offset_ptr_2[fdt_offset_ptr_]

```

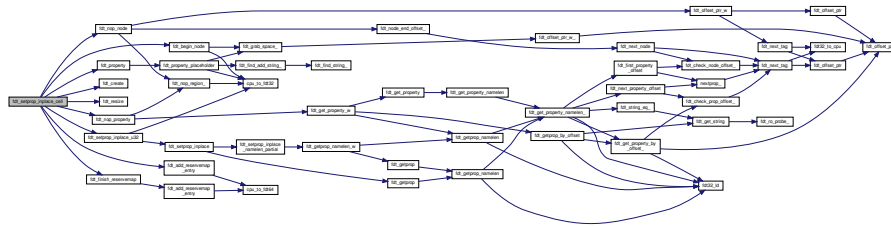
```
int fdt_setprop (
    void * fdt,
    int nodeoffset,
    const char * name,
    const void * val,
    int len )
```

```
graph RL
    fdt_overlay_apply --> overlay_merge
    overlay_merge --> overlay_apply_node
    fdt_setprop_cell --> fdt_setprop_u32
    fdt_setprop_u32 --> fdt_setprop
    fdt_setprop_u64 --> fdt_setprop
```



```
static int fdt_setprop_inplace_cell (
    void * fdt,
    int nodeoffset,
    const char * name,
    uint32_t val ) [inline], [static]
```

This is an alternative name for `fdt_setprop_inplace_u32()` Here is the call graph for this function:



```
int fdt_setprop_inplace_namelen_partial (
    void * fdt,
    int nodeoffset,
    const char * name,
    int namelen,
    uint32_t idx,
    const void * val,
    int len )
```

[illegible][illegible]


```
int fdt_setprop_placeholder (
    void * fdt,
    int nodeoffset,
    const char * name,
    int len,
    void ** prop_data )
```

[illegible]

```
graph LR
    fdt_overlay_apply[fdt_overlay_apply] --> overlay_symbol_update[overlay_symbol_update]
    fdt_overlay_apply --> overlay_merge[overlay_merge]
    overlay_merge --> overlay_apply_node[overlay_apply_node]
    overlay_apply_node --> fdt_setprop[fdt_setprop]
    fdt_setprop_cell[fdt_setprop_cell] --> fdt_setprop_u32[fdt_setprop_u32]
    fdt_setprop_u32 --> fdt_setprop
    fdt_setprop_u64[fdt_setprop_u64] --> fdt_setprop
    fdt_setprop --> fdt_setprop_placeholder[fdt_setprop_placeholder]
    fdt_setprop --> overlay_symbol_update
```

```
static int fdt_setprop_u32 (
    void * fdt,
    int nodeoffset,
    const char * name,
    uint32_t val ) [inline], [static]
```

[illegible]

Here is the caller graph for this function:

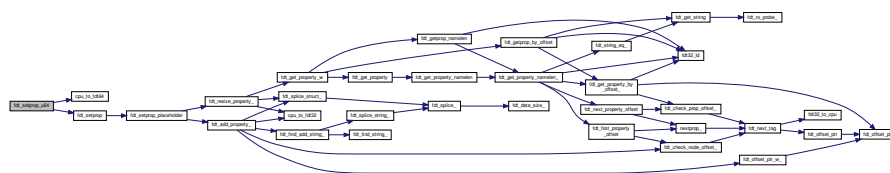


21.90.2.91 fdt_setprop_u64()

```

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:



21.90.2.92 fdt_size_cells()

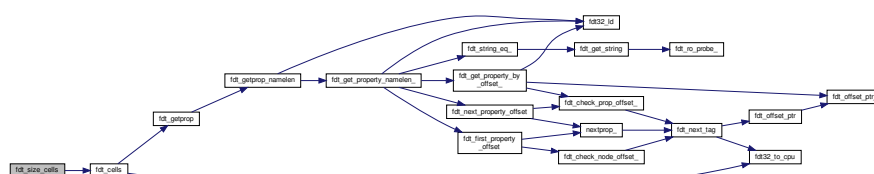
```

int fdt_size_cells (
    const void * fdt,
    int nodeoffset )
  
```

fdt_size_cells - retrieve address range size for a bus represented in the tree : pointer to the device tree blob : offset of the node to find the address range size for

When the node has a valid #size-cells property, returns its value.

returns: 0 <= n < FDT_MAX_NCELLS, on success 1, if the node has no #size-cells property -FDT_ERR_BADNCELLS, if the node has a badly formatted or invalid #size-cells property -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



21.90.2.93 fdt_strerror()

```
const char* fdt_strerror (
    int errval )
```

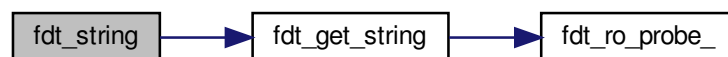
21.90.2.94 fdt_string()

```
const char* fdt_string (
    const void * fdt,
    int stroffset )
```

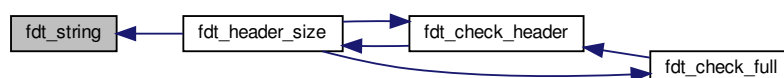
`fdt_string` - retrieve a string from the strings block of a device tree : pointer to the device tree blob : offset of the string within the strings block (native endian)

[fdt_string\(\)](#) retrieves a pointer to a single string from the strings block of the device tree blob at `fdt`.

returns: a pointer to the string, on success NULL, if `stroffset` is out of bounds, or doesn't point to a valid string Here is the call graph for this function:



Here is the caller graph for this function:



21.90.2.95 fdt_stringlist_contains()

```
int fdt_stringlist_contains (
    const char * strlist,
    int listlen,
    const char * str )
```

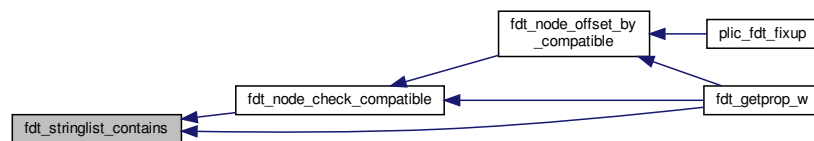
fdt_stringlist_contains - check a string list property for a string : Property containing a list of strings to check : Length of property : String to search for

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

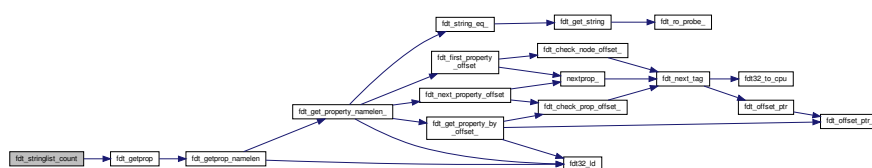
Here is the caller graph for this function:



21.90.2.96 fdt_stringlist_count()

```
int fdt_stringlist_count (
    const void * fdt,
    int nodeoffset,
    const char * property )
```

Here is the call graph for this function:



Here is the caller graph for this function:



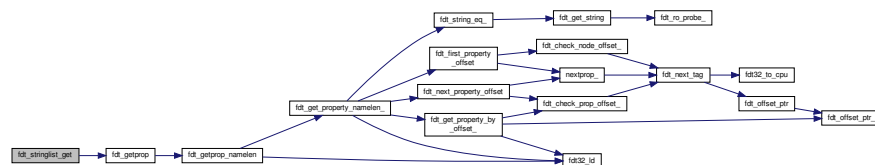
21.90.2.97 fdt_stringlist_get()

```

const char* fdt_stringlist_get (
    const void * fdt,
    int nodeoffset,
    const char * property,
    int index,
    int * lenp )

```

Here is the call graph for this function:



Here is the caller graph for this function:



```
int fdt_stringlist_search (
    const void * fdt,
    int nodeoffset,
    const char * property,
    const char * string )
```

[illegible]

```
graph LR; fdt_getprop_w --> fdt_stringlist_search
```

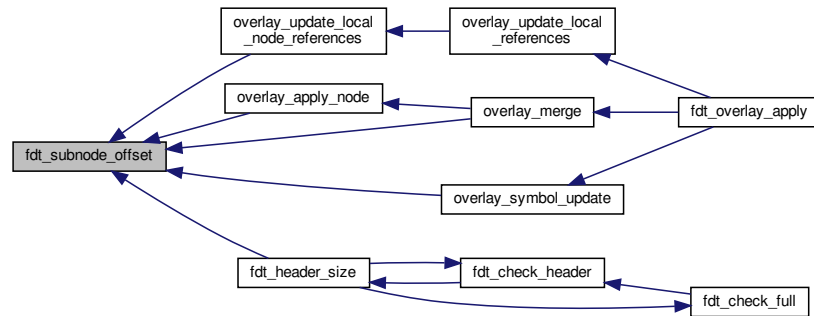
```
int fdt_subnode_offset (
    const void * fdt,
    int parentoffset,
    const char * name )
```

```

graph LR
    A[fdt_subnode_offset] --> B[fdt_subnode_offset_namelen]
    B --> C[fdt_next_node]
    B --> D[fdt_node_name_eq]
    C --> E[fdt_check_node_offset]
    D --> F[fdt_get_name]
    E --> G[fdt_next_tag]
    F --> G
    F --> H[fdt_no_probe]
    G --> I[fdt_offset_ptr]
    G --> J[fdt32_to_cpu]
    I --> K[fdt_offset_ptr]
    J --> K
  
```

The flowchart illustrates the logic of the `fdt_subnode_offset()` function. It begins with the initial state `fdt_subnode_offset`, which leads to `fdt_subnode_offset_namelen`. From there, the flow branches into two paths: one leading to `fdt_next_node` and another to `fdt_node_name_eq`. The `fdt_next_node` path leads to `fdt_check_node_offset`, which then leads to `fdt_next_tag`. The `fdt_node_name_eq` path leads to `fdt_get_name`, which also leads to `fdt_next_tag` and can also lead to `fdt_no_probe`. From `fdt_next_tag`, the flow branches into `fdt_offset_ptr` and `fdt32_to_cpu`, both of which lead to the final output `fdt_offset_ptr`.

Here is the caller graph for this function:



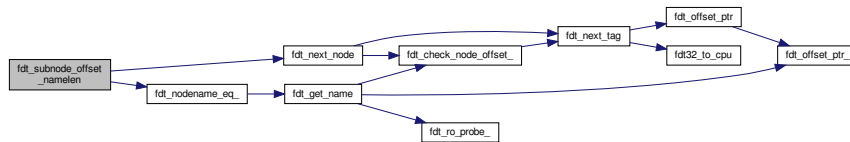
21.90.2.100 fdt_subnode_offset_namelen()

```

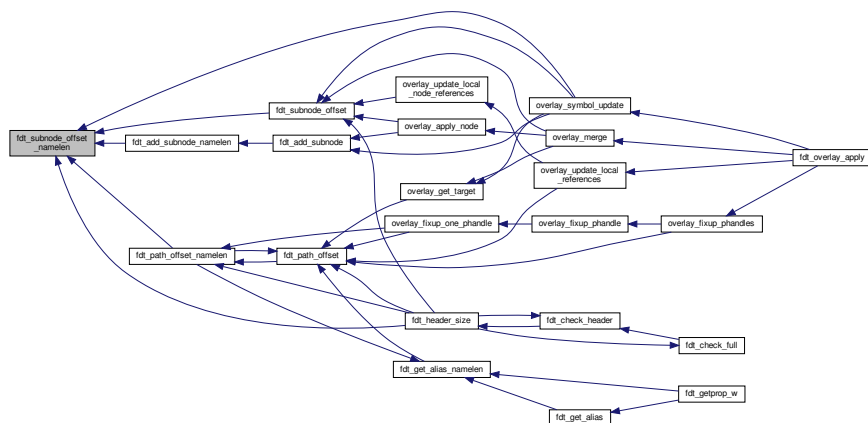
int fdt_subnode_offset_namelen (
    const void * fdt,
    int parentoffset,
    const char * name,
    int namelen )

```

Here is the call graph for this function:



Here is the caller graph for this function:



21.90.2.101 fdt_supernode_atdepth_offset()

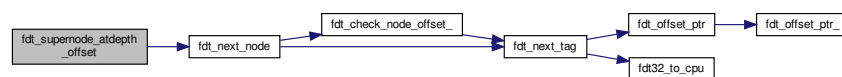
```
int fdt_supernode_atdepth_offset (
    const void * fdt,
    int nodeoffset,
    int supernodedepth,
    int * nodedepth )
```

fdt_supernode_atdepth_offset - find a specific ancestor of a node : pointer to the device tree blob : offset of the node whose parent to find : depth of the ancestor to find : pointer to an integer variable (will be overwritten) or NULL

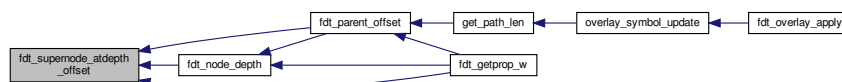
[fdt_supernode_atdepth_offset\(\)](#) finds an ancestor of the given node at a specific depth from the root (where the root itself has depth 0, its immediate subnodes depth 1 and so forth). So `fdt_supernode_atdepth_offset(fdt, nodeoffset, 0, NULL)`; will always return 0, the offset of the root node. 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:

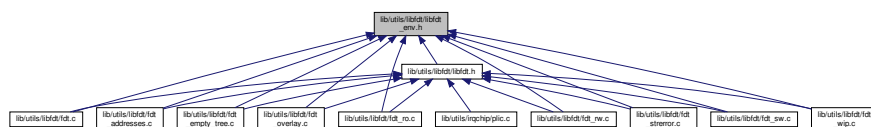
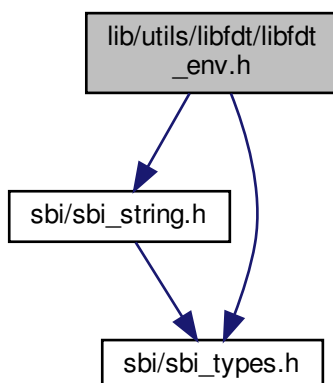


Here is the caller graph for this function:



21.91 lib/utils/libfdt/libfdt_env.h File Reference

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



```
• #define INT_MAX ((int)(- 0LL < 1))
```

- #define INT_MAX ((int)(~0U >> 1))
- #define UINT_MAX ((unsigned int)~0U)
- #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 strlen sbi_strlen
- #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)

21.91.1 Macro Definition Documentation

21.91.1.1 CPU_TO_FDT16

```
#define CPU_TO_FDT16(  
    x ) ((EXTRACT\_BYTE(x, 0) << 8) | EXTRACT\_BYTE(x, 1))
```

21.91.1.2 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))
```

21.91.1.3 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))
```

21.91.1.4 EXTRACT_BYTE

```
#define EXTRACT_BYTE(  
    x,  
    n ) ((unsigned long long)((uint8_t *)&x)[n])
```

21.91.1.5 FDT_BITWISE

```
#define FDT_BITWISE
```

21.91.1.6 FDT_FORCE

```
#define FDT_FORCE
```

21.91.1.7 INT_MAX

```
#define INT_MAX ((int)(~0U >> 1))
```

21.91.1.8 memchr

```
#define memchr sbi_memchr
```

21.91.1.9 memcmp

```
#define memcmp sbi_memcmp
```

21.91.1.10 memcpy

```
#define memcpy sbi_memcpy
```

21.91.1.11 memmove

```
#define memmove sbi_memmove
```

21.91.1.12 memset

```
#define memset sbi_memset
```

21.91.1.13 strchr

```
#define strchr sbi_strchr
```

21.91.1.14 strcmp

```
#define strcmp sbi_strcmp
```

21.91.1.15 strcpy

```
#define strcpy sbi_strcpy
```

21.91.1.16 strlen

```
#define strlen sbi_strlen
```

21.91.1.17 strlen

```
#define strlen sbi_strlen
```

21.91.1.18 strrchr

```
#define strrchr sbi_strrchr
```


21.91.1.19 UINT_MAX

```
#define UINT_MAX ((unsigned int)~0U)
```

21.91.2 Typedef Documentation

21.91.2.1 fdt16_t

```
typedef uint16_t FDT_BITWISE fdt16_t
```

21.91.2.2 fdt32_t

```
typedef uint32_t FDT_BITWISE fdt32_t
```

21.91.2.3 fdt64_t

```
typedef uint64_t FDT_BITWISE fdt64_t
```

21.91.3 Function Documentation

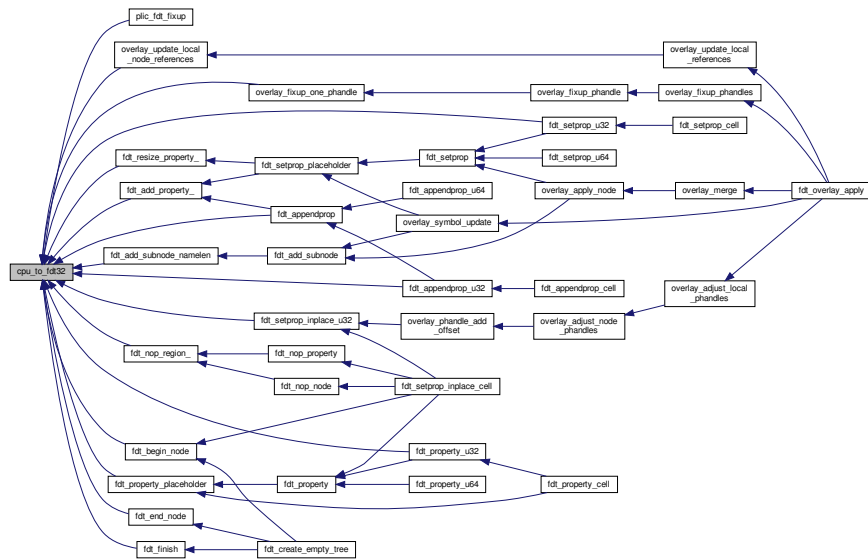
21.91.3.1 cpu_to_fdt16()

```
static fdt16_t cpu_to_fdt16 (  
    uint16_t x ) [inline], [static]
```

21.91.3.2 cpu_to_fdt32()

```
static fdt32_t cpu_to_fdt32 (
    uint32_t x ) [inline], [static]
```

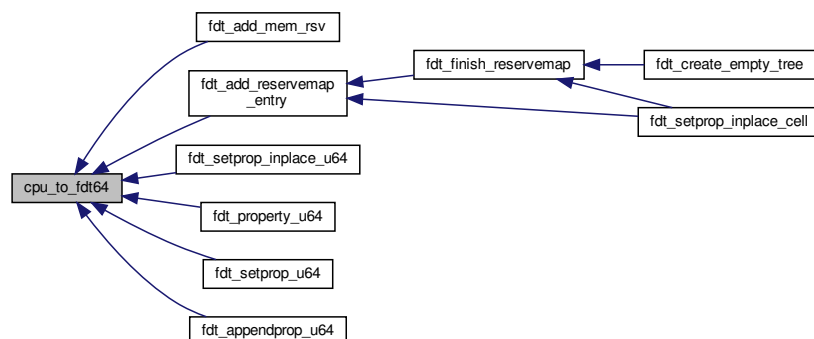
Here is the caller graph for this function:



21.91.3.3 cpu_to_fdt64()

```
static fdt64_t cpu_to_fdt64 (
    uint64_t x ) [inline], [static]
```

Here is the caller graph for this function:



```
static uint16_t fdt16_to_cpu (
    fdt16_t x ) [inline], [static]
```

```
static uint32_t fdt32_to_cpu (
    fdt32_t x )    [inline], [static]
```

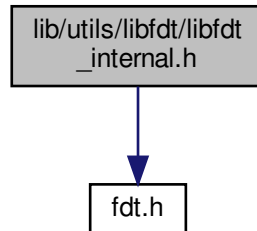
[illegible]

```
static uint64_t fdt64_to_cpu (
    fdt64_t x ) [inline], [static]
```

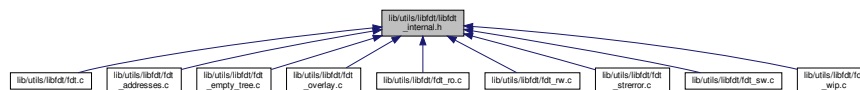
21.92 lib/utls/libfdt/libfdt_internal.h File Reference

```
#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)`

21.92.1 Macro Definition Documentation

21.92.1.1 FDT_ALIGN

```
#define FDT_ALIGN(  
    x,  
    a ) ((x) + (a) - 1) & ~((a) - 1)
```

21.92.1.2 FDT_RO_PROBE

```
#define FDT_RO_PROBE(  
    fdt )
```

Value:

```
{ \br/>    int err_; \br/>    if ((err_ = fdt_ro_probe_(fdt)) != 0) \br/>        return err_; \br/>}
```

21.92.1.3 FDT_SW_MAGIC

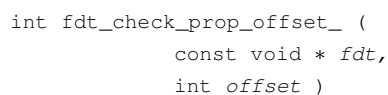
```
#define FDT_SW_MAGIC (~FDT_MAGIC)
```

21.92.1.4 FDT_TAGALIGN

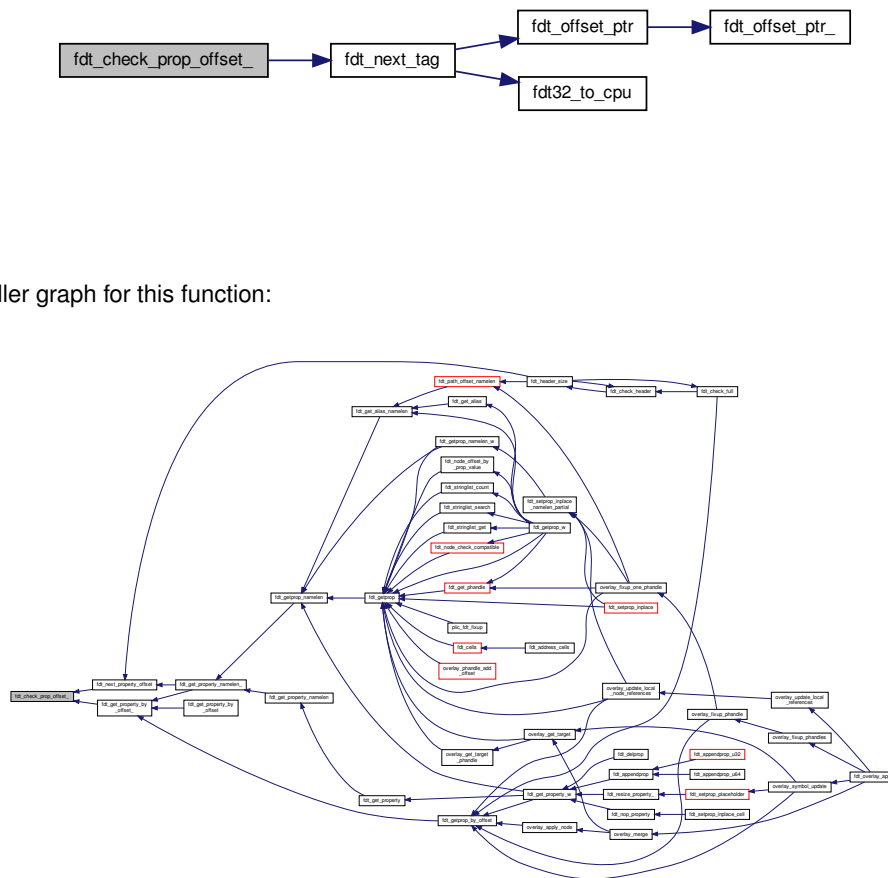
```
#define FDT_TAGALIGN(  
    x ) (FDT_ALIGN((x), FDT_TAGSIZE))
```

21.92.2 Function Documentation

```
int fdt_check_node_offset_ (
    const void * fdt,
    int offset )
```



Here is the caller graph for this function:



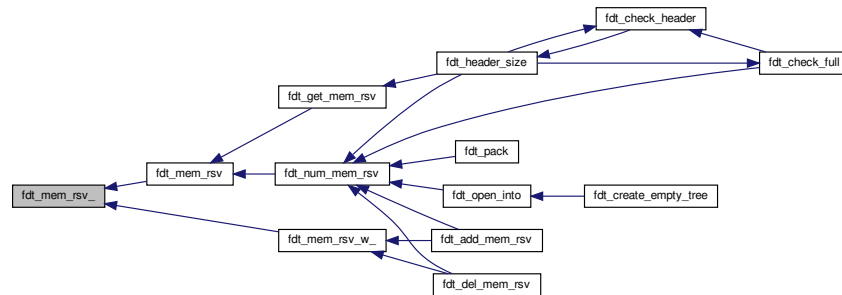
```
const char* fdt_find_string_ (
    const char * strtabs,
    int tabsize,
    const char * s )
```

[illegible]

21.92.2.4 fdt_mem_rsv_()

```
static const struct fdt_reserve_entry* fdt_mem_rsv_ (
    const void * fdt,
    int n ) [static]
```

Here is the caller graph for this function:



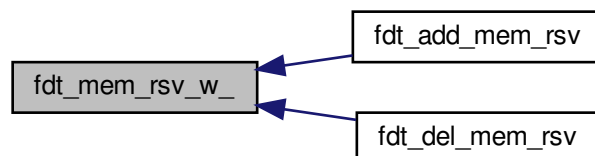
21.92.2.5 fdt_mem_rsv_w_()

```
static struct fdt_reserve_entry* fdt_mem_rsv_w_ (
    void * fdt,
    int n ) [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:




```
int fdt_node_end_offset_ (
    void * fdt,
    int nodeoffset )
```

```

graph LR
    A[fdt_node_end_offset_] --> B[fdt_next_node]
    B --> C[fdt_check_node_offset_]
    B --> D[fdt_next_tag]
    C --> D
    D --> E[fdt_offset_ptr]
    D --> F[fdt32_to_cpu]
    E --> G[fdt_offset_ptr_]

```

```
graph RL
    fdt_del_node --> fdt_node_end_offset_
    fdt_setprop_inplace_cell --> fdt_nop_node
    fdt_nop_node --> fdt_node_end_offset_
    style fdt_node_end_offset_ fill:#ccc,stroke:#333,stroke-width:1px
```

```
static const void* fdt_offset_ptr_ (
    const void * fdt,
    int offset ) [inline], [static]
```

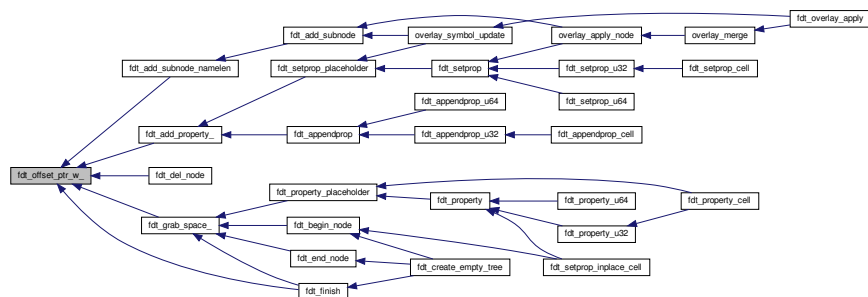
21.92.2.8 fdt_offset_ptr_w_()

```
static void* fdt_offset_ptr_w_ (
    void * fdt,
    int offset ) [inline], [static]
```

Here is the call graph for this function:



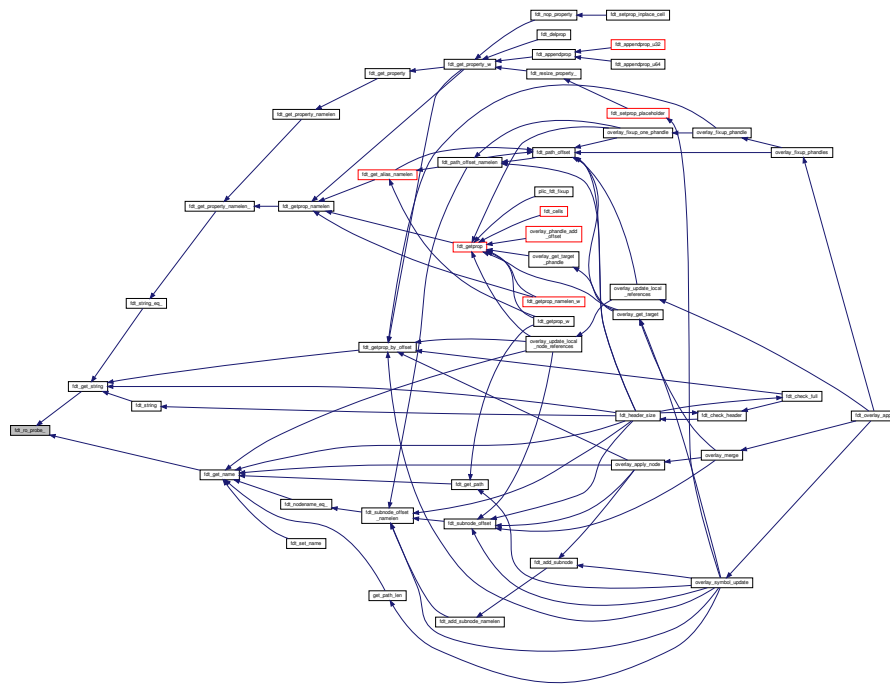
Here is the caller graph for this function:



21.92.2.9 fdt_ro_probe_()

```
int fdt_ro_probe_ (
    const void * fdt )
```

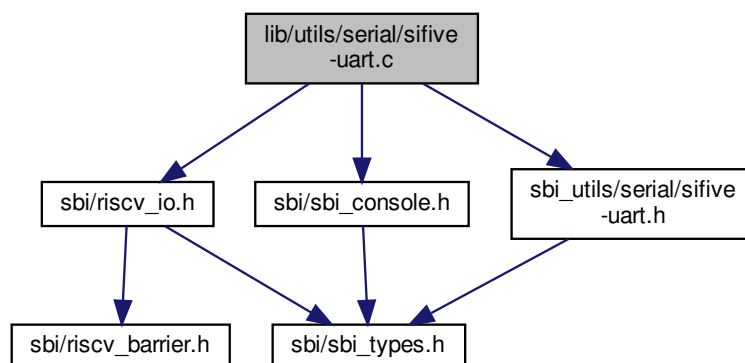
Here is the caller graph for this function:



21.93 lib/utls/serial/sifive-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utls/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`

21.93.1 Macro Definition Documentation

21.93.1.1 UART_REG_DIV

```
#define UART_REG_DIV 6
```

21.93.1.2 UART_REG_IE

```
#define UART_REG_IE 4
```

21.93.1.3 UART_REG_IP

```
#define UART_REG_IP 5
```

21.93.1.4 UART_REG_RXCTRL

```
#define UART_REG_RXCTRL 3
```

21.93.1.5 UART_REG_RXFIFO

```
#define UART_REG_RXFIFO 1
```

21.93.1.6 UART_REG_TXCTRL

```
#define UART_REG_TXCTRL 2
```

21.93.1.7 UART_REG_TXFIFO

```
#define UART_REG_TXFIFO 0
```

21.93.1.8 UART_RXCTRL_RXEN

```
#define UART_RXCTRL_RXEN 0x1
```

21.93.1.9 UART_RXFIFO_DATA

```
#define UART_RXFIFO_DATA 0x000000ff
```

21.93.1.10 UART_RXFIFO_EMPTY

```
#define UART_RXFIFO_EMPTY 0x80000000
```

21.93.1.11 UART_TXCTRL_TXEN

```
#define UART_TXCTRL_TXEN 0x1
```

21.93.1.12 UART_TXFIFO_FULL

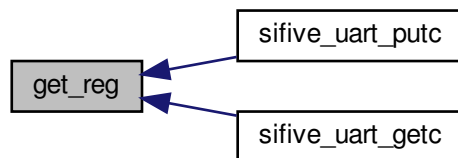
```
#define UART_TXFIFO_FULL 0x80000000
```

21.93.2 Function Documentation

21.93.2.1 get_reg()

```
static u32 get_reg (  
    u32 num ) [static]
```

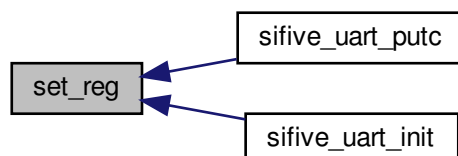
Here is the caller graph for this function:



21.93.2.2 set_reg()

```
static void set_reg (  
    u32 num,  
    u32 val ) [static]
```

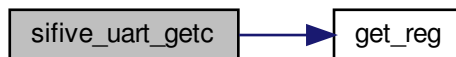
Here is the caller graph for this function:



21.93.2.3 sifive_uart_getc()

```
int sifive_uart_getc (  
    void )
```

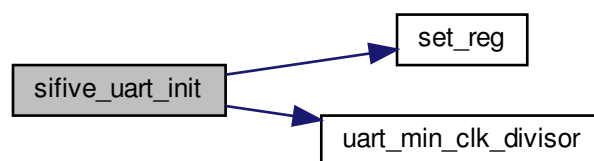
Here is the call graph for this function:



21.93.2.4 sifive_uart_init()

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

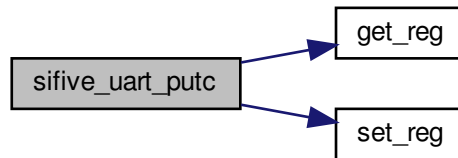
Here is the call graph for this function:



21.93.2.5 sifive_uart_putc()

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

Here is the call graph for this function:



21.93.2.6 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.

$f_baud = f_in / (div + 1) \Rightarrow div = (f_in / f_baud) - 1$ The nearest integer solution requires rounding up as to not exceed max_target_hz. $div = \text{ceil}(f_in / f_baud) - 1 = \text{floor}((f_in - 1 + f_baud) / f_baud) - 1$ 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:



21.93.3 Variable Documentation

21.93.3.1 uart_base

```

volatile void* uart_base [static]
  
```


21.93.3.2 uart_baudrate

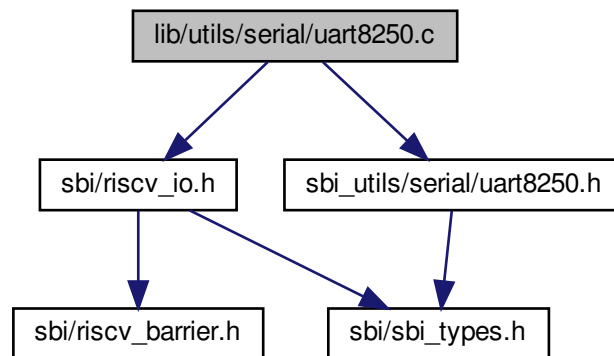
```
u32 uart_baudrate [static]
```

21.93.3.3 uart_in_freq

```
u32 uart_in_freq [static]
```

21.94 lib/utls/serial/uart8250.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi_utls/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`

21.94.1 Macro Definition Documentation

21.94.1.1 UART_DLL_OFFSET

```
#define UART_DLL_OFFSET 0 /* Out: Divisor Latch Low */
```

21.94.1.2 UART_DLM_OFFSET

```
#define UART_DLM_OFFSET 1 /* Out: Divisor Latch High */
```

21.94.1.3 UART_FCR_OFFSET

```
#define UART_FCR_OFFSET 2 /* Out: FIFO Control Register */
```

21.94.1.4 UART_IER_OFFSET

```
#define UART_IER_OFFSET 1 /* I/O: Interrupt Enable Register */
```

21.94.1.5 UART_IIR_OFFSET

```
#define UART_IIR_OFFSET 2 /* I/O: Interrupt Identification Register */
```

21.94.1.6 UART_LCR_OFFSET

```
#define UART_LCR_OFFSET 3 /* Out: Line Control Register */
```

21.94.1.7 UART_LSR_BI

```
#define UART_LSR_BI 0x10 /* Break interrupt indicator */
```

21.94.1.8 UART_LSR_BRK_ERROR_BITS

```
#define UART_LSR_BRK_ERROR_BITS 0x1E /* BI, FE, PE, OE bits */
```

21.94.1.9 UART_LSR_DR

```
#define UART_LSR_DR 0x01 /* Receiver data ready */
```

21.94.1.10 UART_LSR_FE

```
#define UART_LSR_FE 0x08 /* Frame error indicator */
```

21.94.1.11 UART_LSR_FIFOE

```
#define UART_LSR_FIFOE 0x80 /* Fifo error */
```

21.94.1.12 UART_LSR_OE

```
#define UART_LSR_OE 0x02 /* Overrun error indicator */
```

21.94.1.13 UART_LSR_OFFSET

```
#define UART_LSR_OFFSET 5 /* In: Line Status Register */
```

21.94.1.14 UART_LSR_PE

```
#define UART_LSR_PE 0x04 /* Parity error indicator */
```

21.94.1.15 UART_LSR_TEMT

```
#define UART_LSR_TEMT 0x40 /* Transmitter empty */
```

21.94.1.16 UART_LSR_THRE

```
#define UART_LSR_THRE 0x20 /* Transmit-hold-register empty */
```

21.94.1.17 UART_MCR_OFFSET

```
#define UART_MCR_OFFSET 4 /* Out: Modem Control Register */
```

21.94.1.18 UART_MDR1_OFFSET

```
#define UART_MDR1_OFFSET 8 /* I/O: Mode Register */
```

21.94.1.19 UART_MSR_OFFSET

```
#define UART_MSR_OFFSET 6 /* In: Modem Status Register */
```

21.94.1.20 UART_RBR_OFFSET

```
#define UART_RBR_OFFSET 0 /* In: Recieve Buffer Register */
```

21.94.1.21 UART_SCR_OFFSET

```
#define UART_SCR_OFFSET 7 /* I/O: Scratch Register */
```

21.94.1.22 UART_THR_OFFSET

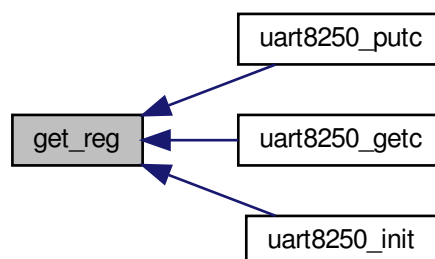
```
#define UART_THR_OFFSET 0 /* Out: Transmitter Holding Register */
```

21.94.2 Function Documentation

21.94.2.1 get_reg()

```
static u32 get_reg (  
    u32 num ) [static]
```

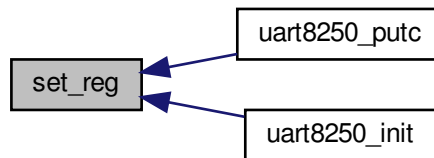
Here is the caller graph for this function:



21.94.2.2 set_reg()

```
static void set_reg (  
    u32 num,  
    u32 val ) [static]
```

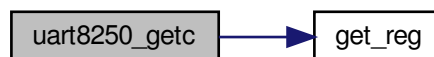
Here is the caller graph for this function:



21.94.2.3 uart8250_getc()

```
int uart8250_getc (  
    void )
```

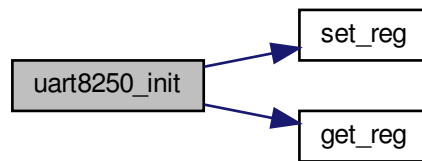
Here is the call graph for this function:



21.94.2.4 uart8250_init()

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

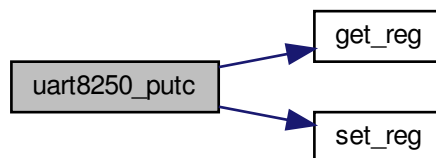
Here is the call graph for this function:



21.94.2.5 uart8250_putc()

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

Here is the call graph for this function:



21.94.3 Variable Documentation

21.94.3.1 uart8250_base

```
volatile void* uart8250_base [static]
```

21.94.3.2 uart8250_baudrate

```
u32 uart8250_baudrate [static]
```

21.94.3.3 uart8250_in_freq

```
u32 uart8250_in_freq [static]
```

21.94.3.4 uart8250_reg_shift

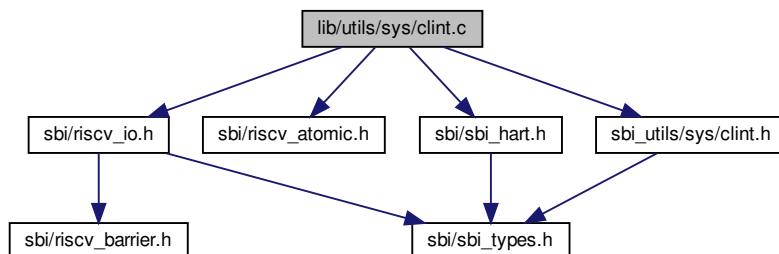
```
u32 uart8250_reg_shift [static]
```

21.94.3.5 uart8250_reg_width

```
u32 uart8250_reg_width [static]
```

21.95 lib/utls/sys/clint.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/riscv_atomic.h>
#include <sbi/sbi_hart.h>
#include <sbi_utils/sys/clint.h>
Include dependency graph for clint.c:
```



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` (unsigned long base, u32 hart_count)
- 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` (unsigned long base, u32 hart_count, bool has_64bit_mmio)

Variables

- static `u32` `clint_ipi_hart_count`
- static volatile void * `clint_ipi_base`
- static volatile `u32` * `clint_ipi`
- static `u32` `clint_time_hart_count`
- static volatile void * `clint_time_base`
- static volatile `u64` * `clint_time_val`
- static volatile `u64` * `clint_time_cmp`
- static `u64`(* `clint_time_rd`)(volatile `u64` *`addr`) = `clint_time_rd32`
- static void(* `clint_time_wr`)(`u64` value, volatile `u64` *`addr`) = `clint_time_wr32`

21.95.1 Function Documentation

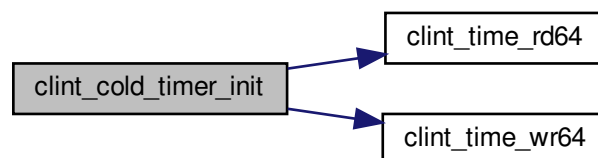
21.95.1.1 `clint_cold_ipi_init()`

```
int clint_cold_ipi_init (  
    unsigned long base,  
    u32 hart_count )
```

21.95.1.2 `clint_cold_timer_init()`

```
int clint_cold_timer_init (  
    unsigned long base,  
    u32 hart_count,  
    bool has_64bit_mmio )
```

Here is the call graph for this function:



21.95.1.3 clint_ipi_clear()

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

Here is the caller graph for this function:



21.95.1.4 clint_ipi_send()

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

21.95.1.5 clint_time_rd32()

```
static u64 clint_time_rd32 (
    volatile u64 * addr ) [static]
```

21.95.1.6 clint_time_rd64()

```
static u64 clint_time_rd64 (
    volatile u64 * addr ) [static]
```

Here is the caller graph for this function:



21.95.1.7 clint_time_wr32()

```
static void clint_time_wr32 (
    u64 value,
    volatile u64 * addr ) [static]
```

21.95.1.8 clint_time_wr64()

```
static void clint_time_wr64 (
    u64 value,
    volatile u64 * addr ) [static]
```

Here is the caller graph for this function:



21.95.1.9 clint_timer_event_start()

```
void clint_timer_event_start (
    u64 next_event )
```

Here is the call graph for this function:



21.95.1.10 clint_timer_event_stop()

```
void clint_timer_event_stop (  
    void )
```

Here is the call graph for this function:



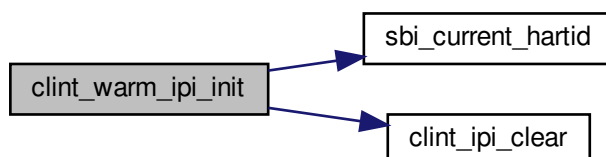
21.95.1.11 clint_timer_value()

```
u64 clint_timer_value (  
    void )
```

21.95.1.12 clint_warm_ipi_init()

```
int clint_warm_ipi_init (  
    void )
```

Here is the call graph for this function:



21.95.1.13 clint_warm_timer_init()

```
int clint_warm_timer_init (  
    void )
```

Here is the call graph for this function:



21.95.2 Variable Documentation

21.95.2.1 clint_ipi

```
volatile u32* clint_ipi [static]
```

21.95.2.2 clint_ipi_base

```
volatile void* clint_ipi_base [static]
```

21.95.2.3 clint_ipi_hart_count

```
u32 clint_ipi_hart_count [static]
```

21.95.2.4 clint_time_base

```
volatile void* clint_time_base [static]
```

21.95.2.5 clint_time_cmp

```
volatile u64* clint_time_cmp [static]
```

21.95.2.6 clint_time_hart_count

```
u32 clint_time_hart_count [static]
```

21.95.2.7 clint_time_rd

```
u64(* clint_time_rd) (volatile u64 *addr) = clint_time_rd32 [static]
```

21.95.2.8 clint_time_val

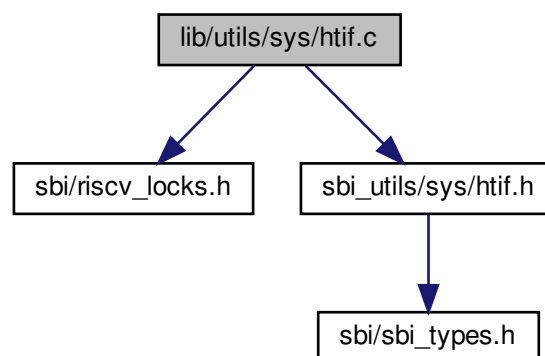
```
volatile u64* clint_time_val [static]
```

21.95.2.9 clint_time_wr

```
void(* clint_time_wr) (u64 value, volatile u64 *addr) = clint_time_wr32 [static]
```

21.96 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_down (u32 type)`

Variables

- `static int htif_console_buf`
- `static spinlock_t htif_lock = SPIN_LOCK_INITIALIZER`

21.96.1 Macro Definition Documentation

21.96.1.1 FROMHOST_CMD

```
#define FROMHOST_CMD(  
    fromhost_value ) ((uint64_t)((fromhost_value) >> HTIF_CMD_SHIFT) & HTIF_CMD_MASK)
```

21.96.1.2 FROMHOST_DATA

```
#define FROMHOST_DATA(  
    fromhost_value ) ((uint64_t)((fromhost_value) >> HTIF_DATA_SHIFT) & HTIF_DATA_←  
MASK)
```

21.96.1.3 FROMHOST_DEV

```
#define FROMHOST_DEV(  
    fromhost_value ) ((uint64_t)((fromhost_value) >> HTIF_DEV_SHIFT) & HTIF_DEV_MA←  
SK)
```

21.96.1.4 HTIF_CMD_BITS

```
#define HTIF_CMD_BITS 8
```

21.96.1.5 HTIF_CMD_MASK

```
#define HTIF_CMD_MASK ((1ULL << HTIF_CMD_BITS) - 1)
```

21.96.1.6 HTIF_CMD_SHIFT

```
#define HTIF_CMD_SHIFT 48
```

21.96.1.7 HTIF_CONSOLE_CMD_GETC

```
#define HTIF_CONSOLE_CMD_GETC 0
```

21.96.1.8 HTIF_CONSOLE_CMD_PUTC

```
#define HTIF_CONSOLE_CMD_PUTC 1
```


21.96.1.9 HTIF_DATA_BITS

```
#define HTIF_DATA_BITS 48
```

21.96.1.10 HTIF_DATA_MASK

```
#define HTIF_DATA_MASK ((1ULL << HTIF_DATA_BITS) - 1)
```

21.96.1.11 HTIF_DATA_SHIFT

```
#define HTIF_DATA_SHIFT 0
```

21.96.1.12 HTIF_DEV_BITS

```
#define HTIF_DEV_BITS 8
```

21.96.1.13 HTIF_DEV_CONSOLE

```
#define HTIF_DEV_CONSOLE 1
```

21.96.1.14 HTIF_DEV_MASK

```
#define HTIF_DEV_MASK ((1ULL << HTIF_DEV_BITS) - 1)
```

21.96.1.15 HTIF_DEV_SHIFT

```
#define HTIF_DEV_SHIFT 56
```

21.96.1.16 HTIF_DEV_SYSTEM

```
#define HTIF_DEV_SYSTEM 0
```

21.96.1.17 PK_SYS_write

```
#define PK_SYS_write 64
```

21.96.1.18 TOHOST_CMD

```
#define TOHOST_CMD(  
    dev,  
    cmd,  
    payload )
```

Value:

```
((  
    if ((dev) || (cmd)) __builtin_trap();  
    (payload); })
```

21.96.2 Function Documentation

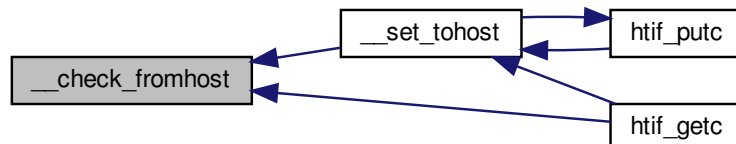
21.96.2.1 __attribute__()

```
volatile uint64_t fromhost __attribute__ (  
    (section(".htif")) )
```

21.96.2.2 __check_fromhost()

```
static void __check_fromhost ( ) [static]
```

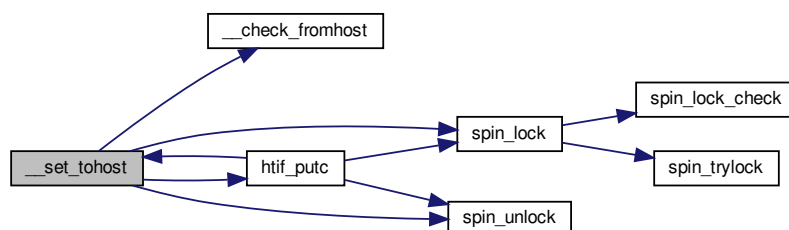
Here is the caller graph for this function:



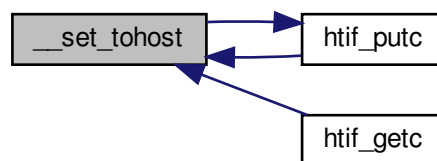
21.96.2.3 __set_tohost()

```
static void __set_tohost (
    uint64_t dev,
    uint64_t cmd,
    uint64_t data ) [static]
```

Here is the call graph for this function:



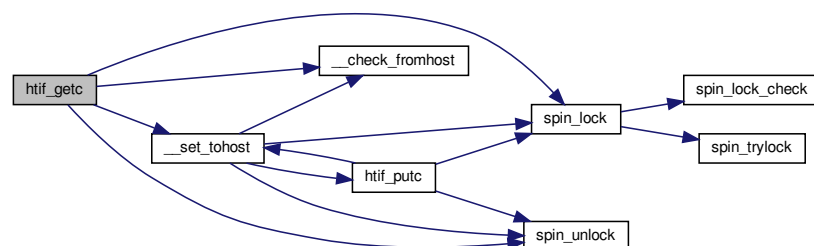
Here is the caller graph for this function:



21.96.2.4 htif_getc()

```
int htif_getc (
    void )
```

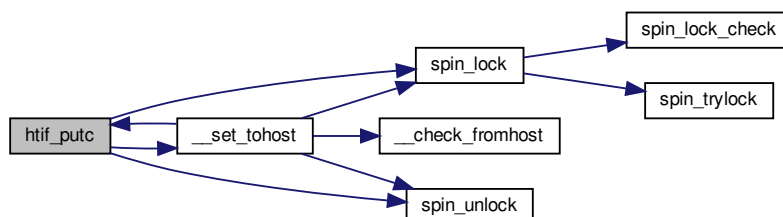
Here is the call graph for this function:



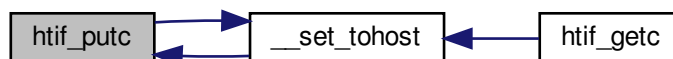
21.96.2.5 htif_putc()

```
void htif_putc (
    char ch )
```

Here is the call graph for this function:



Here is the caller graph for this function:



21.96.2.6 htif_system_down()

```
int htif_system_down (
    u32 type )
```

21.96.3 Variable Documentation

21.96.3.1 htif_console_buf

```
int htif_console_buf [static]
```

21.96.3.2 htif_lock

```
spinlock_t htif_lock = SPIN_LOCK_INITIALIZER [static]
```

21.97 README.md File Reference

Index

- [_AC](#)
 - [sbi_const.h, 187](#)
- [_AT](#)
 - [sbi_const.h, 187](#)
- [_BITUL](#)
 - [sbi_const.h, 187](#)
- [_BITULL](#)
 - [sbi_const.h, 188](#)
- [_UL](#)
 - [sbi_const.h, 188](#)
- [_ULL](#)
 - [sbi_const.h, 188](#)
- [__AC](#)
 - [sbi_const.h, 187](#)
- [__ASM_STR](#)
 - [riscv_asm.h, 78](#)
- [__NOP](#)
 - [riscv_atomic.c, 332](#)
- [__NOT](#)
 - [riscv_atomic.c, 332](#)
- [__RISCV_SPIN_UNLOCKED](#)
 - [riscv_locks.h, 172](#)
- [__STR](#)
 - [sbi_const.h, 187](#)
- [__atomic_op_bit](#)
 - [riscv_atomic.c, 331](#)
- [__atomic_op_bit_ord](#)
 - [riscv_atomic.c, 331](#)
- [__attribute__](#)
 - [htif.c, 658](#)
 - [sbi_hart.c, 379](#)
 - [sbi_hart.h, 212](#)
- [__axchg](#)
 - [riscv_atomic.c, 331](#)
- [__check_fromhost](#)
 - [htif.c, 658](#)
- [__cmpxchg](#)
 - [riscv_atomic.c, 332](#)
- [__ffs](#)
 - [sbi_bitops.h, 179](#)
- [__fls](#)
 - [sbi_bitops.h, 179](#)
- [__io_ar](#)
 - [riscv_io.h, 167](#)
- [__io_aw](#)
 - [riscv_io.h, 167](#)
- [__io_br](#)
 - [riscv_io.h, 167](#)
- [__io_bw](#)
 - [riscv_io.h, 167](#)
- [__io_rar](#)
 - [riscv_io.h, 167](#)
- [__io_raw](#)
 - [riscv_io.h, 167](#)
- [__io_rbr](#)
 - [riscv_io.h, 167](#)
- [__io_rbw](#)
 - [riscv_io.h, 167](#)
- [__noreturn](#)
 - [sbi_types.h, 303](#)
- [__packed](#)
 - [fw_dynamic.h, 76](#)
 - [sbi_platform.h, 270](#)
 - [sbi_scratch.h, 276](#)
 - [sbi_trap.h, 302](#)
 - [sbi_types.h, 303](#)
- [__printf](#)
 - [sbi_console.h, 182, 183](#)
- [__raw_readb](#)
 - [riscv_io.h, 170](#)
- [__raw_readl](#)
 - [riscv_io.h, 170](#)
- [__raw_readq](#)
 - [riscv_io.h, 170](#)
- [__raw_readw](#)
 - [riscv_io.h, 171](#)
- [__raw_writeb](#)
 - [riscv_io.h, 171](#)
- [__raw_writel](#)
 - [riscv_io.h, 171](#)
- [__raw_writelq](#)
 - [riscv_io.h, 171](#)
- [__raw_writew](#)
 - [riscv_io.h, 171](#)
- [__sbi_fifo_enqueue](#)
 - [sbi_fifo.c, 371](#)
- [__sbi_fifo_is_empty](#)
 - [sbi_fifo.c, 371](#)
- [__sbi_fifo_is_full](#)
 - [sbi_fifo.c, 371](#)
- [__sbi_fifo_reset](#)
 - [sbi_fifo.c, 372](#)
- [__sbi_hfence_gvma_all](#)
 - [sbi_hfence.h, 220](#)
- [__sbi_hfence_gvma_gpa](#)
 - [sbi_hfence.h, 220](#)
- [__sbi_hfence_gvma_vmid](#)
 - [sbi_hfence.h, 221](#)

- __sbi_hfence_gvma_vmid_gpa
 - sbi_hfence.h, [221](#)
- __sbi_hfence_vvma_all
 - sbi_hfence.h, [221](#)
- __sbi_hfence_vvma_asid
 - sbi_hfence.h, [222](#)
- __sbi_hfence_vvma_asid_va
 - sbi_hfence.h, [222](#)
- __sbi_hfence_vvma_va
 - sbi_hfence.h, [222](#)
- __sbi_list_add
 - sbi_list.h, [239](#)
- __sbi_list_del
 - sbi_list.h, [240](#)
- __sbi_list_del_entry
 - sbi_list.h, [240](#)
- __sbi_tlb_range_check
 - sbi_tlb.c, [429](#)
- __set_tohost
 - htif.c, [658](#)
- __smp_load_acquire
 - riscv_barrier.h, [91](#)
- __smp_store_release
 - riscv_barrier.h, [91](#)
- __xchg
 - riscv_atomic.c, [332](#)
- a0
 - sbi_trap_regs, [67](#)
- a1
 - sbi_trap_regs, [67](#)
- a2
 - sbi_trap_regs, [67](#)
- a3
 - sbi_trap_regs, [67](#)
- a4
 - sbi_trap_regs, [67](#)
- a5
 - sbi_trap_regs, [67](#)
- a6
 - sbi_trap_regs, [67](#)
- a7
 - sbi_trap_regs, [68](#)
- ATOMIC_INITIALIZER
 - riscv_atomic.h, [86](#)
- ATOMIC_INIT
 - riscv_atomic.h, [86](#)
- address
 - fdt_reserve_entry, [49](#)
- arch_atomic_cmpxchg
 - riscv_atomic.c, [334](#)
 - riscv_atomic.h, [87](#)
- arch_atomic_xchg
 - riscv_atomic.c, [334](#)
 - riscv_atomic.h, [87](#)
- arg1
 - sbi_hart.h, [219](#)
- asid
 - sbi_tlb_info, [64](#)
- atomic_add_return
 - riscv_atomic.c, [334](#)
 - riscv_atomic.h, [87](#)
- atomic_clear_bit
 - riscv_atomic.c, [334](#)
 - riscv_atomic.h, [87](#)
- atomic_raw_clear_bit
 - riscv_atomic.c, [335](#)
 - riscv_atomic.h, [88](#)
- atomic_raw_set_bit
 - riscv_atomic.c, [335](#)
 - riscv_atomic.h, [88](#)
- atomic_raw_xchg_uint
 - riscv_atomic.c, [336](#)
 - riscv_atomic.h, [89](#)
- atomic_raw_xchg_ulong
 - riscv_atomic.c, [336](#)
 - riscv_atomic.h, [89](#)
- atomic_read
 - riscv_atomic.c, [336](#)
 - riscv_atomic.h, [89](#)
- atomic_set_bit
 - riscv_atomic.c, [336](#)
 - riscv_atomic.h, [89](#)
- atomic_sub_return
 - riscv_atomic.c, [337](#)
 - riscv_atomic.h, [90](#)
- atomic_t, [45](#)
 - counter, [45](#)
- atomic_write
 - riscv_atomic.c, [337](#)
 - riscv_atomic.h, [90](#)
- avail
 - sbi_fifo, [53](#)
- avail_hart_mask
 - sbi_hart.c, [389](#)
- avail_hart_mask_lock
 - sbi_hart.c, [389](#)
- axchg
 - riscv_atomic.c, [333](#)
- BANNER
 - sbi_init.c, [394](#)
- BIT_MASK
 - sbi_bits.h, [180](#)
- BIT_WORD
 - sbi_bits.h, [180](#)
- bool
 - sbi_types.h, [306](#)
- boot_cpuid_phys
 - fdt_header, [46](#)
- boot_hart
 - fw_dynamic_info, [50](#)
- CAUSE_BREAKPOINT
 - riscv_encoding.h, [102](#)
- CAUSE_FETCH_ACCESS
 - riscv_encoding.h, [102](#)
- CAUSE_FETCH_GUEST_PAGE_FAULT

- riscv_encoding.h, 102
- CAUSE_FETCH_PAGE_FAULT
 - riscv_encoding.h, 103
- CAUSE_HYPERVISOR_ECALL
 - riscv_encoding.h, 103
- CAUSE_ILLEGAL_INSTRUCTION
 - riscv_encoding.h, 103
- CAUSE_LOAD_ACCESS
 - riscv_encoding.h, 103
- CAUSE_LOAD_GUEST_PAGE_FAULT
 - riscv_encoding.h, 103
- CAUSE_LOAD_PAGE_FAULT
 - riscv_encoding.h, 103
- CAUSE_MACHINE_ECALL
 - riscv_encoding.h, 103
- CAUSE_MISALIGNED_FETCH
 - riscv_encoding.h, 103
- CAUSE_MISALIGNED_LOAD
 - riscv_encoding.h, 104
- CAUSE_MISALIGNED_STORE
 - riscv_encoding.h, 104
- CAUSE_STORE_ACCESS
 - riscv_encoding.h, 104
- CAUSE_STORE_GUEST_PAGE_FAULT
 - riscv_encoding.h, 104
- CAUSE_STORE_PAGE_FAULT
 - riscv_encoding.h, 104
- CAUSE_SUPERVISOR_ECALL
 - riscv_encoding.h, 104
- CAUSE_USER_ECALL
 - riscv_encoding.h, 104
- CLAMP
 - sbi_types.h, 303
- COLDBOOT_WAIT_BITMAP_SIZE
 - sbi_hart.c, 379
- CPU_TO_FDT16
 - libfdt_env.h, 622
- CPU_TO_FDT32
 - libfdt_env.h, 622
- CPU_TO_FDT64
 - libfdt_env.h, 622
- CSR_CYCLEH
 - riscv_encoding.h, 105
- CSR_CYCLE
 - riscv_encoding.h, 104
- CSR_DCSR
 - riscv_encoding.h, 105
- CSR_DPC
 - riscv_encoding.h, 105
- CSR_DSCRATCH
 - riscv_encoding.h, 105
- CSR_FCSR
 - riscv_encoding.h, 105
- CSR_FFLAGS
 - riscv_encoding.h, 105
- CSR_FRM
 - riscv_encoding.h, 105
- CSR_HCOUNTERNEN
 - riscv_encoding.h, 105
- CSR_HEDELEG
 - riscv_encoding.h, 106
- CSR_HGATP
 - riscv_encoding.h, 106
- CSR_HGEIE
 - riscv_encoding.h, 106
- CSR_HGEIP
 - riscv_encoding.h, 106
- CSR_HIDELEG
 - riscv_encoding.h, 106
- CSR_HIE
 - riscv_encoding.h, 106
- CSR_HIP
 - riscv_encoding.h, 106
- CSR_HPMCounter10
 - riscv_encoding.h, 106
- CSR_HPMCounter10H
 - riscv_encoding.h, 107
- CSR_HPMCounter11
 - riscv_encoding.h, 107
- CSR_HPMCounter11H
 - riscv_encoding.h, 107
- CSR_HPMCounter12
 - riscv_encoding.h, 107
- CSR_HPMCounter12H
 - riscv_encoding.h, 107
- CSR_HPMCounter13
 - riscv_encoding.h, 107
- CSR_HPMCounter13H
 - riscv_encoding.h, 107
- CSR_HPMCounter14
 - riscv_encoding.h, 107
- CSR_HPMCounter14H
 - riscv_encoding.h, 108
- CSR_HPMCounter15
 - riscv_encoding.h, 108
- CSR_HPMCounter15H
 - riscv_encoding.h, 108
- CSR_HPMCounter16
 - riscv_encoding.h, 108
- CSR_HPMCounter16H
 - riscv_encoding.h, 108
- CSR_HPMCounter17
 - riscv_encoding.h, 108
- CSR_HPMCounter17H
 - riscv_encoding.h, 108
- CSR_HPMCounter18
 - riscv_encoding.h, 108
- CSR_HPMCounter18H
 - riscv_encoding.h, 109
- CSR_HPMCounter19
 - riscv_encoding.h, 109
- CSR_HPMCounter19H
 - riscv_encoding.h, 109
- CSR_HPMCounter20
 - riscv_encoding.h, 109
- CSR_HPMCounter20H

- riscv_encoding.h, [109](#)
- CSR_HPMCOUNTER21
 - riscv_encoding.h, [109](#)
- CSR_HPMCOUNTER21H
 - riscv_encoding.h, [109](#)
- CSR_HPMCOUNTER22
 - riscv_encoding.h, [109](#)
- CSR_HPMCOUNTER22H
 - riscv_encoding.h, [110](#)
- CSR_HPMCOUNTER23
 - riscv_encoding.h, [110](#)
- CSR_HPMCOUNTER23H
 - riscv_encoding.h, [110](#)
- CSR_HPMCOUNTER24
 - riscv_encoding.h, [110](#)
- CSR_HPMCOUNTER24H
 - riscv_encoding.h, [110](#)
- CSR_HPMCOUNTER25
 - riscv_encoding.h, [110](#)
- CSR_HPMCOUNTER25H
 - riscv_encoding.h, [110](#)
- CSR_HPMCOUNTER26
 - riscv_encoding.h, [110](#)
- CSR_HPMCOUNTER26H
 - riscv_encoding.h, [111](#)
- CSR_HPMCOUNTER27
 - riscv_encoding.h, [111](#)
- CSR_HPMCOUNTER27H
 - riscv_encoding.h, [111](#)
- CSR_HPMCOUNTER28
 - riscv_encoding.h, [111](#)
- CSR_HPMCOUNTER28H
 - riscv_encoding.h, [111](#)
- CSR_HPMCOUNTER29
 - riscv_encoding.h, [111](#)
- CSR_HPMCOUNTER29H
 - riscv_encoding.h, [111](#)
- CSR_HPMCOUNTER3
 - riscv_encoding.h, [111](#)
- CSR_HPMCOUNTER30
 - riscv_encoding.h, [112](#)
- CSR_HPMCOUNTER30H
 - riscv_encoding.h, [112](#)
- CSR_HPMCOUNTER31
 - riscv_encoding.h, [112](#)
- CSR_HPMCOUNTER31H
 - riscv_encoding.h, [112](#)
- CSR_HPMCOUNTER3H
 - riscv_encoding.h, [112](#)
- CSR_HPMCOUNTER4
 - riscv_encoding.h, [112](#)
- CSR_HPMCOUNTER4H
 - riscv_encoding.h, [112](#)
- CSR_HPMCOUNTER5
 - riscv_encoding.h, [112](#)
- CSR_HPMCOUNTER5H
 - riscv_encoding.h, [113](#)
- CSR_HPMCOUNTER6
 - riscv_encoding.h, [113](#)
- CSR_HPMCOUNTER6H
 - riscv_encoding.h, [113](#)
- CSR_HPMCOUNTER7
 - riscv_encoding.h, [113](#)
- CSR_HPMCOUNTER7H
 - riscv_encoding.h, [113](#)
- CSR_HPMCOUNTER8
 - riscv_encoding.h, [113](#)
- CSR_HPMCOUNTER8H
 - riscv_encoding.h, [113](#)
- CSR_HPMCOUNTER9
 - riscv_encoding.h, [113](#)
- CSR_HPMCOUNTER9H
 - riscv_encoding.h, [114](#)
- CSR_HSTATUS
 - riscv_encoding.h, [114](#)
- CSR_HTIMEDELTAH
 - riscv_encoding.h, [114](#)
- CSR_HTIMEDELTA
 - riscv_encoding.h, [114](#)
- CSR_HTINST
 - riscv_encoding.h, [114](#)
- CSR_HTVAL
 - riscv_encoding.h, [114](#)
- CSR_INSTRETH
 - riscv_encoding.h, [114](#)
- CSR_INSTRET
 - riscv_encoding.h, [114](#)
- CSR_MARCHID
 - riscv_encoding.h, [115](#)
- CSR_MCAUSE
 - riscv_encoding.h, [115](#)
- CSR_MCOUNTEREN
 - riscv_encoding.h, [115](#)
- CSR_MCYCLEH
 - riscv_encoding.h, [115](#)
- CSR_MCYCLE
 - riscv_encoding.h, [115](#)
- CSR_MEDELEG
 - riscv_encoding.h, [115](#)
- CSR_MEPC
 - riscv_encoding.h, [115](#)
- CSR_MHARTID
 - riscv_encoding.h, [115](#)
- 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, [116](#)
- CSR_MHPMCOUNTER13H
 - riscv_encoding.h, [116](#)
- 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, [118](#)
- CSR_MHPMCOUNTER18H
 - riscv_encoding.h, [118](#)
- 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, [119](#)
- CSR_MHPMCOUNTER22H
 - riscv_encoding.h, [119](#)
- CSR_MHPMCOUNTER23
 - riscv_encoding.h, [119](#)
- CSR_MHPMCOUNTER23H
 - riscv_encoding.h, [119](#)
- CSR_MHPMCOUNTER24
 - riscv_encoding.h, [119](#)
- CSR_MHPMCOUNTER24H
 - riscv_encoding.h, [119](#)
- CSR_MHPMCOUNTER25
 - riscv_encoding.h, [119](#)
- CSR_MHPMCOUNTER25H
 - riscv_encoding.h, [119](#)
- CSR_MHPMCOUNTER26
 - riscv_encoding.h, [120](#)
- CSR_MHPMCOUNTER26H
 - riscv_encoding.h, [120](#)
- CSR_MHPMCOUNTER27
 - riscv_encoding.h, [120](#)
- CSR_MHPMCOUNTER27H
 - riscv_encoding.h, [120](#)
- CSR_MHPMCOUNTER28
 - riscv_encoding.h, [120](#)
- CSR_MHPMCOUNTER28H
 - riscv_encoding.h, [120](#)
- CSR_MHPMCOUNTER29
 - riscv_encoding.h, [120](#)
- CSR_MHPMCOUNTER29H
 - riscv_encoding.h, [120](#)
- CSR_MHPMCOUNTER3
 - riscv_encoding.h, [121](#)
- CSR_MHPMCOUNTER30
 - riscv_encoding.h, [121](#)
- CSR_MHPMCOUNTER30H
 - riscv_encoding.h, [121](#)
- CSR_MHPMCOUNTER31
 - riscv_encoding.h, [121](#)
- CSR_MHPMCOUNTER31H
 - riscv_encoding.h, [121](#)
- CSR_MHPMCOUNTER3H
 - riscv_encoding.h, [121](#)
- CSR_MHPMCOUNTER4
 - riscv_encoding.h, [121](#)
- CSR_MHPMCOUNTER4H
 - riscv_encoding.h, [121](#)
- CSR_MHPMCOUNTER5
 - riscv_encoding.h, [122](#)
- CSR_MHPMCOUNTER5H
 - riscv_encoding.h, [122](#)
- CSR_MHPMCOUNTER6
 - riscv_encoding.h, [122](#)
- CSR_MHPMCOUNTER6H
 - riscv_encoding.h, [122](#)
- CSR_MHPMCOUNTER7
 - riscv_encoding.h, [122](#)
- CSR_MHPMCOUNTER7H
 - riscv_encoding.h, [122](#)
- CSR_MHPMCOUNTER8
 - riscv_encoding.h, [122](#)
- CSR_MHPMCOUNTER8H
 - riscv_encoding.h, [122](#)
- CSR_MHPMCOUNTER9
 - riscv_encoding.h, [123](#)
- CSR_MHPMCOUNTER9H
 - riscv_encoding.h, [123](#)
- CSR_MHPMEVENT10
 - riscv_encoding.h, [123](#)
- CSR_MHPMEVENT11
 - riscv_encoding.h, [123](#)
- CSR_MHPMEVENT12
 - riscv_encoding.h, [123](#)
- CSR_MHPMEVENT13
 - riscv_encoding.h, [123](#)
- CSR_MHPMEVENT14
 - riscv_encoding.h, [123](#)
- CSR_MHPMEVENT15
 - riscv_encoding.h, [123](#)
- CSR_MHPMEVENT16

- riscv_encoding.h, [124](#)
- CSR_MHPMEVENT17
 - riscv_encoding.h, [124](#)
- CSR_MHPMEVENT18
 - riscv_encoding.h, [124](#)
- CSR_MHPMEVENT19
 - riscv_encoding.h, [124](#)
- CSR_MHPMEVENT20
 - riscv_encoding.h, [124](#)
- CSR_MHPMEVENT21
 - riscv_encoding.h, [124](#)
- CSR_MHPMEVENT22
 - riscv_encoding.h, [124](#)
- CSR_MHPMEVENT23
 - riscv_encoding.h, [124](#)
- CSR_MHPMEVENT24
 - riscv_encoding.h, [125](#)
- CSR_MHPMEVENT25
 - riscv_encoding.h, [125](#)
- CSR_MHPMEVENT26
 - riscv_encoding.h, [125](#)
- CSR_MHPMEVENT27
 - riscv_encoding.h, [125](#)
- CSR_MHPMEVENT28
 - riscv_encoding.h, [125](#)
- CSR_MHPMEVENT29
 - riscv_encoding.h, [125](#)
- CSR_MHPMEVENT3
 - riscv_encoding.h, [125](#)
- CSR_MHPMEVENT30
 - riscv_encoding.h, [125](#)
- CSR_MHPMEVENT31
 - riscv_encoding.h, [126](#)
- CSR_MHPMEVENT4
 - riscv_encoding.h, [126](#)
- CSR_MHPMEVENT5
 - riscv_encoding.h, [126](#)
- CSR_MHPMEVENT6
 - riscv_encoding.h, [126](#)
- CSR_MHPMEVENT7
 - riscv_encoding.h, [126](#)
- CSR_MHPMEVENT8
 - riscv_encoding.h, [126](#)
- CSR_MHPMEVENT9
 - riscv_encoding.h, [126](#)
- CSR_MIDELEG
 - riscv_encoding.h, [126](#)
- CSR_MIMPID
 - riscv_encoding.h, [127](#)
- CSR_MINSTRETH
 - riscv_encoding.h, [127](#)
- CSR_MINSTRET
 - riscv_encoding.h, [127](#)
- CSR_MISA
 - riscv_encoding.h, [127](#)
- CSR_MIE
 - riscv_encoding.h, [127](#)
- CSR_MIP
 - riscv_encoding.h, [127](#)
- riscv_encoding.h, [127](#)
- CSR_MSCRATCH
 - riscv_encoding.h, [127](#)
- CSR_MSTATUSH
 - riscv_encoding.h, [128](#)
- CSR_MSTATUS
 - riscv_encoding.h, [127](#)
- CSR_MTINST
 - riscv_encoding.h, [128](#)
- CSR_MTVAL2
 - riscv_encoding.h, [128](#)
- CSR_MTVAL
 - riscv_encoding.h, [128](#)
- CSR_MTVEC
 - riscv_encoding.h, [128](#)
- CSR_MVENDORID
 - riscv_encoding.h, [128](#)
- CSR_PMPADDR0
 - riscv_encoding.h, [128](#)
- CSR_PMPADDR1
 - riscv_encoding.h, [128](#)
- CSR_PMPADDR10
 - riscv_encoding.h, [129](#)
- CSR_PMPADDR11
 - riscv_encoding.h, [129](#)
- CSR_PMPADDR12
 - riscv_encoding.h, [129](#)
- CSR_PMPADDR13
 - riscv_encoding.h, [129](#)
- CSR_PMPADDR14
 - riscv_encoding.h, [129](#)
- CSR_PMPADDR15
 - riscv_encoding.h, [129](#)
- CSR_PMPADDR2
 - riscv_encoding.h, [129](#)
- CSR_PMPADDR3
 - riscv_encoding.h, [129](#)
- CSR_PMPADDR4
 - riscv_encoding.h, [130](#)
- CSR_PMPADDR5
 - riscv_encoding.h, [130](#)
- CSR_PMPADDR6
 - riscv_encoding.h, [130](#)
- CSR_PMPADDR7
 - riscv_encoding.h, [130](#)
- CSR_PMPADDR8
 - riscv_encoding.h, [130](#)
- CSR_PMPADDR9
 - riscv_encoding.h, [130](#)
- CSR_PMPCFG0
 - riscv_encoding.h, [130](#)
- CSR_PMPCFG1
 - riscv_encoding.h, [130](#)
- CSR_PMPCFG2
 - riscv_encoding.h, [131](#)
- CSR_PMPCFG3
 - riscv_encoding.h, [131](#)
- CSR_SATP

- riscv_encoding.h, [131](#)
- CSR_SCAUSE
 - riscv_encoding.h, [131](#)
- CSR_SCOUNTEREN
 - riscv_encoding.h, [131](#)
- CSR_SEPC
 - riscv_encoding.h, [131](#)
- CSR_SIE
 - riscv_encoding.h, [131](#)
- CSR_SIP
 - riscv_encoding.h, [131](#)
- CSR_SSCRATCH
 - riscv_encoding.h, [132](#)
- CSR_SSTATUS
 - riscv_encoding.h, [132](#)
- CSR_STVAL
 - riscv_encoding.h, [132](#)
- CSR_STVEC
 - riscv_encoding.h, [132](#)
- CSR_TDATA1
 - riscv_encoding.h, [132](#)
- CSR_TDATA2
 - riscv_encoding.h, [132](#)
- CSR_TDATA3
 - riscv_encoding.h, [132](#)
- CSR_TIMEH
 - riscv_encoding.h, [133](#)
- CSR_TIME
 - riscv_encoding.h, [132](#)
- CSR_TSELECT
 - riscv_encoding.h, [133](#)
- CSR_UCAUSE
 - riscv_encoding.h, [133](#)
- CSR_UEPC
 - riscv_encoding.h, [133](#)
- CSR_UIE
 - riscv_encoding.h, [133](#)
- CSR_UIP
 - riscv_encoding.h, [133](#)
- CSR_USCRATCH
 - riscv_encoding.h, [133](#)
- CSR_USTATUS
 - riscv_encoding.h, [133](#)
- CSR_UTVAL
 - riscv_encoding.h, [134](#)
- CSR_UTVEC
 - riscv_encoding.h, [134](#)
- CSR_VSATP
 - riscv_encoding.h, [134](#)
- CSR_VSCAUSE
 - riscv_encoding.h, [134](#)
- CSR_VSEPC
 - riscv_encoding.h, [134](#)
- CSR_VSIE
 - riscv_encoding.h, [134](#)
- CSR_VSIP
 - riscv_encoding.h, [134](#)
- CSR_VSSCRATCH
 - riscv_encoding.h, [134](#)
- CSR_VSSTATUS
 - riscv_encoding.h, [135](#)
- CSR_VSTVAL
 - riscv_encoding.h, [135](#)
- CSR_VSTVEC
 - riscv_encoding.h, [135](#)
- cause
 - sbi_trap_info, [65](#)
- check_block_
 - fdt.c, [453](#)
- check_off_
 - fdt.c, [453](#)
- clint.c
 - clint_cold_ipi_init, [649](#)
 - clint_cold_timer_init, [649](#)
 - clint_ipi, [653](#)
 - clint_ipi_base, [653](#)
 - clint_ipi_clear, [649](#)
 - clint_ipi_hart_count, [653](#)
 - clint_ipi_send, [650](#)
 - clint_time_base, [653](#)
 - clint_time_cmp, [653](#)
 - clint_time_hart_count, [654](#)
 - clint_time_rd, [654](#)
 - clint_time_rd32, [650](#)
 - clint_time_rd64, [650](#)
 - clint_time_val, [654](#)
 - clint_time_wr, [654](#)
 - clint_time_wr32, [650](#)
 - clint_time_wr64, [651](#)
 - clint_timer_event_start, [651](#)
 - clint_timer_event_stop, [651](#)
 - clint_timer_value, [652](#)
 - clint_warm_ipi_init, [652](#)
 - clint_warm_timer_init, [652](#)
- clint.h
 - clint_cold_ipi_init, [320](#)
 - clint_cold_timer_init, [321](#)
 - clint_ipi_clear, [321](#)
 - clint_ipi_send, [321](#)
 - clint_ipi_sync, [322](#)
 - clint_timer_event_start, [322](#)
 - clint_timer_event_stop, [322](#)
 - clint_timer_value, [322](#)
 - clint_warm_ipi_init, [323](#)
 - clint_warm_timer_init, [323](#)
- clint_cold_ipi_init
 - clint.c, [649](#)
 - clint.h, [320](#)
- clint_cold_timer_init
 - clint.c, [649](#)
 - clint.h, [321](#)
- clint_ipi
 - clint.c, [653](#)
- clint_ipi_base
 - clint.c, [653](#)
- clint_ipi_clear

- clint.c, [649](#)
- clint.h, [321](#)
- clint_ipi_hart_count
 - clint.c, [653](#)
- clint_ipi_send
 - clint.c, [650](#)
 - clint.h, [321](#)
- clint_ipi_sync
 - clint.h, [322](#)
- clint_time_base
 - clint.c, [653](#)
- clint_time_cmp
 - clint.c, [653](#)
- clint_time_hart_count
 - clint.c, [654](#)
- clint_time_rd
 - clint.c, [654](#)
- clint_time_rd32
 - clint.c, [650](#)
- clint_time_rd64
 - clint.c, [650](#)
- clint_time_val
 - clint.c, [654](#)
- clint_time_wr
 - clint.c, [654](#)
- clint_time_wr32
 - clint.c, [650](#)
- clint_time_wr64
 - clint.c, [651](#)
- clint_timer_event_start
 - clint.c, [651](#)
 - clint.h, [322](#)
- clint_timer_event_stop
 - clint.c, [651](#)
 - clint.h, [322](#)
- clint_timer_value
 - clint.c, [652](#)
 - clint.h, [322](#)
- clint_warm_ipi_init
 - clint.c, [652](#)
 - clint.h, [323](#)
- clint_warm_timer_init
 - clint.c, [652](#)
 - clint.h, [323](#)
- cmpxchg
 - riscv_atomic.c, [333](#)
- coldboot_done
 - sbi_hart.c, [389](#)
- coldboot_lock
 - sbi_hart.c, [389](#)
- coldboot_lottery
 - sbi_init.c, [400](#)
- coldboot_wait_bitmap
 - sbi_hart.c, [389](#)
- console_getc
 - sbi_platform_operations, [58](#)
- console_init
 - sbi_platform_operations, [58](#)
- console_out_lock
 - sbi_console.c, [353](#)
- console_plat
 - sbi_console.c, [353](#)
- console_putc
 - sbi_platform_operations, [58](#)
- container_of
 - sbi_types.h, [304](#)
- counter
 - atomic_t, [45](#)
- cpu_relax
 - riscv_barrier.h, [91](#)
- cpu_to_fdt16
 - libfdt_env.h, [625](#)
- cpu_to_fdt32
 - libfdt_env.h, [625](#)
- cpu_to_fdt64
 - libfdt_env.h, [626](#)
- csr_clear
 - riscv_asm.h, [78](#)
- csr_read
 - riscv_asm.h, [78](#)
- csr_read_clear
 - riscv_asm.h, [78](#)
- csr_read_num
 - riscv_asm.c, [326](#)
 - riscv_asm.h, [82](#)
- csr_read_set
 - riscv_asm.h, [79](#)
- csr_set
 - riscv_asm.h, [79](#)
- csr_swap
 - riscv_asm.h, [79](#)
- csr_write
 - riscv_asm.h, [80](#)
- csr_write_num
 - riscv_asm.c, [327](#)
 - riscv_asm.h, [82](#)
- ctz
 - riscv_asm.c, [327](#)
- DECLARE_UNPRIVILEGED_LOAD_FUNCTION
 - sbi_unpriv.h, [309](#)
- DECLARE_UNPRIVILEGED_STORE_FUNCTION
 - sbi_unpriv.h, [310](#)
- DEFINE_UNPRIVILEGED_LOAD_FUNCTION
 - sbi_unpriv.c, [444](#)
- DEFINE_UNPRIVILEGED_STORE_FUNCTION
 - sbi_unpriv.c, [445](#)
- data
 - fdt_property, [48](#)
- delegate_traps
 - sbi_hart.c, [379](#)
- disabled_hart_mask
 - sbi_platform, [56](#)
- docs/contributing.md, [73](#)
- docs/firmware/fw.md, [73](#)
- docs/firmware/fw_dynamic.md, [73](#)
- docs/firmware/fw_jump.md, [73](#)

- docs/firmware/fw_payload.md, [73](#)
- docs/firmware/payload_linux.md, [73](#)
- docs/firmware/payload_uboot.md, [73](#)
- docs/library_usage.md, [73](#)
- docs/platform/andes-ae350.md, [73](#)
- docs/platform/ariane-fpga.md, [73](#)
- docs/platform/platform.md, [73](#)
- docs/platform/qemu_virt.md, [73](#)
- docs/platform/sifive_fu540.md, [74](#)
- docs/platform/spike.md, [74](#)
- docs/platform/thead-c910.md, [74](#)
- docs/platform_guide.md, [74](#)
- EXTRACT_BYTE
 - libfdt_env.h, [622](#)
- EXTRACT_FIELD
 - sbi_bits.h, [181](#)
- early_exit
 - sbi_platform_operations, [58](#)
- early_init
 - sbi_platform_operations, [58](#)
- ecall_base
 - sbi_ecall.h, [194](#)
 - sbi_ecall_base.c, [360](#)
- ecall_ipi
 - sbi_ecall.h, [194](#)
 - sbi_ecall_replace.c, [364](#)
- ecall_legacy
 - sbi_ecall.h, [194](#)
 - sbi_ecall_legacy.c, [362](#)
- ecall_rfence
 - sbi_ecall.h, [194](#)
 - sbi_ecall_replace.c, [365](#)
- ecall_time
 - sbi_ecall.h, [194](#)
 - sbi_ecall_replace.c, [365](#)
- ecall_vendor
 - sbi_ecall.h, [194](#)
 - sbi_ecall_vendor.c, [367](#)
- entry_size
 - sbi_fifo, [53](#)
- epc
 - sbi_trap_info, [65](#)
- extid_end
 - sbi_ecall_extension, [52](#)
- extid_start
 - sbi_ecall_extension, [52](#)
- extra_lock
 - sbi_scratch.c, [414](#)
- extra_offset
 - sbi_scratch.c, [414](#)
- FALSE
 - sbi_types.h, [304](#)
- FDT_ALIGN
 - libfdt_internal.h, [628](#)
- FDT_BEGIN_NODE
 - fdt.h, [463](#)
- FDT_BITWISE
 - libfdt_env.h, [623](#)
- FDT_END_NODE
 - fdt.h, [463](#)
- FDT_END
 - fdt.h, [463](#)
- FDT_ERR_BADLAYOUT
 - libfdt.h, [552](#)
- FDT_ERR_BADMAGIC
 - libfdt.h, [553](#)
- FDT_ERR_BADNCELLS
 - libfdt.h, [553](#)
- FDT_ERR_BADOFFSET
 - libfdt.h, [553](#)
- FDT_ERR_BADOVERLAY
 - libfdt.h, [553](#)
- FDT_ERR_BADPATH
 - libfdt.h, [553](#)
- FDT_ERR_BADPHANDLE
 - libfdt.h, [553](#)
- FDT_ERR_BADSTATE
 - libfdt.h, [553](#)
- FDT_ERR_BADSTRUCTURE
 - libfdt.h, [553](#)
- FDT_ERR_BADVALUE
 - libfdt.h, [554](#)
- FDT_ERR_BADVERSION
 - libfdt.h, [554](#)
- FDT_ERR_EXISTS
 - libfdt.h, [554](#)
- FDT_ERR_INTERNAL
 - libfdt.h, [554](#)
- FDT_ERR_MAX
 - libfdt.h, [554](#)
- FDT_ERR_NOPHANDLES
 - libfdt.h, [554](#)
- FDT_ERR_NOSPACE
 - libfdt.h, [554](#)
- FDT_ERR_NOTFOUND
 - libfdt.h, [554](#)
- FDT_ERR_TRUNCATED
 - libfdt.h, [555](#)
- FDT_ERRTABENT
 - fdt_strerror.c, [531](#)
- FDT_ERRTABSIZE
 - fdt_strerror.c, [531](#)
- FDT_FIRST_SUPPORTED_VERSION
 - libfdt.h, [555](#)
- FDT_FORCE
 - libfdt_env.h, [623](#)
- FDT_LAST_SUPPORTED_VERSION
 - libfdt.h, [556](#)
- FDT_MAGIC
 - fdt.h, [464](#)
- FDT_MAX_NCELLS
 - libfdt.h, [556](#)
- FDT_NOP
 - fdt.h, [464](#)
- FDT_PROP

- fdt.h, [464](#)
- FDT_RO_PROBE
 - libfdt_internal.h, [629](#)
- FDT_RW_PROBE
 - fdt_rw.c, [517](#)
- FDT_SW_MAGIC
 - libfdt_internal.h, [629](#)
- FDT_SW_PROBE_MEMRSV
 - fdt_sw.c, [533](#)
- FDT_SW_PROBE_STRUCT
 - fdt_sw.c, [533](#)
- FDT_SW_PROBE
 - fdt_sw.c, [533](#)
- FDT_TAGALIGN
 - libfdt_internal.h, [629](#)
- FDT_TAGSIZE
 - fdt.h, [464](#)
- FDT_V16_SIZE
 - fdt.h, [464](#)
- FDT_V17_SIZE
 - fdt.h, [464](#)
- FDT_V1_SIZE
 - fdt.h, [464](#)
- FDT_V2_SIZE
 - fdt.h, [465](#)
- FDT_V3_SIZE
 - fdt.h, [465](#)
- FROMHOST_CMD
 - htif.c, [655](#)
- FROMHOST_DATA
 - htif.c, [655](#)
- FROMHOST_DEV
 - htif.c, [656](#)
- FW_DYNAMIC_INFO_BOOT_HART_OFFSET
 - fw_dynamic.h, [75](#)
- FW_DYNAMIC_INFO_MAGIC_OFFSET
 - fw_dynamic.h, [75](#)
- FW_DYNAMIC_INFO_MAGIC_VALUE
 - fw_dynamic.h, [75](#)
- FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET
 - fw_dynamic.h, [75](#)
- FW_DYNAMIC_INFO_NEXT_MODE_OFFSET
 - fw_dynamic.h, [75](#)
- FW_DYNAMIC_INFO_NEXT_MODE_M
 - fw_dynamic.h, [75](#)
- FW_DYNAMIC_INFO_NEXT_MODE_S
 - fw_dynamic.h, [75](#)
- FW_DYNAMIC_INFO_NEXT_MODE_U
 - fw_dynamic.h, [76](#)
- FW_DYNAMIC_INFO_OPTIONS_OFFSET
 - fw_dynamic.h, [76](#)
- FW_DYNAMIC_INFO_VERSION_MAX
 - fw_dynamic.h, [76](#)
- FW_DYNAMIC_INFO_VERSION_OFFSET
 - fw_dynamic.h, [76](#)
- fdt.c
 - check_block_, [453](#)
 - check_off_, [453](#)
 - fdt_check_header, [453](#)
 - fdt_check_node_offset_, [454](#)
 - fdt_check_prop_offset_, [455](#)
 - fdt_find_string_, [456](#)
 - fdt_first_subnode, [456](#)
 - fdt_header_size_, [457](#)
 - fdt_move, [457](#)
 - fdt_next_node, [458](#)
 - fdt_next_subnode, [459](#)
 - fdt_next_tag, [460](#)
 - fdt_offset_ptr, [461](#)
 - fdt_ro_probe_, [462](#)
- fdt.h
 - FDT_BEGIN_NODE, [463](#)
 - FDT_END_NODE, [463](#)
 - FDT_END, [463](#)
 - FDT_MAGIC, [464](#)
 - FDT_NOP, [464](#)
 - FDT_PROP, [464](#)
 - FDT_TAGSIZE, [464](#)
 - FDT_V16_SIZE, [464](#)
 - FDT_V17_SIZE, [464](#)
 - FDT_V1_SIZE, [464](#)
 - FDT_V2_SIZE, [465](#)
 - FDT_V3_SIZE, [465](#)
- fdt16_t
 - libfdt_env.h, [625](#)
- fdt16_to_cpu
 - libfdt_env.h, [626](#)
- fdt32_ld
 - libfdt.h, [558](#)
- fdt32_t
 - libfdt_env.h, [625](#)
- fdt32_to_cpu
 - libfdt_env.h, [627](#)
- fdt64_ld
 - libfdt.h, [559](#)
- fdt64_t
 - libfdt_env.h, [625](#)
- fdt64_to_cpu
 - libfdt_env.h, [627](#)
- fdt_add_mem_rsv
 - fdt_rw.c, [517](#)
 - libfdt.h, [560](#)
- fdt_add_property_
 - fdt_rw.c, [517](#)
- fdt_add_reservemap_entry
 - fdt_sw.c, [534](#)
 - libfdt.h, [560](#)
- fdt_add_subnode
 - fdt_rw.c, [518](#)
 - libfdt.h, [561](#)
- fdt_add_subnode_namelen
 - fdt_rw.c, [519](#)
 - libfdt.h, [562](#)
- fdt_address_cells
 - fdt_addresses.c, [466](#)
 - libfdt.h, [562](#)

fdt_addresses.c
 fdt_address_cells, 466
 fdt_cells, 466
 fdt_size_cells, 467
fdt_appendprop
 fdt_rw.c, 519
 libfdt.h, 563
fdt_appendprop_cell
 libfdt.h, 563
fdt_appendprop_string
 libfdt.h, 552
fdt_appendprop_u32
 libfdt.h, 564
fdt_appendprop_u64
 libfdt.h, 564
fdt_begin_node
 fdt_sw.c, 534
 libfdt.h, 565
fdt_blocks_misordered_
 fdt_rw.c, 520
fdt_boot_cpuid_phys
 libfdt.h, 552
fdt_cells
 fdt_addresses.c, 466
fdt_check_full
 fdt_ro.c, 483
 libfdt.h, 565
fdt_check_header
 fdt.c, 453
 libfdt.h, 566
fdt_check_node_offset_
 fdt.c, 454
 libfdt_internal.h, 629
fdt_check_prop_offset_
 fdt.c, 455
 libfdt_internal.h, 630
fdt_create
 fdt_sw.c, 535
 libfdt.h, 567
fdt_create_empty_tree
 fdt_empty_tree.c, 468
 libfdt.h, 568
fdt_data_size_
 fdt_rw.c, 520
fdt_del_mem_rsv
 fdt_rw.c, 521
 libfdt.h, 568
fdt_del_node
 fdt_rw.c, 521
 libfdt.h, 568
fdt_delprop
 fdt_rw.c, 522
 libfdt.h, 569
fdt_empty_tree.c
 fdt_create_empty_tree, 468
fdt_end_node
 fdt_sw.c, 535
 libfdt.h, 569
fdt_errtable
 fdt_strerror.c, 531
fdt_find_add_string_
 fdt_rw.c, 522
 fdt_sw.c, 536
fdt_find_string_
 fdt.c, 456
 libfdt_internal.h, 631
fdt_finish
 fdt_sw.c, 537
 libfdt.h, 570
fdt_finish_reservemap
 fdt_sw.c, 537
 libfdt.h, 571
fdt_first_property_offset
 fdt_ro.c, 484
 libfdt.h, 571
fdt_first_subnode
 fdt.c, 456
 libfdt.h, 572
fdt_for_each_property_offset
 libfdt.h, 555
fdt_for_each_subnode
 libfdt.h, 555
fdt_get_alias
 fdt_ro.c, 485
 libfdt.h, 573
fdt_get_alias_namelen
 fdt_ro.c, 486
 libfdt.h, 574
fdt_get_header
 libfdt.h, 556
fdt_get_max_phandle
 fdt_ro.c, 486
 libfdt.h, 574
fdt_get_mem_rsv
 fdt_ro.c, 487
 libfdt.h, 575
fdt_get_name
 fdt_ro.c, 488
 libfdt.h, 576
fdt_get_path
 fdt_ro.c, 488
 libfdt.h, 577
fdt_get_phandle
 fdt_ro.c, 489
 libfdt.h, 578
fdt_get_property
 fdt_ro.c, 490
 libfdt.h, 578
fdt_get_property_by_offset
 fdt_ro.c, 491
 libfdt.h, 579
fdt_get_property_by_offset_
 fdt_ro.c, 491
fdt_get_property_namelen
 fdt_ro.c, 492
 libfdt.h, 580

- fdt_get_property_namelen_
 - fdt_ro.c, [493](#)
- fdt_get_property_w
 - libfdt.h, [580](#)
- fdt_get_string
 - fdt_ro.c, [494](#)
 - libfdt.h, [581](#)
- fdt_getprop
 - fdt_ro.c, [495](#)
 - libfdt.h, [582](#)
- fdt_getprop_by_offset
 - fdt_ro.c, [496](#)
 - libfdt.h, [583](#)
- fdt_getprop_namelen
 - fdt_ro.c, [497](#)
 - libfdt.h, [584](#)
- fdt_getprop_namelen_w
 - libfdt.h, [585](#)
- fdt_getprop_w
 - libfdt.h, [586](#)
- fdt_grab_space_
 - fdt_sw.c, [538](#)
- fdt_header, [45](#)
 - boot_cpuid_phys, [46](#)
 - last_comp_version, [46](#)
 - magic, [46](#)
 - off_dt_strings, [46](#)
 - off_dt_struct, [46](#)
 - off_mem_rsvmap, [46](#)
 - size_dt_strings, [47](#)
 - size_dt_struct, [47](#)
 - totalsize, [47](#)
 - version, [47](#)
- fdt_header_size
 - libfdt.h, [586](#)
- fdt_header_size_
 - fdt.c, [457](#)
 - libfdt.h, [587](#)
- fdt_last_comp_version
 - libfdt.h, [556](#)
- fdt_magic
 - libfdt.h, [556](#)
- fdt_mem_rsv
 - fdt_ro.c, [498](#)
- fdt_mem_rsv_
 - libfdt_internal.h, [631](#)
- fdt_mem_rsv_w_
 - libfdt_internal.h, [632](#)
- fdt_move
 - fdt.c, [457](#)
 - libfdt.h, [587](#)
- fdt_next_node
 - fdt.c, [458](#)
 - libfdt.h, [588](#)
- fdt_next_property_offset
 - fdt_ro.c, [499](#)
 - libfdt.h, [589](#)
- fdt_next_subnode
 - fdt.c, [459](#)
 - libfdt.h, [590](#)
- fdt_next_tag
 - fdt.c, [460](#)
 - libfdt.h, [591](#)
- fdt_node_check_compatible
 - fdt_ro.c, [500](#)
 - libfdt.h, [592](#)
- fdt_node_depth
 - fdt_ro.c, [501](#)
 - libfdt.h, [593](#)
- fdt_node_end_offset_
 - fdt_wip.c, [542](#)
 - libfdt_internal.h, [632](#)
- fdt_node_header, [47](#)
 - name, [47](#)
 - tag, [48](#)
- fdt_node_offset_by_compatible
 - fdt_ro.c, [501](#)
 - libfdt.h, [593](#)
- fdt_node_offset_by_phandle
 - fdt_ro.c, [502](#)
 - libfdt.h, [594](#)
- fdt_node_offset_by_prop_value
 - fdt_ro.c, [503](#)
 - libfdt.h, [595](#)
- fdt_nodename_eq_
 - fdt_ro.c, [504](#)
- fdt_nop_node
 - fdt_wip.c, [543](#)
 - libfdt.h, [596](#)
- fdt_nop_property
 - fdt_wip.c, [544](#)
 - libfdt.h, [597](#)
- fdt_nop_region_
 - fdt_wip.c, [544](#)
- fdt_num_mem_rsv
 - fdt_ro.c, [505](#)
 - libfdt.h, [598](#)
- fdt_off_dt_strings
 - libfdt.h, [557](#)
- fdt_off_dt_struct
 - libfdt.h, [557](#)
- fdt_off_mem_rsvmap
 - libfdt.h, [557](#)
- fdt_offset_ptr
 - fdt.c, [461](#)
 - libfdt.h, [598](#)
- fdt_offset_ptr_
 - libfdt_internal.h, [633](#)
- fdt_offset_ptr_w
 - libfdt.h, [599](#)
- fdt_offset_ptr_w_
 - libfdt_internal.h, [633](#)
- fdt_open_into
 - fdt_rw.c, [523](#)
 - libfdt.h, [600](#)
- fdt_overlay.c

- fdt_overlay_apply, [470](#)
- get_path_len, [471](#)
- overlay_adjust_local_phandles, [471](#)
- overlay_adjust_node_phandles, [472](#)
- overlay_apply_node, [473](#)
- overlay_fixup_one_phandle, [474](#)
- overlay_fixup_phandle, [474](#)
- overlay_fixup_phandles, [475](#)
- overlay_get_target, [476](#)
- overlay_get_target_phandle, [476](#)
- overlay_merge, [477](#)
- overlay_phandle_add_offset, [478](#)
- overlay_symbol_update, [479](#)
- overlay_update_local_node_references, [479](#)
- overlay_update_local_references, [480](#)
- fdt_overlay_apply
 - fdt_overlay.c, [470](#)
 - libfdt.h, [601](#)
- fdt_pack
 - fdt_rw.c, [524](#)
 - libfdt.h, [601](#)
- fdt_packblocks_
 - fdt_rw.c, [524](#)
- fdt_parent_offset
 - fdt_ro.c, [505](#)
 - libfdt.h, [601](#)
- fdt_path_offset
 - fdt_ro.c, [506](#)
 - libfdt.h, [602](#)
- fdt_path_offset_namelen
 - fdt_ro.c, [507](#)
 - libfdt.h, [603](#)
- fdt_property, [48](#)
 - data, [48](#)
 - fdt_sw.c, [539](#)
 - len, [48](#)
 - libfdt.h, [604](#)
 - nameoff, [48](#)
 - tag, [49](#)
- fdt_property_cell
 - libfdt.h, [604](#)
- fdt_property_placeholder
 - fdt_sw.c, [539](#)
 - libfdt.h, [605](#)
- fdt_property_string
 - libfdt.h, [557](#)
- fdt_property_u32
 - libfdt.h, [605](#)
- fdt_property_u64
 - libfdt.h, [606](#)
- fdt_reserve_entry, [49](#)
 - address, [49](#)
 - size, [49](#)
- fdt_resize
 - fdt_sw.c, [540](#)
 - libfdt.h, [606](#)
- fdt_resize_property_
 - fdt_rw.c, [524](#)
- fdt_ro.c
 - fdt_check_full, [483](#)
 - fdt_first_property_offset, [484](#)
 - fdt_get_alias, [485](#)
 - fdt_get_alias_namelen, [486](#)
 - fdt_get_max_phandle, [486](#)
 - fdt_get_mem_rsv, [487](#)
 - fdt_get_name, [488](#)
 - fdt_get_path, [488](#)
 - fdt_get_phandle, [489](#)
 - fdt_get_property, [490](#)
 - fdt_get_property_by_offset, [491](#)
 - fdt_get_property_by_offset_, [491](#)
 - fdt_get_property_namelen, [492](#)
 - fdt_get_property_namelen_, [493](#)
 - fdt_get_string, [494](#)
 - fdt_getprop, [495](#)
 - fdt_getprop_by_offset, [496](#)
 - fdt_getprop_namelen, [497](#)
 - fdt_mem_rsv, [498](#)
 - fdt_next_property_offset, [499](#)
 - fdt_node_check_compatible, [500](#)
 - fdt_node_depth, [501](#)
 - fdt_node_offset_by_compatible, [501](#)
 - fdt_node_offset_by_phandle, [502](#)
 - fdt_node_offset_by_prop_value, [503](#)
 - fdt_nodename_eq_, [504](#)
 - fdt_num_mem_rsv, [505](#)
 - fdt_parent_offset, [505](#)
 - fdt_path_offset, [506](#)
 - fdt_path_offset_namelen, [507](#)
 - fdt_string, [508](#)
 - fdt_string_eq_, [508](#)
 - fdt_stringlist_contains, [509](#)
 - fdt_stringlist_count, [510](#)
 - fdt_stringlist_get, [510](#)
 - fdt_stringlist_search, [511](#)
 - fdt_subnode_offset, [512](#)
 - fdt_subnode_offset_namelen, [512](#)
 - fdt_supernode_atdepth_offset, [513](#)
 - nextprop_, [514](#)
- fdt_ro_probe_
 - fdt.c, [462](#)
 - libfdt_internal.h, [634](#)
- fdt_rw.c
 - FDT_RW_PROBE, [517](#)
 - fdt_add_mem_rsv, [517](#)
 - fdt_add_property_, [517](#)
 - fdt_add_subnode, [518](#)
 - fdt_add_subnode_namelen, [519](#)
 - fdt_appendprop, [519](#)
 - fdt_blocks_misordered_, [520](#)
 - fdt_data_size_, [520](#)
 - fdt_del_mem_rsv, [521](#)
 - fdt_del_node, [521](#)
 - fdt_delprop, [522](#)
 - fdt_find_add_string_, [522](#)
 - fdt_open_into, [523](#)

- fdt_pack, 524
- fdt_packblocks_, 524
- fdt_resize_property_, 524
- fdt_rw_probe_, 525
- fdt_set_name, 525
- fdt_setprop, 526
- fdt_setprop_placeholder, 526
- fdt_splice_, 527
- fdt_splice_mem_rsv_, 528
- fdt_splice_string_, 528
- fdt_splice_struct_, 529
- fdt_rw_probe_
 - fdt_rw.c, 525
- fdt_set_hdr_
 - libfdt.h, 557, 607, 608
- fdt_set_name
 - fdt_rw.c, 525
 - libfdt.h, 608
- fdt_setprop
 - fdt_rw.c, 526
 - libfdt.h, 609
- fdt_setprop_cell
 - libfdt.h, 609
- fdt_setprop_empty
 - libfdt.h, 557
- fdt_setprop_inplace
 - fdt_wip.c, 545
 - libfdt.h, 610
- fdt_setprop_inplace_cell
 - libfdt.h, 610
- fdt_setprop_inplace_namelen_partial
 - fdt_wip.c, 546
 - libfdt.h, 611
- fdt_setprop_inplace_u32
 - libfdt.h, 611
- fdt_setprop_inplace_u64
 - libfdt.h, 612
- fdt_setprop_placeholder
 - fdt_rw.c, 526
 - libfdt.h, 612
- fdt_setprop_string
 - libfdt.h, 558
- fdt_setprop_u32
 - libfdt.h, 613
- fdt_setprop_u64
 - libfdt.h, 614
- fdt_size_cells
 - fdt_addresses.c, 467
 - libfdt.h, 614
- fdt_size_dt_strings
 - libfdt.h, 558
- fdt_size_dt_struct
 - libfdt.h, 558
- fdt_splice_
 - fdt_rw.c, 527
- fdt_splice_mem_rsv_
 - fdt_rw.c, 528
- fdt_splice_string_
 - fdt_rw.c, 528
- fdt_splice_struct_
 - fdt_rw.c, 529
- fdt_strerror
 - fdt_strerror.c, 531
 - libfdt.h, 615
- fdt_strerror.c
 - FDT_ERRTABENT, 531
 - FDT_ERRTABSIZE, 531
 - fdt_errtable, 531
 - fdt_strerror, 531
- fdt_string
 - fdt_ro.c, 508
 - libfdt.h, 615
- fdt_string_eq_
 - fdt_ro.c, 508
- fdt_stringlist_contains
 - fdt_ro.c, 509
 - libfdt.h, 615
- fdt_stringlist_count
 - fdt_ro.c, 510
 - libfdt.h, 616
- fdt_stringlist_get
 - fdt_ro.c, 510
 - libfdt.h, 617
- fdt_stringlist_search
 - fdt_ro.c, 511
 - libfdt.h, 617
- fdt_subnode_offset
 - fdt_ro.c, 512
 - libfdt.h, 618
- fdt_subnode_offset_namelen
 - fdt_ro.c, 512
 - libfdt.h, 619
- fdt_supernode_atdepth_offset
 - fdt_ro.c, 513
 - libfdt.h, 619
- fdt_sw.c
 - FDT_SW_PROBE_MEMRSV, 533
 - FDT_SW_PROBE_STRUCT, 533
 - FDT_SW_PROBE, 533
 - fdt_add_reservemap_entry, 534
 - fdt_begin_node, 534
 - fdt_create, 535
 - fdt_end_node, 535
 - fdt_find_add_string_, 536
 - fdt_finish, 537
 - fdt_finish_reservemap, 537
 - fdt_grab_space_, 538
 - fdt_property, 539
 - fdt_property_placeholder, 539
 - fdt_resize, 540
 - fdt_sw_probe_, 540
 - fdt_sw_probe_memrsv_, 541
 - fdt_sw_probe_struct_, 541
- fdt_sw_probe_
 - fdt_sw.c, 540
- fdt_sw_probe_memrsv_

- fdt_sw.c, [541](#)
- fdt_sw_probe_struct
 - fdt_sw.c, [541](#)
- fdt_totalsize
 - libfdt.h, [558](#)
- fdt_version
 - libfdt.h, [558](#)
- fdt_wip.c
 - fdt_node_end_offset, [542](#)
 - fdt_nop_node, [543](#)
 - fdt_nop_property, [544](#)
 - fdt_nop_region, [544](#)
 - fdt_setprop_inplace, [545](#)
 - fdt_setprop_inplace_namelen_partial, [546](#)
- features
 - sbi_platform, [56](#)
- ffs
 - sbi_bitops.h, [179](#)
- ffz
 - sbi_bitops.h, [179](#)
- final_exit
 - sbi_platform_operations, [58](#)
- final_init
 - sbi_platform_operations, [58](#)
- firmware_context
 - sbi_platform, [56](#)
- fls
 - sbi_bitops.h, [180](#)
- format
 - sbi_console.h, [186](#)
- fp_init
 - sbi_hart.c, [380](#)
- fw_dynamic.h
 - __packed, [76](#)
 - FW_DYNAMIC_INFO_BOOT_HART_OFFSET, [75](#)
 - FW_DYNAMIC_INFO_MAGIC_OFFSET, [75](#)
 - FW_DYNAMIC_INFO_MAGIC_VALUE, [75](#)
 - FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET, [75](#)
 - FW_DYNAMIC_INFO_NEXT_MODE_OFFSET, [75](#)
 - FW_DYNAMIC_INFO_NEXT_MODE_M, [75](#)
 - FW_DYNAMIC_INFO_NEXT_MODE_S, [75](#)
 - FW_DYNAMIC_INFO_NEXT_MODE_U, [76](#)
 - FW_DYNAMIC_INFO_OPTIONS_OFFSET, [76](#)
 - FW_DYNAMIC_INFO_VERSION_MAX, [76](#)
 - FW_DYNAMIC_INFO_VERSION_OFFSET, [76](#)
- fw_dynamic_info, [49](#)
 - boot_hart, [50](#)
 - magic, [50](#)
 - next_addr, [50](#)
 - next_mode, [50](#)
 - options, [50](#)
 - version, [51](#)
- fw_size
 - sbi_scratch, [62](#)
- fw_start
 - sbi_scratch, [62](#)
- GET_PRECISION
 - riscv_fp.h, [165](#)
- GET_RS1
 - riscv_encoding.h, [135](#)
- GET_RS1S
 - riscv_encoding.h, [135](#)
- GET_RS2
 - riscv_encoding.h, [135](#)
- GET_RS2C
 - riscv_encoding.h, [136](#)
- GET_RS2S
 - riscv_encoding.h, [136](#)
- GET_RM
 - riscv_encoding.h, [135](#)
 - riscv_fp.h, [165](#)
- GET_SP
 - riscv_encoding.h, [136](#)
- get_path_len
 - fdt_overlay.c, [471](#)
- get_reg
 - sifive-uart.c, [638](#)
 - uart8250.c, [645](#)
- get_ticks
 - sbi_timer.c, [423](#)
- get_tlbr_flush_limit
 - sbi_platform_operations, [59](#)
- gp
 - sbi_trap_regs, [68](#)
- h2s
 - sbi_hart.c, [379](#)
- HSTATUS_SP2P
 - riscv_encoding.h, [136](#)
- HSTATUS_SP2V
 - riscv_encoding.h, [136](#)
- HSTATUS_SPRV
 - riscv_encoding.h, [136](#)
- HSTATUS_SPV
 - riscv_encoding.h, [136](#)
- HSTATUS_VTSR
 - riscv_encoding.h, [137](#)
- HSTATUS_VTVM
 - riscv_encoding.h, [137](#)
- HTIF_CMD_BITS
 - htif.c, [656](#)
- HTIF_CMD_MASK
 - htif.c, [656](#)
- HTIF_CMD_SHIFT
 - htif.c, [656](#)
- HTIF_CONSOLE_CMD_GETC
 - htif.c, [656](#)
- HTIF_CONSOLE_CMD_PUTC
 - htif.c, [656](#)
- HTIF_DATA_BITS
 - htif.c, [656](#)
- HTIF_DATA_MASK
 - htif.c, [657](#)
- HTIF_DATA_SHIFT
 - htif.c, [657](#)
- HTIF_DEV_BITS

- htif.c, 657
- HTIF_DEV_CONSOLE
 - htif.c, 657
- HTIF_DEV_MASK
 - htif.c, 657
- HTIF_DEV_SHIFT
 - htif.c, 657
- HTIF_DEV_SYSTEM
 - htif.c, 657
- handle
 - sbi_ecall_extension, 52
- hart_count
 - sbi_platform, 56
- hart_stack_size
 - sbi_platform, 56
- hartid_to_scratch
 - sbi_scratch, 62
- head
 - sbi_ecall_extension, 52
- htif.c
 - __attribute__, 658
 - __check_fromhost, 658
 - __set_tohost, 658
 - FROMHOST_CMD, 655
 - FROMHOST_DATA, 655
 - FROMHOST_DEV, 656
 - HTIF_CMD_BITS, 656
 - HTIF_CMD_MASK, 656
 - HTIF_CMD_SHIFT, 656
 - HTIF_CONSOLE_CMD_GETC, 656
 - HTIF_CONSOLE_CMD_PUTC, 656
 - HTIF_DATA_BITS, 656
 - HTIF_DATA_MASK, 657
 - HTIF_DATA_SHIFT, 657
 - HTIF_DEV_BITS, 657
 - HTIF_DEV_CONSOLE, 657
 - HTIF_DEV_MASK, 657
 - HTIF_DEV_SHIFT, 657
 - HTIF_DEV_SYSTEM, 657
 - htif_console_buf, 660
 - htif_getc, 659
 - htif_lock, 660
 - htif_putc, 660
 - htif_system_down, 660
 - PK_SYS_write, 657
 - TOHOST_CMD, 658
- htif.h
 - htif_getc, 324
 - htif_putc, 325
 - htif_system_down, 325
- htif_console_buf
 - htif.c, 660
- htif_getc
 - htif.c, 659
 - htif.h, 324
- htif_lock
 - htif.c, 660
- htif_putc
 - htif.c, 660
 - htif.h, 325
- htif_system_down
 - htif.c, 660
 - htif.h, 325
- IMM_I
 - riscv_encoding.h, 137
- IMM_S
 - riscv_encoding.h, 137
- INSERT_FIELD
 - sbi_bits.h, 181
- INSN_16BIT_MASK
 - riscv_encoding.h, 137
- INSN_32BIT_MASK
 - riscv_encoding.h, 137
- INSN_IS_16BIT
 - riscv_encoding.h, 137
- INSN_IS_32BIT
 - riscv_encoding.h, 138
- INSN_LEN
 - riscv_encoding.h, 138
- INSN_MASK_C_FLDSP
 - riscv_encoding.h, 138
- INSN_MASK_C_FLWSP
 - riscv_encoding.h, 138
- INSN_MASK_C_FLD
 - riscv_encoding.h, 138
- INSN_MASK_C_FLW
 - riscv_encoding.h, 138
- INSN_MASK_C_FSDSP
 - riscv_encoding.h, 139
- INSN_MASK_C_FSWSP
 - riscv_encoding.h, 139
- INSN_MASK_C_FSD
 - riscv_encoding.h, 139
- INSN_MASK_C_FSW
 - riscv_encoding.h, 139
- INSN_MASK_C_LDSP
 - riscv_encoding.h, 139
- INSN_MASK_C_LWSP
 - riscv_encoding.h, 139
- INSN_MASK_C_LD
 - riscv_encoding.h, 139
- INSN_MASK_C_LW
 - riscv_encoding.h, 139
- INSN_MASK_C_SDSP
 - riscv_encoding.h, 140
- INSN_MASK_C_SWSP
 - riscv_encoding.h, 140
- INSN_MASK_C_SD
 - riscv_encoding.h, 140
- INSN_MASK_C_SW
 - riscv_encoding.h, 140
- INSN_MASK_FLD
 - riscv_encoding.h, 140
- INSN_MASK_FLQ
 - riscv_encoding.h, 140
- INSN_MASK_FLW

- riscv_encoding.h, 140
- INSN_MASK_FSD
 - riscv_encoding.h, 140
- INSN_MASK_FSQ
 - riscv_encoding.h, 141
- INSN_MASK_FSW
 - riscv_encoding.h, 141
- INSN_MASK_LBU
 - riscv_encoding.h, 141
- INSN_MASK_LHU
 - riscv_encoding.h, 141
- INSN_MASK_LWU
 - riscv_encoding.h, 142
- INSN_MASK_LB
 - riscv_encoding.h, 141
- INSN_MASK_LD
 - riscv_encoding.h, 141
- INSN_MASK_LH
 - riscv_encoding.h, 141
- INSN_MASK_LW
 - riscv_encoding.h, 141
- INSN_MASK_SB
 - riscv_encoding.h, 142
- INSN_MASK_SD
 - riscv_encoding.h, 142
- INSN_MASK_SH
 - riscv_encoding.h, 142
- INSN_MASK_SW
 - riscv_encoding.h, 142
- INSN_MASK_WFI
 - riscv_encoding.h, 142
- INSN_MATCH_C_FLDSP
 - riscv_encoding.h, 142
- INSN_MATCH_C_FLWSP
 - riscv_encoding.h, 143
- INSN_MATCH_C_FLD
 - riscv_encoding.h, 142
- INSN_MATCH_C_FLW
 - riscv_encoding.h, 143
- INSN_MATCH_C_FSDSP
 - riscv_encoding.h, 143
- INSN_MATCH_C_FSWSP
 - riscv_encoding.h, 143
- INSN_MATCH_C_FSD
 - riscv_encoding.h, 143
- INSN_MATCH_C_FSW
 - riscv_encoding.h, 143
- INSN_MATCH_C_LDSP
 - riscv_encoding.h, 143
- INSN_MATCH_C_LWSP
 - riscv_encoding.h, 144
- INSN_MATCH_C_LD
 - riscv_encoding.h, 143
- INSN_MATCH_C_LW
 - riscv_encoding.h, 144
- INSN_MATCH_C_SDSP
 - riscv_encoding.h, 144
- INSN_MATCH_C_SWSP
 - riscv_encoding.h, 144
- INSN_MATCH_C_SD
 - riscv_encoding.h, 144
- INSN_MATCH_C_SW
 - riscv_encoding.h, 144
- INSN_MATCH_FLD
 - riscv_encoding.h, 144
- INSN_MATCH_FLQ
 - riscv_encoding.h, 144
- INSN_MATCH_FLW
 - riscv_encoding.h, 145
- INSN_MATCH_FSD
 - riscv_encoding.h, 145
- INSN_MATCH_FSQ
 - riscv_encoding.h, 145
- INSN_MATCH_FSW
 - riscv_encoding.h, 145
- INSN_MATCH_LBU
 - riscv_encoding.h, 145
- INSN_MATCH_LHU
 - riscv_encoding.h, 146
- INSN_MATCH_LWU
 - riscv_encoding.h, 146
- INSN_MATCH_LB
 - riscv_encoding.h, 145
- INSN_MATCH_LD
 - riscv_encoding.h, 145
- INSN_MATCH_LH
 - riscv_encoding.h, 145
- INSN_MATCH_LW
 - riscv_encoding.h, 146
- INSN_MATCH_SB
 - riscv_encoding.h, 146
- INSN_MATCH_SD
 - riscv_encoding.h, 146
- INSN_MATCH_SH
 - riscv_encoding.h, 146
- INSN_MATCH_SW
 - riscv_encoding.h, 146
- INSN_MATCH_WFI
 - riscv_encoding.h, 146
- INT_MAX
 - libfdt_env.h, 623
- IRQ_M_EXT
 - riscv_encoding.h, 147
- IRQ_M_SOFT
 - riscv_encoding.h, 147
- IRQ_M_TIMER
 - riscv_encoding.h, 147
- IRQ_S_EXT
 - riscv_encoding.h, 147
- IRQ_S_GEXT
 - riscv_encoding.h, 147
- IRQ_S_SOFT
 - riscv_encoding.h, 147
- IRQ_S_TIMER
 - riscv_encoding.h, 147
- IRQ_VS_EXT

- riscv_encoding.h, 147
- IRQ_VS_SOFT
 - riscv_encoding.h, 148
- IRQ_VS_TIMER
 - riscv_encoding.h, 148
- illegal_insn_func
 - sbi_illegal_insn.c, 390
- illegal_insn_table
 - sbi_illegal_insn.c, 393
- include/sbi/fw_dynamic.h, 74
- include/sbi/riscv_asm.h, 77
- include/sbi/riscv_atomic.h, 86
- include/sbi/riscv_barrier.h, 90
- include/sbi/riscv_encoding.h, 93
- include/sbi/riscv_fp.h, 164
- include/sbi/riscv_io.h, 165
- include/sbi/riscv_locks.h, 172
- include/sbi/sbi_bitops.h, 177
- include/sbi/sbi_bits.h, 180
- include/sbi/sbi_console.h, 181
- include/sbi/sbi_const.h, 186
- include/sbi/sbi_ecall.h, 189
- include/sbi/sbi_ecall_interface.h, 194
- include/sbi/sbi_emulate_csr.h, 200
- include/sbi/sbi_error.h, 202
- include/sbi/sbi_fifo.h, 205
- include/sbi/sbi_hart.h, 211
- include/sbi/sbi_hfence.h, 220
- include/sbi/sbi_illegal_insn.h, 223
- include/sbi/sbi_init.h, 225
- include/sbi/sbi_ipi.h, 229
- include/sbi/sbi_list.h, 236
- include/sbi/sbi_misaligned_ldst.h, 243
- include/sbi/sbi_platform.h, 246
- include/sbi/sbi_scratch.h, 270
- include/sbi/sbi_string.h, 276
- include/sbi/sbi_system.h, 280
- include/sbi/sbi_timer.h, 284
- include/sbi/sbi_tlb.h, 289
- include/sbi/sbi_trap.h, 293
- include/sbi/sbi_types.h, 302
- include/sbi/sbi_unpriv.h, 309
- include/sbi/sbi_version.h, 311
- include/sbi_utils/irqchip/plic.h, 312
- include/sbi_utils/serial/sifive-uart.h, 315
- include/sbi_utils/serial/uart8250.h, 317
- include/sbi_utils/sys/clint.h, 320
- include/sbi_utils/sys/htif.h, 324
- init_coldboot
 - sbi_init.c, 394
- init_count_offset
 - sbi_init.c, 400
- init_warmboot
 - sbi_init.c, 395
- int16_t
 - sbi_types.h, 306
- int32_t
 - sbi_types.h, 306
- ipi_clear
 - sbi_platform_operations, 59
- ipi_data_off
 - sbi_ipi.c, 408
- ipi_exit
 - sbi_platform_operations, 59
- ipi_halt_event
 - sbi_ipi.c, 408
- ipi_halt_ops
 - sbi_ipi.c, 408
- ipi_init
 - sbi_platform_operations, 59
- ipi_ops_array
 - sbi_ipi.c, 409
- ipi_send
 - sbi_platform_operations, 59
- ipi_smode_event
 - sbi_ipi.c, 409
- ipi_smode_ops
 - sbi_ipi.c, 409
- irqchip_exit
 - sbi_platform_operations, 59
- irqchip_init
 - sbi_platform_operations, 59
- LGREG
 - riscv_asm.h, 80
- LOG_REGBYTES
 - riscv_encoding.h, 148
- last_comp_version
 - fdt_header, 46
- len
 - fdt_property, 48
- lib/sbi/riscv_asm.c, 326
- lib/sbi/riscv_atomic.c, 330
- lib/sbi/riscv_locks.c, 337
- lib/sbi/sbi_console.c, 342
- lib/sbi/sbi_ecall.c, 353
- lib/sbi/sbi_ecall_base.c, 358
- lib/sbi/sbi_ecall_legacy.c, 360
- lib/sbi/sbi_ecall_replace.c, 362
- lib/sbi/sbi_ecall_vendor.c, 366
- lib/sbi/sbi_emulate_csr.c, 368
- lib/sbi/sbi_fifo.c, 370
- lib/sbi/sbi_hart.c, 378
- lib/sbi/sbi_illegal_insn.c, 390
- lib/sbi/sbi_init.c, 393
- lib/sbi/sbi_ipi.c, 400
- lib/sbi/sbi_misaligned_ldst.c, 410
- lib/sbi/sbi_scratch.c, 412
- lib/sbi/sbi_string.c, 414
- lib/sbi/sbi_system.c, 418
- lib/sbi/sbi_timer.c, 422
- lib/sbi/sbi_tlb.c, 428
- lib/sbi/sbi_trap.c, 440
- lib/sbi/sbi_unpriv.c, 443
- lib/utis/irqchip/plic.c, 447
- lib/utis/libfdt/fdt.c, 452
- lib/utis/libfdt/fdt.h, 463

lib/utils/libfdt/fdt_addresses.c, [465](#)
 lib/utils/libfdt/fdt_empty_tree.c, [467](#)
 lib/utils/libfdt/fdt_overlay.c, [469](#)
 lib/utils/libfdt/fdt_ro.c, [481](#)
 lib/utils/libfdt/fdt_rw.c, [515](#)
 lib/utils/libfdt/fdt_strerror.c, [530](#)
 lib/utils/libfdt/fdt_sw.c, [532](#)
 lib/utils/libfdt/fdt_wip.c, [542](#)
 lib/utils/libfdt/libfdt.h, [546](#)
 lib/utils/libfdt/libfdt_env.h, [620](#)
 lib/utils/libfdt/libfdt_internal.h, [628](#)
 lib/utils/serial/sifive-uart.c, [635](#)
 lib/utils/serial/uart8250.c, [641](#)
 lib/utils/sys/clint.c, [648](#)
 lib/utils/sys/htif.c, [654](#)
 libfdt.h
 FDT_ERR_BADLAYOUT, [552](#)
 FDT_ERR_BADMAGIC, [553](#)
 FDT_ERR_BADNCELLS, [553](#)
 FDT_ERR_BADOFFSET, [553](#)
 FDT_ERR_BADOVERLAY, [553](#)
 FDT_ERR_BADPATH, [553](#)
 FDT_ERR_BADPHANDLE, [553](#)
 FDT_ERR_BADSTATE, [553](#)
 FDT_ERR_BADSTRUCTURE, [553](#)
 FDT_ERR_BADVALUE, [554](#)
 FDT_ERR_BADVERSION, [554](#)
 FDT_ERR_EXISTS, [554](#)
 FDT_ERR_INTERNAL, [554](#)
 FDT_ERR_MAX, [554](#)
 FDT_ERR_NOPHANDLES, [554](#)
 FDT_ERR_NOSPACE, [554](#)
 FDT_ERR_NOTFOUND, [554](#)
 FDT_ERR_TRUNCATED, [555](#)
 FDT_FIRST_SUPPORTED_VERSION, [555](#)
 FDT_LAST_SUPPORTED_VERSION, [556](#)
 FDT_MAX_NCELLS, [556](#)
 fdt32_ld, [558](#)
 fdt64_ld, [559](#)
 fdt_add_mem_rsv, [560](#)
 fdt_add_reservemap_entry, [560](#)
 fdt_add_subnode, [561](#)
 fdt_add_subnode_namelen, [562](#)
 fdt_address_cells, [562](#)
 fdt_appendprop, [563](#)
 fdt_appendprop_cell, [563](#)
 fdt_appendprop_string, [552](#)
 fdt_appendprop_u32, [564](#)
 fdt_appendprop_u64, [564](#)
 fdt_begin_node, [565](#)
 fdt_boot_cpuid_phys, [552](#)
 fdt_check_full, [565](#)
 fdt_check_header, [566](#)
 fdt_create, [567](#)
 fdt_create_empty_tree, [568](#)
 fdt_del_mem_rsv, [568](#)
 fdt_del_node, [568](#)
 fdt_delprop, [569](#)
 fdt_end_node, [569](#)
 fdt_finish, [570](#)
 fdt_finish_reservemap, [571](#)
 fdt_first_property_offset, [571](#)
 fdt_first_subnode, [572](#)
 fdt_for_each_property_offset, [555](#)
 fdt_for_each_subnode, [555](#)
 fdt_get_alias, [573](#)
 fdt_get_alias_namelen, [574](#)
 fdt_get_header, [556](#)
 fdt_get_max_phandle, [574](#)
 fdt_get_mem_rsv, [575](#)
 fdt_get_name, [576](#)
 fdt_get_path, [577](#)
 fdt_get_phandle, [578](#)
 fdt_get_property, [578](#)
 fdt_get_property_by_offset, [579](#)
 fdt_get_property_namelen, [580](#)
 fdt_get_property_w, [580](#)
 fdt_get_string, [581](#)
 fdt_getprop, [582](#)
 fdt_getprop_by_offset, [583](#)
 fdt_getprop_namelen, [584](#)
 fdt_getprop_namelen_w, [585](#)
 fdt_getprop_w, [586](#)
 fdt_header_size, [586](#)
 fdt_header_size_, [587](#)
 fdt_last_comp_version, [556](#)
 fdt_magic, [556](#)
 fdt_move, [587](#)
 fdt_next_node, [588](#)
 fdt_next_property_offset, [589](#)
 fdt_next_subnode, [590](#)
 fdt_next_tag, [591](#)
 fdt_node_check_compatible, [592](#)
 fdt_node_depth, [593](#)
 fdt_node_offset_by_compatible, [593](#)
 fdt_node_offset_by_phandle, [594](#)
 fdt_node_offset_by_prop_value, [595](#)
 fdt_nop_node, [596](#)
 fdt_nop_property, [597](#)
 fdt_num_mem_rsv, [598](#)
 fdt_off_dt_strings, [557](#)
 fdt_off_dt_struct, [557](#)
 fdt_off_mem_rsvmap, [557](#)
 fdt_offset_ptr, [598](#)
 fdt_offset_ptr_w, [599](#)
 fdt_open_into, [600](#)
 fdt_overlay_apply, [601](#)
 fdt_pack, [601](#)
 fdt_parent_offset, [601](#)
 fdt_path_offset, [602](#)
 fdt_path_offset_namelen, [603](#)
 fdt_property, [604](#)
 fdt_property_cell, [604](#)
 fdt_property_placeholder, [605](#)
 fdt_property_string, [557](#)
 fdt_property_u32, [605](#)

- fdt_property_u64, 606
- fdt_resize, 606
- fdt_set_hdr_, 557, 607, 608
- fdt_set_name, 608
- fdt_setprop, 609
- fdt_setprop_cell, 609
- fdt_setprop_empty, 557
- fdt_setprop_inplace, 610
- fdt_setprop_inplace_cell, 610
- fdt_setprop_inplace_namelen_partial, 611
- fdt_setprop_inplace_u32, 611
- fdt_setprop_inplace_u64, 612
- fdt_setprop_placeholder, 612
- fdt_setprop_string, 558
- fdt_setprop_u32, 613
- fdt_setprop_u64, 614
- fdt_size_cells, 614
- fdt_size_dt_strings, 558
- fdt_size_dt_struct, 558
- fdt_strerror, 615
- fdt_string, 615
- fdt_stringlist_contains, 615
- fdt_stringlist_count, 616
- fdt_stringlist_get, 617
- fdt_stringlist_search, 617
- fdt_subnode_offset, 618
- fdt_subnode_offset_namelen, 619
- fdt_supernode_atdepth_offset, 619
- fdt_totalsize, 558
- fdt_version, 558
- libfdt_env.h
 - CPU_TO_FDT16, 622
 - CPU_TO_FDT32, 622
 - CPU_TO_FDT64, 622
 - cpu_to_fdt16, 625
 - cpu_to_fdt32, 625
 - cpu_to_fdt64, 626
 - EXTRACT_BYTE, 622
 - FDT_BITWISE, 623
 - FDT_FORCE, 623
 - fdt16_t, 625
 - fdt16_to_cpu, 626
 - fdt32_t, 625
 - fdt32_to_cpu, 627
 - fdt64_t, 625
 - fdt64_to_cpu, 627
 - INT_MAX, 623
 - memchr, 623
 - memcmp, 623
 - memcpy, 623
 - memmove, 623
 - memset, 624
 - strchr, 624
 - strcmp, 624
 - strcpy, 624
 - strlen, 624
 - strnlen, 624
 - strrchr, 624
 - UINT_MAX, 624
- libfdt_internal.h
 - FDT_ALIGN, 628
 - FDT_RO_PROBE, 629
 - FDT_SW_MAGIC, 629
 - FDT_TAGALIGN, 629
 - fdt_check_node_offset_, 629
 - fdt_check_prop_offset_, 630
 - fdt_find_string_, 631
 - fdt_mem_rsv_, 631
 - fdt_mem_rsv_w_, 632
 - fdt_node_end_offset_, 632
 - fdt_offset_ptr_, 633
 - fdt_offset_ptr_w_, 633
 - fdt_ro_probe_, 634
- likely
 - sbi_types.h, 304
- lock
 - spinlock_t, 71
- log2roundup
 - sbi_hart.c, 380
- MASK_FUNC3
 - riscv_encoding.h, 148
- MAX
 - sbi_types.h, 304
- MIP_MEIP
 - riscv_encoding.h, 148
- MIP_MSIP
 - riscv_encoding.h, 148
- MIP_MTIP
 - riscv_encoding.h, 148
- MIP_SEIP
 - riscv_encoding.h, 148
- MIP_SGEIP
 - riscv_encoding.h, 149
- MIP_SSIP
 - riscv_encoding.h, 149
- MIP_STIP
 - riscv_encoding.h, 149
- MIP_VSEIP
 - riscv_encoding.h, 149
- MIP_VSSIP
 - riscv_encoding.h, 149
- MIP_VSTIP
 - riscv_encoding.h, 149
- MIN
 - sbi_types.h, 304
- MSTATUS32_SD
 - riscv_encoding.h, 149
- MSTATUS64_SD
 - riscv_encoding.h, 150
- MSTATUS_FS
 - riscv_encoding.h, 150
- MSTATUS_MIE
 - riscv_encoding.h, 150
- MSTATUS_MPIE
 - riscv_encoding.h, 150
- MSTATUS_MPP_SHIFT

- riscv_encoding.h, 150
- MSTATUS_MPRV
 - riscv_encoding.h, 150
- MSTATUS_MPP
 - riscv_encoding.h, 150
- MSTATUS_MXR
 - riscv_encoding.h, 150
- MSTATUS_SIE
 - riscv_encoding.h, 151
- MSTATUS_SPIE_SHIFT
 - riscv_encoding.h, 151
- MSTATUS_SPIE
 - riscv_encoding.h, 151
- MSTATUS_SPP_SHIFT
 - riscv_encoding.h, 151
- MSTATUS_SPP
 - riscv_encoding.h, 151
- MSTATUS_SUM
 - riscv_encoding.h, 151
- MSTATUS_SD
 - riscv_encoding.h, 151
- MSTATUS_TSR
 - riscv_encoding.h, 151
- MSTATUS_TVM
 - riscv_encoding.h, 152
- MSTATUS_TW
 - riscv_encoding.h, 152
- MSTATUS_UBE
 - riscv_encoding.h, 152
- MSTATUS_XS
 - riscv_encoding.h, 152
- MSTATUSH_MBE
 - riscv_encoding.h, 152
- MSTATUSH_MPV
 - riscv_encoding.h, 152
- MSTATUSH_SBE
 - riscv_encoding.h, 152
- magic
 - fdt_header, 46
 - fw_dynamic_info, 50
- mb
 - riscv_barrier.h, 92
- memchr
 - libfdt_env.h, 623
- memcmp
 - libfdt_env.h, 623
- memcpy
 - libfdt_env.h, 623
- memmove
 - libfdt_env.h, 623
- memset
 - libfdt_env.h, 624
- mepc
 - sbi_trap_regs, 68
- misa_check_extension
 - sbi_platform_operations, 59
- misa_extension
 - riscv_asm.h, 80
- misa_extension_imp
 - riscv_asm.c, 327
 - riscv_asm.h, 82
- misa_get_xlen
 - sbi_platform_operations, 60
- misa_string
 - riscv_asm.h, 83
- misa_xlen
 - riscv_asm.c, 328
 - riscv_asm.h, 83
- mstatus
 - sbi_trap_regs, 68
- mstatus_init
 - sbi_hart.c, 380
- mstatusH
 - sbi_trap_regs, 68
- NULL
 - sbi_types.h, 304
- name
 - fdt_node_header, 47
 - sbi_ipi_event_ops, 54
 - sbi_platform, 56
- nameoff
 - fdt_property, 48
- next
 - sbi_dlist, 51
- next_addr
 - fw_dynamic_info, 50
 - sbi_hart.h, 219
 - sbi_scratch, 62
- next_arg1
 - sbi_scratch, 62
- next_mode
 - fw_dynamic_info, 50
 - sbi_hart.h, 219
 - sbi_scratch, 63
- next_virt
 - sbi_hart.h, 219
- nextprop_
 - fdt_ro.c, 514
- num_entries
 - sbi_fifo, 53
- OPENSBI_VERSION_MAJOR
 - sbi_version.h, 311
- OPENSBI_VERSION_MINOR
 - sbi_version.h, 312
- OPENSBI_VERSION
 - sbi_version.h, 311
- off_dt_strings
 - fdt_header, 46
- off_dt_struct
 - fdt_header, 46
- off_mem_rsvmap
 - fdt_header, 46
- offsetof
 - sbi_types.h, 305
- opensbi_version

- sbi_platform, [56](#)
- options
 - fw_dynamic_info, [50](#)
 - sbi_scratch, [63](#)
- out_sz
 - sbi_console.h, [186](#)
- overlay_adjust_local_phandles
 - fdt_overlay.c, [471](#)
- overlay_adjust_node_phandles
 - fdt_overlay.c, [472](#)
- overlay_apply_node
 - fdt_overlay.c, [473](#)
- overlay_fixup_one_phandle
 - fdt_overlay.c, [474](#)
- overlay_fixup_phandle
 - fdt_overlay.c, [474](#)
- overlay_fixup_phandles
 - fdt_overlay.c, [475](#)
- overlay_get_target
 - fdt_overlay.c, [476](#)
- overlay_get_target_phandle
 - fdt_overlay.c, [476](#)
- overlay_merge
 - fdt_overlay.c, [477](#)
- overlay_phandle_add_offset
 - fdt_overlay.c, [478](#)
- overlay_symbol_update
 - fdt_overlay.c, [479](#)
- overlay_update_local_node_references
 - fdt_overlay.c, [479](#)
- overlay_update_local_references
 - fdt_overlay.c, [480](#)
- PAD_ALTERNATE
 - sbi_console.c, [344](#)
- PAD_RIGHT
 - sbi_console.c, [344](#)
- PAD_ZERO
 - sbi_console.c, [344](#)
- PAGE_MASK
 - riscv_asm.h, [80](#)
- PAGE_SHIFT
 - riscv_asm.h, [81](#)
- PAGE_SIZE
 - riscv_asm.h, [81](#)
- PK_SYS_write
 - htif.c, [657](#)
- PLIC_CONTEXT_BASE
 - plic.c, [448](#)
- PLIC_CONTEXT_STRIDE
 - plic.c, [448](#)
- PLIC_ENABLE_BASE
 - plic.c, [448](#)
- PLIC_ENABLE_STRIDE
 - plic.c, [448](#)
- PLIC_PENDING_BASE
 - plic.c, [448](#)
- PLIC_PRIORITY_BASE
 - plic.c, [448](#)
- PMP_A_NA4
 - riscv_encoding.h, [153](#)
- PMP_A_NAPOT
 - riscv_encoding.h, [153](#)
- PMP_A_TOR
 - riscv_encoding.h, [153](#)
- PMP_COUNT
 - riscv_encoding.h, [153](#)
- PMP_SHIFT
 - riscv_encoding.h, [153](#)
- PMP_A
 - riscv_encoding.h, [152](#)
- PMP_L
 - riscv_encoding.h, [153](#)
- PMP_R
 - riscv_encoding.h, [153](#)
- PMP_W
 - riscv_encoding.h, [153](#)
- PMP_X
 - riscv_encoding.h, [154](#)
- PRECISION_D
 - riscv_fp.h, [165](#)
- PRECISION_S
 - riscv_fp.h, [165](#)
- PRINT_BUF_LEN
 - sbi_console.c, [344](#)
- PRV_M
 - riscv_encoding.h, [154](#)
- PRV_S
 - riscv_encoding.h, [154](#)
- PRV_U
 - riscv_encoding.h, [154](#)
- PTE_PPN_SHIFT
 - riscv_encoding.h, [154](#)
- PTE_SOFT
 - riscv_encoding.h, [155](#)
- PTE_TABLE
 - riscv_encoding.h, [155](#)
- PTE_A
 - riscv_encoding.h, [154](#)
- PTE_D
 - riscv_encoding.h, [154](#)
- PTE_G
 - riscv_encoding.h, [154](#)
- PTE_R
 - riscv_encoding.h, [155](#)
- PTE_U
 - riscv_encoding.h, [155](#)
- PTE_V
 - riscv_encoding.h, [155](#)
- PTE_W
 - riscv_encoding.h, [155](#)
- PTE_X
 - riscv_encoding.h, [155](#)
- physical_addr_t
 - sbi_types.h, [306](#)
- physical_size_t
 - sbi_types.h, [306](#)

- platform_addr
 - sbi_scratch, [63](#)
- platform_ops_addr
 - sbi_platform, [56](#)
- platform_version
 - sbi_platform, [57](#)
- plic.c
 - PLIC_CONTEXT_BASE, [448](#)
 - PLIC_CONTEXT_STRIDE, [448](#)
 - PLIC_ENABLE_BASE, [448](#)
 - PLIC_ENABLE_STRIDE, [448](#)
 - PLIC_PENDING_BASE, [448](#)
 - PLIC_PRIORITY_BASE, [448](#)
 - plic_base, [451](#)
 - plic_cold_irqchip_init, [448](#)
 - plic_fdt_fixup, [449](#)
 - plic_hart_count, [451](#)
 - plic_num_sources, [451](#)
 - plic_set_ie, [449](#)
 - plic_set_priority, [450](#)
 - plic_set_thresh, [450](#)
 - plic_warm_irqchip_init, [450](#)
- plic.h
 - plic_cold_irqchip_init, [313](#)
 - plic_fdt_fixup, [313](#)
 - plic_set_ie, [313](#)
 - plic_set_thresh, [314](#)
 - plic_warm_irqchip_init, [314](#)
- plic_base
 - plic.c, [451](#)
- plic_cold_irqchip_init
 - plic.c, [448](#)
 - plic.h, [313](#)
- plic_fdt_fixup
 - plic.c, [449](#)
 - plic.h, [313](#)
- plic_hart_count
 - plic.c, [451](#)
- plic_num_sources
 - plic.c, [451](#)
- plic_set_ie
 - plic.c, [449](#)
 - plic.h, [313](#)
- plic_set_priority
 - plic.c, [450](#)
- plic_set_thresh
 - plic.c, [450](#)
 - plic.h, [314](#)
- plic_warm_irqchip_init
 - plic.c, [450](#)
 - plic.h, [314](#)
- pmp_get
 - riscv_asm.c, [328](#)
 - riscv_asm.h, [84](#)
- pmp_init
 - sbi_hart.c, [381](#)
- pmp_region_count
 - sbi_platform_operations, [60](#)
- pmp_region_info
 - sbi_platform_operations, [60](#)
- pmp_set
 - riscv_asm.c, [329](#)
 - riscv_asm.h, [85](#)
- prev
 - sbi_dlist, [51](#)
- print
 - sbi_console.c, [345](#)
- putc
 - sbi_console.c, [345](#)
- printi
 - sbi_console.c, [346](#)
- prints
 - sbi_console.c, [347](#)
- probe
 - sbi_ecall_extension, [52](#)
- process
 - sbi_ipi_event_ops, [54](#)
- qlock
 - sbi_fifo, [53](#)
- queue
 - sbi_fifo, [54](#)
- README.md, [661](#)
- REG_MASK
 - riscv_encoding.h, [156](#)
- REG_OFFSET
 - riscv_encoding.h, [156](#)
- REG_PTR
 - riscv_encoding.h, [156](#)
- REG_L
 - riscv_asm.h, [81](#)
- REG_S
 - riscv_asm.h, [81](#)
- REGBYTES
 - riscv_encoding.h, [156](#)
- RISCV_ACQUIRE_BARRIER
 - riscv_barrier.h, [92](#)
- RISCV_FENCE
 - riscv_barrier.h, [92](#)
- RISCV_PGLEVEL_BITS
 - riscv_encoding.h, [156](#)
- RISCV_PGSHIFT
 - riscv_encoding.h, [156](#)
- RISCV_PGSIZE
 - riscv_encoding.h, [156](#)
- RISCV_RELEASE_BARRIER
 - riscv_barrier.h, [92](#)
- ROUNDDOWN
 - sbi_types.h, [305](#)
- ROUNDUP
 - sbi_types.h, [305](#)
- RV_X
 - riscv_encoding.h, [157](#)
- RVC_LD_IMM
 - riscv_encoding.h, [157](#)
- RVC_LDSP_IMM

- riscv_encoding.h, 157
- RVC_LW_IMM
 - riscv_encoding.h, 157
- RVC_LWSP_IMM
 - riscv_encoding.h, 157
- RVC_RS1S
 - riscv_encoding.h, 158
- RVC_RS2
 - riscv_encoding.h, 158
- RVC_RS2S
 - riscv_encoding.h, 158
- RVC_SDSP_IMM
 - riscv_encoding.h, 158
- RVC_SWSP_IMM
 - riscv_encoding.h, 158
- ra
 - sbi_trap_regs, 68
- readb
 - riscv_io.h, 168
- readb_relaxed
 - riscv_io.h, 168
- readl
 - riscv_io.h, 168
- readl_relaxed
 - riscv_io.h, 168
- readq
 - riscv_io.h, 168
- readq_relaxed
 - riscv_io.h, 168
- readw
 - riscv_io.h, 168
- readw_relaxed
 - riscv_io.h, 169
- riscv_asm.c
 - csr_read_num, 326
 - csr_write_num, 327
 - ctz, 327
 - misa_extension_imp, 327
 - misa_xlen, 328
 - pmp_get, 328
 - pmp_set, 329
- riscv_asm.h
 - __ASM_STR, 78
 - csr_clear, 78
 - csr_read, 78
 - csr_read_clear, 78
 - csr_read_num, 82
 - csr_read_set, 79
 - csr_set, 79
 - csr_swap, 79
 - csr_write, 80
 - csr_write_num, 82
 - LGREG, 80
 - misa_extension, 80
 - misa_extension_imp, 82
 - misa_string, 83
 - misa_xlen, 83
 - PAGE_MASK, 80
 - PAGE_SHIFT, 81
 - PAGE_SIZE, 81
 - pmp_get, 84
 - pmp_set, 85
 - REG_L, 81
 - REG_S, 81
 - SZREG, 81
 - wfi, 81
- riscv_atomic.c
 - __NOP, 332
 - __NOT, 332
 - __atomic_op_bit, 331
 - __atomic_op_bit_ord, 331
 - __axchg, 331
 - __cmpxchg, 332
 - __xchg, 332
 - arch_atomic_cmpxchg, 334
 - arch_atomic_xchg, 334
 - atomic_add_return, 334
 - atomic_clear_bit, 334
 - atomic_raw_clear_bit, 335
 - atomic_raw_set_bit, 335
 - atomic_raw_xchg_uint, 336
 - atomic_raw_xchg_ulong, 336
 - atomic_read, 336
 - atomic_set_bit, 336
 - atomic_sub_return, 337
 - atomic_write, 337
 - axchg, 333
 - cmpxchg, 333
 - xchg, 333
- riscv_atomic.h
 - ATOMIC_INITIALIZER, 86
 - ATOMIC_INIT, 86
 - arch_atomic_cmpxchg, 87
 - arch_atomic_xchg, 87
 - atomic_add_return, 87
 - atomic_clear_bit, 87
 - atomic_raw_clear_bit, 88
 - atomic_raw_set_bit, 88
 - atomic_raw_xchg_uint, 89
 - atomic_raw_xchg_ulong, 89
 - atomic_read, 89
 - atomic_set_bit, 89
 - atomic_sub_return, 90
 - atomic_write, 90
- riscv_barrier.h
 - __smp_load_acquire, 91
 - __smp_store_release, 91
 - cpu_relax, 91
 - mb, 92
 - RISCV_ACQUIRE_BARRIER, 92
 - RISCV_FENCE, 92
 - RISCV_RELEASE_BARRIER, 92
 - rmb, 92
 - smp_mb, 92
 - smp_rmb, 92
 - smp_wmb, 93

- wmb, [93](#)
- riscv_encoding.h
 - CAUSE_BREAKPOINT, [102](#)
 - CAUSE_FETCH_ACCESS, [102](#)
 - CAUSE_FETCH_GUEST_PAGE_FAULT, [102](#)
 - CAUSE_FETCH_PAGE_FAULT, [103](#)
 - CAUSE_HYPERVISOR_ECALL, [103](#)
 - CAUSE_ILLEGAL_INSTRUCTION, [103](#)
 - CAUSE_LOAD_ACCESS, [103](#)
 - CAUSE_LOAD_GUEST_PAGE_FAULT, [103](#)
 - CAUSE_LOAD_PAGE_FAULT, [103](#)
 - CAUSE_MACHINE_ECALL, [103](#)
 - CAUSE_MISALIGNED_FETCH, [103](#)
 - CAUSE_MISALIGNED_LOAD, [104](#)
 - CAUSE_MISALIGNED_STORE, [104](#)
 - CAUSE_STORE_ACCESS, [104](#)
 - CAUSE_STORE_GUEST_PAGE_FAULT, [104](#)
 - CAUSE_STORE_PAGE_FAULT, [104](#)
 - CAUSE_SUPERVISOR_ECALL, [104](#)
 - CAUSE_USER_ECALL, [104](#)
 - CSR_CYCLEH, [105](#)
 - CSR_CYCLE, [104](#)
 - CSR_DCSR, [105](#)
 - CSR_DPC, [105](#)
 - CSR_DSCRATCH, [105](#)
 - CSR_FCSR, [105](#)
 - CSR_FFLAGS, [105](#)
 - CSR_FRM, [105](#)
 - CSR_HCOUNTERNEN, [105](#)
 - CSR_HEDELEG, [106](#)
 - CSR_HGATP, [106](#)
 - CSR_HGEIE, [106](#)
 - CSR_HGEIP, [106](#)
 - CSR_HIDELEG, [106](#)
 - CSR_HIE, [106](#)
 - CSR_HIP, [106](#)
 - CSR_HPMCounter10, [106](#)
 - CSR_HPMCounter10H, [107](#)
 - CSR_HPMCounter11, [107](#)
 - CSR_HPMCounter11H, [107](#)
 - CSR_HPMCounter12, [107](#)
 - CSR_HPMCounter12H, [107](#)
 - CSR_HPMCounter13, [107](#)
 - CSR_HPMCounter13H, [107](#)
 - CSR_HPMCounter14, [107](#)
 - CSR_HPMCounter14H, [108](#)
 - CSR_HPMCounter15, [108](#)
 - CSR_HPMCounter15H, [108](#)
 - CSR_HPMCounter16, [108](#)
 - CSR_HPMCounter16H, [108](#)
 - CSR_HPMCounter17, [108](#)
 - CSR_HPMCounter17H, [108](#)
 - CSR_HPMCounter18, [108](#)
 - CSR_HPMCounter18H, [109](#)
 - CSR_HPMCounter19, [109](#)
 - CSR_HPMCounter19H, [109](#)
 - CSR_HPMCounter20, [109](#)
 - CSR_HPMCounter20H, [109](#)
 - CSR_HPMCounter21, [109](#)
 - CSR_HPMCounter21H, [109](#)
 - CSR_HPMCounter22, [109](#)
 - CSR_HPMCounter22H, [110](#)
 - CSR_HPMCounter23, [110](#)
 - CSR_HPMCounter23H, [110](#)
 - CSR_HPMCounter24, [110](#)
 - CSR_HPMCounter24H, [110](#)
 - CSR_HPMCounter25, [110](#)
 - CSR_HPMCounter25H, [110](#)
 - CSR_HPMCounter26, [110](#)
 - CSR_HPMCounter26H, [111](#)
 - CSR_HPMCounter27, [111](#)
 - CSR_HPMCounter27H, [111](#)
 - CSR_HPMCounter28, [111](#)
 - CSR_HPMCounter28H, [111](#)
 - CSR_HPMCounter29, [111](#)
 - CSR_HPMCounter29H, [111](#)
 - CSR_HPMCounter3, [111](#)
 - CSR_HPMCounter30, [112](#)
 - CSR_HPMCounter30H, [112](#)
 - CSR_HPMCounter31, [112](#)
 - CSR_HPMCounter31H, [112](#)
 - CSR_HPMCounter3H, [112](#)
 - CSR_HPMCounter4, [112](#)
 - CSR_HPMCounter4H, [112](#)
 - CSR_HPMCounter5, [112](#)
 - CSR_HPMCounter5H, [113](#)
 - CSR_HPMCounter6, [113](#)
 - CSR_HPMCounter6H, [113](#)
 - CSR_HPMCounter7, [113](#)
 - CSR_HPMCounter7H, [113](#)
 - CSR_HPMCounter8, [113](#)
 - CSR_HPMCounter8H, [113](#)
 - CSR_HPMCounter9, [113](#)
 - CSR_HPMCounter9H, [114](#)
 - CSR_HSTATUS, [114](#)
 - CSR_HTIMEDELTAH, [114](#)
 - CSR_HTIMEDELTA, [114](#)
 - CSR_HTINST, [114](#)
 - CSR_HTVAl, [114](#)
 - CSR_INSTRETH, [114](#)
 - CSR_INSTRET, [114](#)
 - CSR_MARCHID, [115](#)
 - CSR_MCAUSE, [115](#)
 - CSR_MCounterEN, [115](#)
 - CSR_MCYCLEH, [115](#)
 - CSR_MCYCLE, [115](#)
 - CSR_MEDELEG, [115](#)
 - CSR_MEPC, [115](#)
 - CSR_MHARTID, [115](#)
 - CSR_MHPMCounter10, [116](#)
 - CSR_MHPMCounter10H, [116](#)
 - CSR_MHPMCounter11, [116](#)
 - CSR_MHPMCounter11H, [116](#)
 - CSR_MHPMCounter12, [116](#)
 - CSR_MHPMCounter12H, [116](#)
 - CSR_MHPMCounter13, [116](#)

CSR_MHPMCOUNTER13H, [116](#)
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, [118](#)
CSR_MHPMCOUNTER18H, [118](#)
CSR_MHPMCOUNTER19, [118](#)
CSR_MHPMCOUNTER19H, [118](#)
CSR_MHPMCOUNTER20, [118](#)
CSR_MHPMCOUNTER20H, [118](#)
CSR_MHPMCOUNTER21, [118](#)
CSR_MHPMCOUNTER21H, [118](#)
CSR_MHPMCOUNTER22, [119](#)
CSR_MHPMCOUNTER22H, [119](#)
CSR_MHPMCOUNTER23, [119](#)
CSR_MHPMCOUNTER23H, [119](#)
CSR_MHPMCOUNTER24, [119](#)
CSR_MHPMCOUNTER24H, [119](#)
CSR_MHPMCOUNTER25, [119](#)
CSR_MHPMCOUNTER25H, [119](#)
CSR_MHPMCOUNTER26, [120](#)
CSR_MHPMCOUNTER26H, [120](#)
CSR_MHPMCOUNTER27, [120](#)
CSR_MHPMCOUNTER27H, [120](#)
CSR_MHPMCOUNTER28, [120](#)
CSR_MHPMCOUNTER28H, [120](#)
CSR_MHPMCOUNTER29, [120](#)
CSR_MHPMCOUNTER29H, [120](#)
CSR_MHPMCOUNTER3, [121](#)
CSR_MHPMCOUNTER30, [121](#)
CSR_MHPMCOUNTER30H, [121](#)
CSR_MHPMCOUNTER31, [121](#)
CSR_MHPMCOUNTER31H, [121](#)
CSR_MHPMCOUNTER3H, [121](#)
CSR_MHPMCOUNTER4, [121](#)
CSR_MHPMCOUNTER4H, [121](#)
CSR_MHPMCOUNTER5, [122](#)
CSR_MHPMCOUNTER5H, [122](#)
CSR_MHPMCOUNTER6, [122](#)
CSR_MHPMCOUNTER6H, [122](#)
CSR_MHPMCOUNTER7, [122](#)
CSR_MHPMCOUNTER7H, [122](#)
CSR_MHPMCOUNTER8, [122](#)
CSR_MHPMCOUNTER8H, [122](#)
CSR_MHPMCOUNTER9, [123](#)
CSR_MHPMCOUNTER9H, [123](#)
CSR_MHPMEVENT10, [123](#)
CSR_MHPMEVENT11, [123](#)
CSR_MHPMEVENT12, [123](#)
CSR_MHPMEVENT13, [123](#)
CSR_MHPMEVENT14, [123](#)
CSR_MHPMEVENT15, [123](#)
CSR_MHPMEVENT16, [124](#)
CSR_MHPMEVENT17, [124](#)
CSR_MHPMEVENT18, [124](#)
CSR_MHPMEVENT19, [124](#)
CSR_MHPMEVENT20, [124](#)
CSR_MHPMEVENT21, [124](#)
CSR_MHPMEVENT22, [124](#)
CSR_MHPMEVENT23, [124](#)
CSR_MHPMEVENT24, [125](#)
CSR_MHPMEVENT25, [125](#)
CSR_MHPMEVENT26, [125](#)
CSR_MHPMEVENT27, [125](#)
CSR_MHPMEVENT28, [125](#)
CSR_MHPMEVENT29, [125](#)
CSR_MHPMEVENT3, [125](#)
CSR_MHPMEVENT30, [125](#)
CSR_MHPMEVENT31, [126](#)
CSR_MHPMEVENT4, [126](#)
CSR_MHPMEVENT5, [126](#)
CSR_MHPMEVENT6, [126](#)
CSR_MHPMEVENT7, [126](#)
CSR_MHPMEVENT8, [126](#)
CSR_MHPMEVENT9, [126](#)
CSR_MIDELEG, [126](#)
CSR_MIMPID, [127](#)
CSR_MINSTRETH, [127](#)
CSR_MINSTRET, [127](#)
CSR_MISA, [127](#)
CSR_MIE, [127](#)
CSR_MIP, [127](#)
CSR_MSCRATCH, [127](#)
CSR_MSTATUSH, [128](#)
CSR_MSTATUS, [127](#)
CSR_MTINST, [128](#)
CSR_MTVAL2, [128](#)
CSR_MTVAL, [128](#)
CSR_MTVEC, [128](#)
CSR_MVENDORID, [128](#)
CSR_PMPADDR0, [128](#)
CSR_PMPADDR1, [128](#)
CSR_PMPADDR10, [129](#)
CSR_PMPADDR11, [129](#)
CSR_PMPADDR12, [129](#)
CSR_PMPADDR13, [129](#)
CSR_PMPADDR14, [129](#)
CSR_PMPADDR15, [129](#)
CSR_PMPADDR2, [129](#)
CSR_PMPADDR3, [129](#)
CSR_PMPADDR4, [130](#)
CSR_PMPADDR5, [130](#)
CSR_PMPADDR6, [130](#)
CSR_PMPADDR7, [130](#)
CSR_PMPADDR8, [130](#)
CSR_PMPADDR9, [130](#)
CSR_PMPCFG0, [130](#)
CSR_PMPCFG1, [130](#)
CSR_PMPCFG2, [131](#)
CSR_PMPCFG3, [131](#)
CSR_SATP, [131](#)

CSR_SCAUSE, 131
CSR_SCOUNTEREN, 131
CSR_SEPC, 131
CSR_SIE, 131
CSR_SIP, 131
CSR_SSCRATCH, 132
CSR_SSTATUS, 132
CSR_STVAL, 132
CSR_STVEC, 132
CSR_TDATA1, 132
CSR_TDATA2, 132
CSR_TDATA3, 132
CSR_TIMEH, 133
CSR_TIME, 132
CSR_TSELECT, 133
CSR_UCAUSE, 133
CSR_UEPC, 133
CSR_UIE, 133
CSR_UIP, 133
CSR_USCRATCH, 133
CSR_USTATUS, 133
CSR_UTVAL, 134
CSR_UTVEC, 134
CSR_VSATP, 134
CSR_VSCAUSE, 134
CSR_VSEPC, 134
CSR_VSIE, 134
CSR_VSIP, 134
CSR_VSSCRATCH, 134
CSR_VSSTATUS, 135
CSR_VSTVAL, 135
CSR_VSTVEC, 135
GET_RS1, 135
GET_RS1S, 135
GET_RS2, 135
GET_RS2C, 136
GET_RS2S, 136
GET_RM, 135
GET_SP, 136
HSTATUS_SP2P, 136
HSTATUS_SP2V, 136
HSTATUS_SPRV, 136
HSTATUS_SPV, 136
HSTATUS_VTSR, 137
HSTATUS_VTVM, 137
IMM_I, 137
IMM_S, 137
INSN_16BIT_MASK, 137
INSN_32BIT_MASK, 137
INSN_IS_16BIT, 137
INSN_IS_32BIT, 138
INSN_LEN, 138
INSN_MASK_C_FLDSP, 138
INSN_MASK_C_FLWSP, 138
INSN_MASK_C_FLD, 138
INSN_MASK_C_FLW, 138
INSN_MASK_C_FSDSP, 139
INSN_MASK_C_FSWSP, 139
INSN_MASK_C_FSD, 139
INSN_MASK_C_FSW, 139
INSN_MASK_C_LDSP, 139
INSN_MASK_C_LWSP, 139
INSN_MASK_C_LD, 139
INSN_MASK_C_LW, 139
INSN_MASK_C_SDSP, 140
INSN_MASK_C_SWSP, 140
INSN_MASK_C_SD, 140
INSN_MASK_C_SW, 140
INSN_MASK_FLD, 140
INSN_MASK_FLQ, 140
INSN_MASK_FLW, 140
INSN_MASK_FSD, 140
INSN_MASK_FSQ, 141
INSN_MASK_FSW, 141
INSN_MASK_LBU, 141
INSN_MASK_LHU, 141
INSN_MASK_LWU, 142
INSN_MASK_LB, 141
INSN_MASK_LD, 141
INSN_MASK_LH, 141
INSN_MASK_LW, 141
INSN_MASK_SB, 142
INSN_MASK_SD, 142
INSN_MASK_SH, 142
INSN_MASK_SW, 142
INSN_MASK_WFI, 142
INSN_MATCH_C_FLDSP, 142
INSN_MATCH_C_FLWSP, 143
INSN_MATCH_C_FLD, 142
INSN_MATCH_C_FLW, 143
INSN_MATCH_C_FSDSP, 143
INSN_MATCH_C_FSWSP, 143
INSN_MATCH_C_FSD, 143
INSN_MATCH_C_FSW, 143
INSN_MATCH_C_LDSP, 143
INSN_MATCH_C_LWSP, 144
INSN_MATCH_C_LD, 143
INSN_MATCH_C_LW, 144
INSN_MATCH_C_SDSP, 144
INSN_MATCH_C_SWSP, 144
INSN_MATCH_C_SD, 144
INSN_MATCH_C_SW, 144
INSN_MATCH_FLD, 144
INSN_MATCH_FLQ, 144
INSN_MATCH_FLW, 145
INSN_MATCH_FSD, 145
INSN_MATCH_FSQ, 145
INSN_MATCH_FSW, 145
INSN_MATCH_LBU, 145
INSN_MATCH_LHU, 146
INSN_MATCH_LWU, 146
INSN_MATCH_LB, 145
INSN_MATCH_LD, 145
INSN_MATCH_LH, 145
INSN_MATCH_LW, 146
INSN_MATCH_SB, 146

INSN_MATCH_SD, [146](#)
 INSN_MATCH_SH, [146](#)
 INSN_MATCH_SW, [146](#)
 INSN_MATCH_WFI, [146](#)
 IRQ_M_EXT, [147](#)
 IRQ_M_SOFT, [147](#)
 IRQ_M_TIMER, [147](#)
 IRQ_S_EXT, [147](#)
 IRQ_S_GEXT, [147](#)
 IRQ_S_SOFT, [147](#)
 IRQ_S_TIMER, [147](#)
 IRQ_VS_EXT, [147](#)
 IRQ_VS_SOFT, [148](#)
 IRQ_VS_TIMER, [148](#)
 LOG_REGBYTES, [148](#)
 MASK_FUNCT3, [148](#)
 MIP_MEIP, [148](#)
 MIP_MSIP, [148](#)
 MIP_MTIP, [148](#)
 MIP_SEIP, [148](#)
 MIP_SGEIP, [149](#)
 MIP_SSIP, [149](#)
 MIP_STIP, [149](#)
 MIP_VSEIP, [149](#)
 MIP_VSSIP, [149](#)
 MIP_VSTIP, [149](#)
 MSTATUS32_SD, [149](#)
 MSTATUS64_SD, [150](#)
 MSTATUS_FS, [150](#)
 MSTATUS_MIE, [150](#)
 MSTATUS_MPIE, [150](#)
 MSTATUS_MPP_SHIFT, [150](#)
 MSTATUS_MPRV, [150](#)
 MSTATUS_MPP, [150](#)
 MSTATUS_MXR, [150](#)
 MSTATUS_SIE, [151](#)
 MSTATUS_SPIE_SHIFT, [151](#)
 MSTATUS_SPIE, [151](#)
 MSTATUS_SPP_SHIFT, [151](#)
 MSTATUS_SPP, [151](#)
 MSTATUS_SUM, [151](#)
 MSTATUS_SD, [151](#)
 MSTATUS_TSR, [151](#)
 MSTATUS_TVM, [152](#)
 MSTATUS_TW, [152](#)
 MSTATUS_UBE, [152](#)
 MSTATUS_XS, [152](#)
 MSTATUSH_MBE, [152](#)
 MSTATUSH_MPV, [152](#)
 MSTATUSH_SBE, [152](#)
 PMP_A_NA4, [153](#)
 PMP_A_NAPOT, [153](#)
 PMP_A_TOR, [153](#)
 PMP_COUNT, [153](#)
 PMP_SHIFT, [153](#)
 PMP_A, [152](#)
 PMP_L, [153](#)
 PMP_R, [153](#)
 PMP_W, [153](#)
 PMP_X, [154](#)
 PRV_M, [154](#)
 PRV_S, [154](#)
 PRV_U, [154](#)
 PTE_PPN_SHIFT, [154](#)
 PTE_SOFT, [155](#)
 PTE_TABLE, [155](#)
 PTE_A, [154](#)
 PTE_D, [154](#)
 PTE_G, [154](#)
 PTE_R, [155](#)
 PTE_U, [155](#)
 PTE_V, [155](#)
 PTE_W, [155](#)
 PTE_X, [155](#)
 REG_MASK, [156](#)
 REG_OFFSET, [156](#)
 REG_PTR, [156](#)
 REGBYTES, [156](#)
 RISCV_PGLEVEL_BITS, [156](#)
 RISCV_PGSHIFT, [156](#)
 RISCV_PGSIZE, [156](#)
 RV_X, [157](#)
 RVC_LD_IMM, [157](#)
 RVC_LDSP_IMM, [157](#)
 RVC_LW_IMM, [157](#)
 RVC_LWSP_IMM, [157](#)
 RVC_RS1S, [158](#)
 RVC_RS2, [158](#)
 RVC_RS2S, [158](#)
 RVC_SDSP_IMM, [158](#)
 RVC_SWSP_IMM, [158](#)
 SATP32_ASID, [159](#)
 SATP32_MODE, [159](#)
 SATP32_PPN, [159](#)
 SATP64_ASID, [159](#)
 SATP64_MODE, [159](#)
 SATP64_PPN, [159](#)
 SATP_MODE_OFF, [160](#)
 SATP_MODE_SV32, [160](#)
 SATP_MODE_SV39, [160](#)
 SATP_MODE_SV48, [160](#)
 SATP_MODE_SV57, [160](#)
 SATP_MODE_SV64, [160](#)
 SATP_MODE, [159](#)
 SET_RD, [160](#)
 SH_RS1, [161](#)
 SH_RS2, [161](#)
 SH_RS2C, [161](#)
 SH_RD, [161](#)
 SHIFT_RIGHT, [161](#)
 SIP_SSIP, [161](#)
 SIP_STIP, [161](#)
 SSTATUS32_SD, [162](#)
 SSTATUS64_SD, [162](#)
 SSTATUS64_UXL, [162](#)
 SSTATUS_FS, [162](#)

- SSTATUS_MXR, 162
- SSTATUS_SIE, 162
- SSTATUS_SPIE_SHIFT, 163
- SSTATUS_SPIE, 162
- SSTATUS_SPP_SHIFT, 163
- SSTATUS_SPP, 163
- SSTATUS_SUM, 163
- SSTATUS_SD, 162
- SSTATUS_XS, 163
- riscv_fp.h
 - GET_PRECISION, 165
 - GET_RM, 165
 - PRECISION_D, 165
 - PRECISION_S, 165
- riscv_io.h
 - __io_ar, 167
 - __io_aw, 167
 - __io_br, 167
 - __io_bw, 167
 - __io_rar, 167
 - __io_raw, 167
 - __io_rbr, 167
 - __io_rbw, 167
 - __raw_readb, 170
 - __raw_readl, 170
 - __raw_readq, 170
 - __raw_readw, 171
 - __raw_writeb, 171
 - __raw_writel, 171
 - __raw_writeq, 171
 - __raw_writew, 171
 - readb, 168
 - readb_relaxed, 168
 - readl, 168
 - readl_relaxed, 168
 - readq, 168
 - readq_relaxed, 168
 - readw, 168
 - readw_relaxed, 169
 - writeb, 169
 - writeb_relaxed, 169
 - writel, 169
 - writel_relaxed, 169
 - writeq, 169
 - writeq_relaxed, 170
 - writew, 170
 - writew_relaxed, 170
- riscv_locks.c
 - spin_lock, 338
 - spin_lock_check, 339
 - spin_trylock, 340
 - spin_unlock, 341
- riscv_locks.h
 - __RISCV_SPIN_UNLOCKED, 172
 - SPIN_LOCK_INITIALIZER, 172
 - SPIN_LOCK_INIT, 172
 - spin_lock, 173
 - spin_lock_check, 174
 - spin_trylock, 175
 - spin_unlock, 176
- rmb
 - riscv_barrier.h, 92
- s0
 - sbi_trap_regs, 68
- s1
 - sbi_trap_regs, 68
- s10
 - sbi_trap_regs, 69
- s11
 - sbi_trap_regs, 69
- s16
 - sbi_types.h, 306
- s2
 - sbi_trap_regs, 69
- s3
 - sbi_trap_regs, 69
- s32
 - sbi_types.h, 306
- s4
 - sbi_trap_regs, 69
- s5
 - sbi_trap_regs, 69
- s6
 - sbi_trap_regs, 69
- s7
 - sbi_trap_regs, 69
- s8
 - sbi_trap_regs, 70
 - sbi_types.h, 307
- s9
 - sbi_trap_regs, 70
- SATP32_ASID
 - riscv_encoding.h, 159
- SATP32_MODE
 - riscv_encoding.h, 159
- SATP32_PPN
 - riscv_encoding.h, 159
- SATP64_ASID
 - riscv_encoding.h, 159
- SATP64_MODE
 - riscv_encoding.h, 159
- SATP64_PPN
 - riscv_encoding.h, 159
- SATP_MODE_OFF
 - riscv_encoding.h, 160
- SATP_MODE_SV32
 - riscv_encoding.h, 160
- SATP_MODE_SV39
 - riscv_encoding.h, 160
- SATP_MODE_SV48
 - riscv_encoding.h, 160
- SATP_MODE_SV57
 - riscv_encoding.h, 160
- SATP_MODE_SV64
 - riscv_encoding.h, 160
- SATP_MODE

- riscv_encoding.h, [159](#)
- SBI_DENIED
 - sbi_error.h, [202](#)
- SBI_ECALL_VERSION_MAJOR
 - sbi_ecall.h, [190](#)
- SBI_ECALL_VERSION_MINOR
 - sbi_ecall.h, [190](#)
- SBI_EFAIL
 - sbi_error.h, [202](#)
- SBI_EILL
 - sbi_error.h, [203](#)
- SBI_EINVAL
 - sbi_error.h, [203](#)
- SBI_EIO
 - sbi_error.h, [203](#)
- SBI_ENODEV
 - sbi_error.h, [203](#)
- SBI_ENOENT
 - sbi_error.h, [203](#)
- SBI_ENOMEM
 - sbi_error.h, [203](#)
- SBI_ENOSPC
 - sbi_error.h, [203](#)
- SBI_ENOSYS
 - sbi_error.h, [203](#)
- SBI_ENOTSUPP
 - sbi_error.h, [204](#)
- SBI_ETIMEDOUT
 - sbi_error.h, [204](#)
- SBI_ETRAP
 - sbi_error.h, [204](#)
- SBI_EUNKNOWN
 - sbi_error.h, [204](#)
- SBI_EXT_0_1_CLEAR_IPI
 - sbi_ecall_interface.h, [195](#)
- SBI_EXT_0_1_CONSOLE_GETCHAR
 - sbi_ecall_interface.h, [195](#)
- SBI_EXT_0_1_CONSOLE_PUTCHAR
 - sbi_ecall_interface.h, [195](#)
- SBI_EXT_0_1_REMOTE_FENCE_I
 - sbi_ecall_interface.h, [196](#)
- SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID
 - sbi_ecall_interface.h, [196](#)
- SBI_EXT_0_1_REMOTE_SFENCE_VMA
 - sbi_ecall_interface.h, [196](#)
- SBI_EXT_0_1_SEND_IPI
 - sbi_ecall_interface.h, [196](#)
- SBI_EXT_0_1_SET_TIMER
 - sbi_ecall_interface.h, [196](#)
- SBI_EXT_0_1_SHUTDOWN
 - sbi_ecall_interface.h, [196](#)
- SBI_EXT_BASE_GET_IMP_ID
 - sbi_ecall_interface.h, [196](#)
- SBI_EXT_BASE_GET_IMP_VERSION
 - sbi_ecall_interface.h, [197](#)
- SBI_EXT_BASE_GET_MARCHID
 - sbi_ecall_interface.h, [197](#)
- SBI_EXT_BASE_GET_MIMPID
 - sbi_ecall_interface.h, [197](#)
- SBI_EXT_BASE_GET_MVENDORID
 - sbi_ecall_interface.h, [197](#)
- SBI_EXT_BASE_GET_SPEC_VERSION
 - sbi_ecall_interface.h, [197](#)
- SBI_EXT_BASE_PROBE_EXT
 - sbi_ecall_interface.h, [197](#)
- SBI_EXT_BASE
 - sbi_ecall_interface.h, [196](#)
- SBI_EXT_IPI_SEND_IPI
 - sbi_ecall_interface.h, [197](#)
- SBI_EXT_IPI
 - sbi_ecall_interface.h, [197](#)
- SBI_EXT_RFENCE_REMOTE_FENCE_I
 - sbi_ecall_interface.h, [198](#)
- SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_V↔MID
 - sbi_ecall_interface.h, [198](#)
- SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA
 - sbi_ecall_interface.h, [198](#)
- SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID
 - sbi_ecall_interface.h, [198](#)
- SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA
 - sbi_ecall_interface.h, [198](#)
- SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID
 - sbi_ecall_interface.h, [198](#)
- SBI_EXT_RFENCE_REMOTE_SFENCE_VMA
 - sbi_ecall_interface.h, [198](#)
- SBI_EXT_RFENCE
 - sbi_ecall_interface.h, [198](#)
- SBI_EXT_TIME_SET_TIMER
 - sbi_ecall_interface.h, [199](#)
- SBI_EXT_TIME
 - sbi_ecall_interface.h, [199](#)
- SBI_EXT_VENDOR_END
 - sbi_ecall_interface.h, [199](#)
- SBI_EXT_VENDOR_START
 - sbi_ecall_interface.h, [199](#)
- SBI_INIT_LIST_HEAD
 - sbi_list.h, [237](#)
- SBI_INVALID_ADDR
 - sbi_error.h, [204](#)
- SBI_IPI_EVENT_MAX
 - sbi_ipi.h, [230](#)
- SBI_LIST_HEAD_INIT
 - sbi_list.h, [238](#)
- SBI_LIST_HEAD
 - sbi_ecall.c, [357](#)
 - sbi_list.h, [238](#)
- SBI_LIST_POISON_NEXT
 - sbi_list.h, [239](#)
- SBI_LIST_POISON_PREV
 - sbi_list.h, [239](#)
- SBI_OPENSBI_IMPID
 - sbi_ecall.h, [190](#)
- SBI_OK
 - sbi_error.h, [204](#)
- SBI_PLATFORM_DEFAULT_FEATURES

- sbi_platform.h, [248](#)
- SBI_PLATFORM_DISABLED_HART_OFFSET
 - sbi_platform.h, [248](#)
- SBI_PLATFORM_FEATURES_OFFSET
 - sbi_platform.h, [248](#)
- SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET
 - sbi_platform.h, [248](#)
- SBI_PLATFORM_HART_COUNT_OFFSET
 - sbi_platform.h, [248](#)
- SBI_PLATFORM_HART_STACK_SIZE_OFFSET
 - sbi_platform.h, [248](#)
- SBI_PLATFORM_NAME_OFFSET
 - sbi_platform.h, [249](#)
- SBI_PLATFORM_OPENSBI_VERSION_OFFSET
 - sbi_platform.h, [250](#)
- SBI_PLATFORM_OPS_OFFSET
 - sbi_platform.h, [250](#)
- SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT
 - AULT
 - sbi_platform.h, [250](#)
- SBI_PLATFORM_VERSION_OFFSET
 - sbi_platform.h, [251](#)
- SBI_PLATFORM_VERSION
 - sbi_platform.h, [250](#)
- SBI_SCRATCH_EXTRA_SPACE_OFFSET
 - sbi_scratch.h, [271](#)
- SBI_SCRATCH_FW_SIZE_OFFSET
 - sbi_scratch.h, [272](#)
- SBI_SCRATCH_FW_START_OFFSET
 - sbi_scratch.h, [272](#)
- SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET
 - sbi_scratch.h, [272](#)
- SBI_SCRATCH_NEXT_ADDR_OFFSET
 - sbi_scratch.h, [272](#)
- SBI_SCRATCH_NEXT_ARG1_OFFSET
 - sbi_scratch.h, [272](#)
- SBI_SCRATCH_NEXT_MODE_OFFSET
 - sbi_scratch.h, [272](#)
- SBI_SCRATCH_OPTIONS_OFFSET
 - sbi_scratch.h, [273](#)
- SBI_SCRATCH_PLATFORM_ADDR_OFFSET
 - sbi_scratch.h, [273](#)
- SBI_SCRATCH_SIZE
 - sbi_scratch.h, [273](#)
- SBI_SCRATCH_TMP0_OFFSET
 - sbi_scratch.h, [273](#)
- SBI_SCRATCH_WARMBOOT_ADDR_OFFSET
 - sbi_scratch.h, [274](#)
- SBI_SPEC_VERSION_MAJOR_MASK
 - sbi_ecall_interface.h, [199](#)
- SBI_SPEC_VERSION_MAJOR_OFFSET
 - sbi_ecall_interface.h, [199](#)
- SBI_SPEC_VERSION_MINOR_MASK
 - sbi_ecall_interface.h, [199](#)
- SBI_TLB_FIFO_NUM_ENTRIES
 - sbi_tlb.h, [291](#)
- SBI_TLB_FLUSH_ALL
 - sbi_tlb.h, [291](#)
- SBI_TLB_INFO_SIZE
 - sbi_tlb.h, [291](#)
- SBI_TRAP_REGS_OFFSET
 - sbi_trap.h, [296](#)
- SBI_TRAP_REGS_SIZE
 - sbi_trap.h, [298](#)
- SBI_TRAP_REGS_a0
 - sbi_trap.h, [295](#)
- SBI_TRAP_REGS_a1
 - sbi_trap.h, [295](#)
- SBI_TRAP_REGS_a2
 - sbi_trap.h, [295](#)
- SBI_TRAP_REGS_a3
 - sbi_trap.h, [295](#)
- SBI_TRAP_REGS_a4
 - sbi_trap.h, [295](#)
- SBI_TRAP_REGS_a5
 - sbi_trap.h, [295](#)
- SBI_TRAP_REGS_a6
 - sbi_trap.h, [296](#)
- SBI_TRAP_REGS_a7
 - sbi_trap.h, [296](#)
- SBI_TRAP_REGS_gp
 - sbi_trap.h, [296](#)
- SBI_TRAP_REGS_last
 - sbi_trap.h, [296](#)
- SBI_TRAP_REGS_mepc
 - sbi_trap.h, [296](#)
- SBI_TRAP_REGS_mstatus
 - sbi_trap.h, [296](#)
- SBI_TRAP_REGS_mstatusH
 - sbi_trap.h, [296](#)
- SBI_TRAP_REGS_ra
 - sbi_trap.h, [297](#)
- SBI_TRAP_REGS_s0
 - sbi_trap.h, [297](#)
- SBI_TRAP_REGS_s1
 - sbi_trap.h, [297](#)
- SBI_TRAP_REGS_s10
 - sbi_trap.h, [297](#)
- SBI_TRAP_REGS_s11
 - sbi_trap.h, [297](#)
- SBI_TRAP_REGS_s2
 - sbi_trap.h, [297](#)
- SBI_TRAP_REGS_s3
 - sbi_trap.h, [297](#)
- SBI_TRAP_REGS_s4
 - sbi_trap.h, [298](#)
- SBI_TRAP_REGS_s5
 - sbi_trap.h, [298](#)
- SBI_TRAP_REGS_s6
 - sbi_trap.h, [298](#)
- SBI_TRAP_REGS_s7
 - sbi_trap.h, [298](#)
- SBI_TRAP_REGS_s8
 - sbi_trap.h, [298](#)
- SBI_TRAP_REGS_s9
 - sbi_trap.h, [298](#)

SBI_TRAP_REGS_sp
 sbi_trap.h, 298
 SBI_TRAP_REGS_t0
 sbi_trap.h, 299
 SBI_TRAP_REGS_t1
 sbi_trap.h, 299
 SBI_TRAP_REGS_t2
 sbi_trap.h, 299
 SBI_TRAP_REGS_t3
 sbi_trap.h, 299
 SBI_TRAP_REGS_t4
 sbi_trap.h, 299
 SBI_TRAP_REGS_t5
 sbi_trap.h, 299
 SBI_TRAP_REGS_t6
 sbi_trap.h, 299
 SBI_TRAP_REGS_tp
 sbi_trap.h, 299
 SBI_TRAP_REGS_zero
 sbi_trap.h, 300
 SET_RD
 riscv_encoding.h, 160
 SH_RS1
 riscv_encoding.h, 161
 SH_RS2
 riscv_encoding.h, 161
 SH_RS2C
 riscv_encoding.h, 161
 SH_RD
 riscv_encoding.h, 161
 SHIFT_RIGHT
 riscv_encoding.h, 161
 SIP_SSIP
 riscv_encoding.h, 161
 SIP_STIP
 riscv_encoding.h, 161
 SPIN_LOCK_INITIALIZER
 riscv_locks.h, 172
 SPIN_LOCK_INIT
 riscv_locks.h, 172
 SSTATUS32_SD
 riscv_encoding.h, 162
 SSTATUS64_SD
 riscv_encoding.h, 162
 SSTATUS64_UXL
 riscv_encoding.h, 162
 SSTATUS_FS
 riscv_encoding.h, 162
 SSTATUS_MXR
 riscv_encoding.h, 162
 SSTATUS_SIE
 riscv_encoding.h, 162
 SSTATUS_SPIE_SHIFT
 riscv_encoding.h, 163
 SSTATUS_SPIE
 riscv_encoding.h, 162
 SSTATUS_SPP_SHIFT
 riscv_encoding.h, 163
 SSTATUS_SPP
 riscv_encoding.h, 163
 SSTATUS_SUM
 riscv_encoding.h, 163
 SSTATUS_SD
 riscv_encoding.h, 162
 SSTATUS_XS
 riscv_encoding.h, 163
 STRINGIFY
 sbi_const.h, 188
 STR
 sbi_types.h, 305
 SZREG
 riscv_asm.h, 81
 sbi_bitops.h
 __ffs, 179
 __fls, 179
 ffs, 179
 ffz, 179
 fls, 180
 sbi_bits.h
 BIT_MASK, 180
 BIT_WORD, 180
 EXTRACT_FIELD, 181
 INSERT_FIELD, 181
 sbi_boot_prints
 sbi_init.c, 396
 sbi_console.c
 console_out_lock, 353
 console_plat, 353
 PAD_ALTERNATE, 344
 PAD_RIGHT, 344
 PAD_ZERO, 344
 PRINT_BUF_LEN, 344
 print, 345
 putc, 345
 printi, 346
 prints, 347
 sbi_console_init, 347
 sbi_dprintf, 348
 sbi_getc, 349
 sbi_gets, 349
 sbi_isprintable, 350
 sbi_printf, 350
 sbi_putc, 351
 sbi_puts, 351
 sbi_snprintf, 352
 sbi_sprintf, 352
 va_arg, 344
 va_end, 344
 va_list, 345
 va_start, 344
 sbi_console.h
 __printf, 182, 183
 format, 186
 out_sz, 186
 sbi_console_init, 183
 sbi_getc, 183

- sbi_gets, 184
- sbi_isprintable, 184
- sbi_putc, 185
- sbi_puts, 185
- sbi_console_init
 - sbi_console.c, 347
 - sbi_console.h, 183
- sbi_const.h
 - _AC, 187
 - _AT, 187
 - _BITUL, 187
 - _BITULL, 188
 - _UL, 188
 - _ULL, 188
 - __AC, 187
 - __STR, 187
 - STRINGIFY, 188
 - ULL, 188
 - UL, 188
- sbi_current_hartid
 - sbi_hart.c, 382
 - sbi_hart.h, 212
- sbi_dlist, 51
 - next, 51
 - prev, 51
- sbi_dprintf
 - sbi_console.c, 348
- sbi_ecall.c
 - SBI_LIST_HEAD, 357
 - sbi_ecall_find_extension, 354
 - sbi_ecall_handler, 354
 - sbi_ecall_init, 355
 - sbi_ecall_register_extension, 356
 - sbi_ecall_unregister_extension, 356
 - sbi_ecall_version_major, 357
 - sbi_ecall_version_minor, 357
- sbi_ecall.h
 - ecall_base, 194
 - ecall_ipi, 194
 - ecall_legacy, 194
 - ecall_rfence, 194
 - ecall_time, 194
 - ecall_vendor, 194
 - SBI_ECALL_VERSION_MAJOR, 190
 - SBI_ECALL_VERSION_MINOR, 190
 - SBI_OPENSBI_IMPID, 190
 - sbi_ecall_find_extension, 190
 - sbi_ecall_handler, 190
 - sbi_ecall_init, 191
 - sbi_ecall_register_extension, 192
 - sbi_ecall_unregister_extension, 192
 - sbi_ecall_version_major, 193
 - sbi_ecall_version_minor, 193
- sbi_ecall_base.c
 - ecall_base, 360
 - sbi_ecall_base_handler, 358
 - sbi_ecall_base_probe, 359
- sbi_ecall_base_handler
 - sbi_ecall_base.c, 358
- sbi_ecall_base_probe
 - sbi_ecall_base.c, 359
- sbi_ecall_extension, 52
 - extid_end, 52
 - extid_start, 52
 - handle, 52
 - head, 52
 - probe, 52
- sbi_ecall_find_extension
 - sbi_ecall.c, 354
 - sbi_ecall.h, 190
- sbi_ecall_handler
 - sbi_ecall.c, 354
 - sbi_ecall.h, 190
- sbi_ecall_init
 - sbi_ecall.c, 355
 - sbi_ecall.h, 191
- sbi_ecall_interface.h
 - SBI_EXT_0_1_CLEAR_IPI, 195
 - SBI_EXT_0_1_CONSOLE_GETCHAR, 195
 - SBI_EXT_0_1_CONSOLE_PUTCHAR, 195
 - SBI_EXT_0_1_REMOTE_FENCE_I, 196
 - SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID, 196
 - SBI_EXT_0_1_REMOTE_SFENCE_VMA, 196
 - SBI_EXT_0_1_SEND_IPI, 196
 - SBI_EXT_0_1_SET_TIMER, 196
 - SBI_EXT_0_1_SHUTDOWN, 196
 - SBI_EXT_BASE_GET_IMP_ID, 196
 - SBI_EXT_BASE_GET_IMP_VERSION, 197
 - SBI_EXT_BASE_GET_MARCHID, 197
 - SBI_EXT_BASE_GET_MIMPID, 197
 - SBI_EXT_BASE_GET_MVENDORID, 197
 - SBI_EXT_BASE_GET_SPEC_VERSION, 197
 - SBI_EXT_BASE_PROBE_EXT, 197
 - SBI_EXT_BASE, 196
 - SBI_EXT_IPI_SEND_IPI, 197
 - SBI_EXT_IPI, 197
 - SBI_EXT_RFENCE_REMOTE_FENCE_I, 198
 - SBI_EXT_RFENCE_REMOTE_HFENCE_GVM↔A_V MID, 198
 - SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA, 198
 - SBI_EXT_RFENCE_REMOTE_HFENCE_VVM↔A_ASID, 198
 - SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA, 198
 - SBI_EXT_RFENCE_REMOTE_SFENCE_VMA↔ASID, 198
 - SBI_EXT_RFENCE_REMOTE_SFENCE_VMA, 198
 - SBI_EXT_RFENCE, 198
 - SBI_EXT_TIME_SET_TIMER, 199
 - SBI_EXT_TIME, 199
 - SBI_EXT_VENDOR_END, 199
 - SBI_EXT_VENDOR_START, 199
 - SBI_SPEC_VERSION_MAJOR_MASK, 199

- SBI_SPEC_VERSION_MAJOR_OFFSET, 199
- SBI_SPEC_VERSION_MINOR_MASK, 199
- sbi_ecall_ipi_handler
 - sbi_ecall_replace.c, 363
- sbi_ecall_legacy.c
 - ecall_legacy, 362
 - sbi_ecall_legacy_handler, 361
 - sbi_load_hart_mask_unpriv, 361
- sbi_ecall_legacy_handler
 - sbi_ecall_legacy.c, 361
- sbi_ecall_register_extension
 - sbi_ecall.c, 356
 - sbi_ecall.h, 192
- sbi_ecall_replace.c
 - ecall_ipi, 364
 - ecall_rfence, 365
 - ecall_time, 365
 - sbi_ecall_ipi_handler, 363
 - sbi_ecall_rfence_handler, 363
 - sbi_ecall_time_handler, 364
- sbi_ecall_rfence_handler
 - sbi_ecall_replace.c, 363
- sbi_ecall_time_handler
 - sbi_ecall_replace.c, 364
- sbi_ecall_unregister_extension
 - sbi_ecall.c, 356
 - sbi_ecall.h, 192
- sbi_ecall_vendor.c
 - ecall_vendor, 367
 - sbi_ecall_vendor_handler, 366
 - sbi_ecall_vendor_probe, 367
- sbi_ecall_vendor_handler
 - sbi_ecall_vendor.c, 366
- sbi_ecall_vendor_probe
 - sbi_ecall_vendor.c, 367
- sbi_ecall_version_major
 - sbi_ecall.c, 357
 - sbi_ecall.h, 193
- sbi_ecall_version_minor
 - sbi_ecall.c, 357
 - sbi_ecall.h, 193
- sbi_emulate_csr.c
 - sbi_emulate_csr_read, 368
 - sbi_emulate_csr_write, 369
- sbi_emulate_csr.h
 - sbi_emulate_csr_read, 200
 - sbi_emulate_csr_write, 201
- sbi_emulate_csr_read
 - sbi_emulate_csr.c, 368
 - sbi_emulate_csr.h, 200
- sbi_emulate_csr_write
 - sbi_emulate_csr.c, 369
 - sbi_emulate_csr.h, 201
- sbi_error.h
 - SBI_DENIED, 202
 - SBI_EFAIL, 202
 - SBI_EILL, 203
 - SBI_EINVAL, 203
 - SBI_EIO, 203
 - SBI_ENODEV, 203
 - SBI_ENOENT, 203
 - SBI_ENOMEM, 203
 - SBI_ENOSPC, 203
 - SBI_ENOSYS, 203
 - SBI_ENOTSUPP, 204
 - SBI_ETIMEDOUT, 204
 - SBI_ETRAP, 204
 - SBI_EUNKNOWN, 204
 - SBI_INVALID_ADDR, 204
 - SBI_OK, 204
- sbi_exit
 - sbi_init.c, 397
 - sbi_init.h, 225
- sbi_fifo, 53
 - avail, 53
 - entry_size, 53
 - num_entries, 53
 - qlock, 53
 - queue, 54
 - tail, 54
- sbi_fifo.c
 - __sbi_fifo_enqueue, 371
 - __sbi_fifo_is_empty, 371
 - __sbi_fifo_is_full, 371
 - __sbi_fifo_reset, 372
 - sbi_fifo_avail, 372
 - sbi_fifo_dequeue, 373
 - sbi_fifo_enqueue, 374
 - sbi_fifo_init, 374
 - sbi_fifo_inplace_update, 375
 - sbi_fifo_is_empty, 376
 - sbi_fifo_is_full, 376
 - sbi_fifo_reset, 377
- sbi_fifo.h
 - sbi_fifo_avail, 206
 - sbi_fifo_dequeue, 206
 - sbi_fifo_enqueue, 207
 - sbi_fifo_init, 208
 - sbi_fifo_inplace_update, 209
 - sbi_fifo_inplace_update_types, 206
 - sbi_fifo_is_empty, 209
 - sbi_fifo_is_full, 210
- sbi_fifo_avail
 - sbi_fifo.c, 372
 - sbi_fifo.h, 206
- sbi_fifo_dequeue
 - sbi_fifo.c, 373
 - sbi_fifo.h, 206
- sbi_fifo_enqueue
 - sbi_fifo.c, 374
 - sbi_fifo.h, 207
- sbi_fifo_init
 - sbi_fifo.c, 374
 - sbi_fifo.h, 208
- sbi_fifo_inplace_update
 - sbi_fifo.c, 375

- sbi_fifo.h, 209
- sbi_fifo_inplace_update_types
 - sbi_fifo.h, 206
- sbi_fifo_is_empty
 - sbi_fifo.c, 376
 - sbi_fifo.h, 209
- sbi_fifo_is_full
 - sbi_fifo.c, 376
 - sbi_fifo.h, 210
- sbi_fifo_reset
 - sbi_fifo.c, 377
- sbi_get_insn
 - sbi_unpriv.c, 445
 - sbi_unpriv.h, 310
- sbi_getc
 - sbi_console.c, 349
 - sbi_console.h, 183
- sbi_gets
 - sbi_console.c, 349
 - sbi_console.h, 184
- sbi_hart.c
 - __attribute__, 379
 - avail_hart_mask, 389
 - avail_hart_mask_lock, 389
 - COLDBOOT_WAIT_BITMAP_SIZE, 379
 - coldboot_done, 389
 - coldboot_lock, 389
 - coldboot_wait_bitmap, 389
 - delegate_traps, 379
 - fp_init, 380
 - h2s, 379
 - log2roundup, 380
 - mstatus_init, 380
 - pmp_init, 381
 - sbi_current_hartid, 382
 - sbi_hart_available_mask, 382
 - sbi_hart_delegation_dump, 383
 - sbi_hart_get_trap_info, 383
 - sbi_hart_id_to_scratch, 384
 - sbi_hart_init, 384
 - sbi_hart_mark_available, 385
 - sbi_hart_pmp_dump, 386
 - sbi_hart_set_trap_info, 387
 - sbi_hart_unmark_available, 387
 - sbi_hart_wait_for_coldboot, 387
 - sbi_hart_wake_coldboot_harts, 388
 - trap_info_offset, 389
- sbi_hart.h
 - __attribute__, 212
 - arg1, 219
 - next_addr, 219
 - next_mode, 219
 - next_virt, 219
 - sbi_current_hartid, 212
 - sbi_hart_available_mask, 212
 - sbi_hart_delegation_dump, 213
 - sbi_hart_get_trap_info, 213
 - sbi_hart_id_to_scratch, 214
 - sbi_hart_init, 214
 - sbi_hart_mark_available, 215
 - sbi_hart_pmp_dump, 216
 - sbi_hart_set_trap_info, 217
 - sbi_hart_unmark_available, 217
 - sbi_hart_wait_for_coldboot, 217
 - sbi_hart_wake_coldboot_harts, 218
- sbi_hart_available_mask
 - sbi_hart.c, 382
 - sbi_hart.h, 212
- sbi_hart_delegation_dump
 - sbi_hart.c, 383
 - sbi_hart.h, 213
- sbi_hart_get_trap_info
 - sbi_hart.c, 383
 - sbi_hart.h, 213
- sbi_hart_id_to_scratch
 - sbi_hart.c, 384
 - sbi_hart.h, 214
- sbi_hart_init
 - sbi_hart.c, 384
 - sbi_hart.h, 214
- sbi_hart_mark_available
 - sbi_hart.c, 385
 - sbi_hart.h, 215
- sbi_hart_pmp_dump
 - sbi_hart.c, 386
 - sbi_hart.h, 216
- sbi_hart_set_trap_info
 - sbi_hart.c, 387
 - sbi_hart.h, 217
- sbi_hart_unmark_available
 - sbi_hart.c, 387
 - sbi_hart.h, 217
- sbi_hart_wait_for_coldboot
 - sbi_hart.c, 387
 - sbi_hart.h, 217
- sbi_hart_wake_coldboot_harts
 - sbi_hart.c, 388
 - sbi_hart.h, 218
- sbi_hfence.h
 - __sbi_hfence_gvma_all, 220
 - __sbi_hfence_gvma_gpa, 220
 - __sbi_hfence_gvma_vmid, 221
 - __sbi_hfence_gvma_vmid_gpa, 221
 - __sbi_hfence_vvma_all, 221
 - __sbi_hfence_vvma_asid, 222
 - __sbi_hfence_vvma_asid_va, 222
 - __sbi_hfence_vvma_va, 222
- sbi_illegal_insn.c
 - illegal_insn_func, 390
 - illegal_insn_table, 393
 - sbi_illegal_insn_handler, 391
 - system_opcode_insn, 391
 - truly_illegal_insn, 392
- sbi_illegal_insn.h
 - sbi_illegal_insn_handler, 224
- sbi_illegal_insn_handler

- sbi_illegal_insn.c, 391
 - sbi_illegal_insn.h, 224
- sbi_init
 - sbi_init.c, 398
 - sbi_init.h, 227
- sbi_init.c
 - BANNER, 394
 - coldboot_lottery, 400
 - init_coldboot, 394
 - init_count_offset, 400
 - init_warmboot, 395
 - sbi_boot_prints, 396
 - sbi_exit, 397
 - sbi_init, 398
 - sbi_init_count, 399
- sbi_init.h
 - sbi_exit, 225
 - sbi_init, 227
 - sbi_init_count, 228
- sbi_init_count
 - sbi_init.c, 399
 - sbi_init.h, 228
- sbi_ipi.c
 - ipi_data_off, 408
 - ipi_halt_event, 408
 - ipi_halt_ops, 408
 - ipi_ops_array, 409
 - ipi_smode_event, 409
 - ipi_smode_ops, 409
 - sbi_ipi_clear_smode, 401
 - sbi_ipi_event_create, 401
 - sbi_ipi_event_destroy, 402
 - sbi_ipi_exit, 402
 - sbi_ipi_init, 403
 - sbi_ipi_process, 403
 - sbi_ipi_process_halt, 404
 - sbi_ipi_process_smode, 405
 - sbi_ipi_send, 405
 - sbi_ipi_send_halt, 406
 - sbi_ipi_send_many, 406
 - sbi_ipi_send_smode, 407
- sbi_ipi.h
 - SBI_IPI_EVENT_MAX, 230
 - sbi_ipi_clear_smode, 230
 - sbi_ipi_event_create, 230
 - sbi_ipi_event_destroy, 231
 - sbi_ipi_exit, 231
 - sbi_ipi_init, 232
 - sbi_ipi_process, 232
 - sbi_ipi_send_halt, 233
 - sbi_ipi_send_many, 234
 - sbi_ipi_send_smode, 235
- sbi_ipi_clear_smode
 - sbi_ipi.c, 401
 - sbi_ipi.h, 230
- sbi_ipi_event_create
 - sbi_ipi.c, 401
 - sbi_ipi.h, 230
- sbi_ipi_event_destroy
 - sbi_ipi.c, 402
 - sbi_ipi.h, 231
- sbi_ipi_event_ops, 54
 - name, 54
 - process, 54
 - sync, 55
 - update, 55
- sbi_ipi_exit
 - sbi_ipi.c, 402
 - sbi_ipi.h, 231
- sbi_ipi_init
 - sbi_ipi.c, 403
 - sbi_ipi.h, 232
- sbi_ipi_process
 - sbi_ipi.c, 403
 - sbi_ipi.h, 232
- sbi_ipi_process_halt
 - sbi_ipi.c, 404
- sbi_ipi_process_smode
 - sbi_ipi.c, 405
- sbi_ipi_send
 - sbi_ipi.c, 405
- sbi_ipi_send_halt
 - sbi_ipi.c, 406
 - sbi_ipi.h, 233
- sbi_ipi_send_many
 - sbi_ipi.c, 406
 - sbi_ipi.h, 234
- sbi_ipi_send_smode
 - sbi_ipi.c, 407
 - sbi_ipi.h, 235
- sbi_isprintable
 - sbi_console.c, 350
 - sbi_console.h, 184
- sbi_list.h
 - __sbi_list_add, 239
 - __sbi_list_del, 240
 - __sbi_list_del_entry, 240
 - SBI_INIT_LIST_HEAD, 237
 - SBI_LIST_HEAD_INIT, 238
 - SBI_LIST_HEAD, 238
 - SBI_LIST_POISON_NEXT, 239
 - SBI_LIST_POISON_PREV, 239
 - sbi_list_add, 241
 - sbi_list_add_tail, 241
 - sbi_list_del, 242
 - sbi_list_del_init, 242
 - sbi_list_entry, 237
 - sbi_list_first_entry, 237
 - sbi_list_for_each, 238
 - sbi_list_for_each_entry, 238
 - sbi_list_last_entry, 239
- sbi_list_add
 - sbi_list.h, 241
- sbi_list_add_tail
 - sbi_list.h, 241
- sbi_list_del
 - sbi_list.h, 241

- sbi_list.h, [242](#)
- sbi_list_del_init
 - sbi_list.h, [242](#)
- sbi_list_entry
 - sbi_list.h, [237](#)
- sbi_list_first_entry
 - sbi_list.h, [237](#)
- sbi_list_for_each
 - sbi_list.h, [238](#)
- sbi_list_for_each_entry
 - sbi_list.h, [238](#)
- sbi_list_last_entry
 - sbi_list.h, [239](#)
- sbi_load_hart_mask_unpriv
 - sbi_ecall_legacy.c, [361](#)
- sbi_load_u64
 - sbi_unpriv.c, [446](#)
- sbi_memchr
 - sbi_string.c, [415](#)
 - sbi_string.h, [277](#)
- sbi_memcmp
 - sbi_string.c, [415](#)
 - sbi_string.h, [277](#)
- sbi_memcpy
 - sbi_string.c, [415](#)
 - sbi_string.h, [277](#)
- sbi_memmove
 - sbi_string.c, [415](#)
 - sbi_string.h, [277](#)
- sbi_memset
 - sbi_string.c, [416](#)
 - sbi_string.h, [277](#)
- sbi_misaligned_ldst.c
 - sbi_misaligned_load_handler, [410](#)
 - sbi_misaligned_store_handler, [411](#)
- sbi_misaligned_ldst.h
 - sbi_misaligned_load_handler, [244](#)
 - sbi_misaligned_store_handler, [245](#)
- sbi_misaligned_load_handler
 - sbi_misaligned_ldst.c, [410](#)
 - sbi_misaligned_ldst.h, [244](#)
- sbi_misaligned_store_handler
 - sbi_misaligned_ldst.c, [411](#)
 - sbi_misaligned_ldst.h, [245](#)
- sbi_platform, [55](#)
 - disabled_hart_mask, [56](#)
 - features, [56](#)
 - firmware_context, [56](#)
 - hart_count, [56](#)
 - hart_stack_size, [56](#)
 - name, [56](#)
 - opensbi_version, [56](#)
 - platform_ops_addr, [56](#)
 - platform_version, [57](#)
- sbi_platform.h
 - __packed, [270](#)
 - SBI_PLATFORM_DEFAULT_FEATURES, [248](#)
 - SBI_PLATFORM_DISABLED_HART_OFFSET, [248](#)
 - SBI_PLATFORM_FEATURES_OFFSET, [248](#)
 - SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET, [248](#)
 - SBI_PLATFORM_HART_COUNT_OFFSET, [248](#)
 - SBI_PLATFORM_HART_STACK_SIZE_OFFSET, [248](#)
 - SBI_PLATFORM_NAME_OFFSET, [249](#)
 - SBI_PLATFORM_OPENSBI_VERSION_OFFSET, [250](#)
 - SBI_PLATFORM_OPS_OFFSET, [250](#)
 - SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT, [250](#)
 - SBI_PLATFORM_VERSION_OFFSET, [251](#)
 - SBI_PLATFORM_VERSION, [250](#)
 - sbi_platform_console_getc, [251](#)
 - sbi_platform_console_init, [252](#)
 - sbi_platform_console_putc, [252](#)
 - sbi_platform_early_exit, [253](#)
 - sbi_platform_early_init, [253](#)
 - sbi_platform_features, [251](#)
 - sbi_platform_final_exit, [254](#)
 - sbi_platform_final_init, [255](#)
 - sbi_platform_hart_count, [255](#)
 - sbi_platform_hart_disabled, [256](#)
 - sbi_platform_hart_stack_size, [257](#)
 - sbi_platform_has_hart_hotplug, [249](#)
 - sbi_platform_has_mcounteren, [249](#)
 - sbi_platform_has_mfaults_delegation, [249](#)
 - sbi_platform_has_pmp, [249](#)
 - sbi_platform_has_scounteren, [249](#)
 - sbi_platform_has_timer_value, [249](#)
 - sbi_platform_ipi_clear, [257](#)
 - sbi_platform_ipi_exit, [258](#)
 - sbi_platform_ipi_init, [258](#)
 - sbi_platform_ipi_send, [259](#)
 - sbi_platform_irqchip_exit, [259](#)
 - sbi_platform_irqchip_init, [260](#)
 - sbi_platform_misa_extension, [260](#)
 - sbi_platform_misa_xlen, [261](#)
 - sbi_platform_name, [261](#)
 - sbi_platform_ops, [250](#)
 - sbi_platform_pmp_region_count, [262](#)
 - sbi_platform_pmp_region_info, [263](#)
 - sbi_platform_ptr, [250](#)
 - sbi_platform_system_reboot, [263](#)
 - sbi_platform_system_shutdown, [264](#)
 - sbi_platform_thishart_ptr, [250](#)
 - sbi_platform_timer_event_start, [265](#)
 - sbi_platform_timer_event_stop, [265](#)
 - sbi_platform_timer_exit, [266](#)
 - sbi_platform_timer_init, [266](#)
 - sbi_platform_timer_value, [267](#)
 - sbi_platform_tlbr_flush_limit, [267](#)
 - sbi_platform_vendor_ext_check, [268](#)
 - sbi_platform_vendor_ext_provider, [269](#)
 - sbi_platform_console_getc

- sbi_platform.h, 251
- sbi_platform_console_init
 - sbi_platform.h, 252
- sbi_platform_console_putc
 - sbi_platform.h, 252
- sbi_platform_early_exit
 - sbi_platform.h, 253
- sbi_platform_early_init
 - sbi_platform.h, 253
- sbi_platform_features
 - sbi_platform.h, 251
- sbi_platform_final_exit
 - sbi_platform.h, 254
- sbi_platform_final_init
 - sbi_platform.h, 255
- sbi_platform_hart_count
 - sbi_platform.h, 255
- sbi_platform_hart_disabled
 - sbi_platform.h, 256
- sbi_platform_hart_stack_size
 - sbi_platform.h, 257
- sbi_platform_has_hart_hotplug
 - sbi_platform.h, 249
- sbi_platform_has_mcounteren
 - sbi_platform.h, 249
- sbi_platform_has_mfaults_delegation
 - sbi_platform.h, 249
- sbi_platform_has_pmp
 - sbi_platform.h, 249
- sbi_platform_has_scounteren
 - sbi_platform.h, 249
- sbi_platform_has_timer_value
 - sbi_platform.h, 249
- sbi_platform_ipi_clear
 - sbi_platform.h, 257
- sbi_platform_ipi_exit
 - sbi_platform.h, 258
- sbi_platform_ipi_init
 - sbi_platform.h, 258
- sbi_platform_ipi_send
 - sbi_platform.h, 259
- sbi_platform_irqchip_exit
 - sbi_platform.h, 259
- sbi_platform_irqchip_init
 - sbi_platform.h, 260
- sbi_platform_misa_extension
 - sbi_platform.h, 260
- sbi_platform_misa_xlen
 - sbi_platform.h, 261
- sbi_platform_name
 - sbi_platform.h, 261
- sbi_platform_operations, 57
 - console_getc, 58
 - console_init, 58
 - console_putc, 58
 - early_exit, 58
 - early_init, 58
 - final_exit, 58
 - final_init, 58
 - get_tlbr_flush_limit, 59
 - ipi_clear, 59
 - ipi_exit, 59
 - ipi_init, 59
 - ipi_send, 59
 - irqchip_exit, 59
 - irqchip_init, 59
 - misa_check_extension, 59
 - misa_get_xlen, 60
 - pmp_region_count, 60
 - pmp_region_info, 60
 - system_reboot, 60
 - system_shutdown, 60
 - timer_event_start, 60
 - timer_event_stop, 60
 - timer_exit, 61
 - timer_init, 61
 - timer_value, 61
 - vendor_ext_check, 61
 - vendor_ext_provider, 61
- sbi_platform_ops
 - sbi_platform.h, 250
- sbi_platform_pmp_region_count
 - sbi_platform.h, 262
- sbi_platform_pmp_region_info
 - sbi_platform.h, 263
- sbi_platform_ptr
 - sbi_platform.h, 250
- sbi_platform_system_reboot
 - sbi_platform.h, 263
- sbi_platform_system_shutdown
 - sbi_platform.h, 264
- sbi_platform_thishart_ptr
 - sbi_platform.h, 250
- sbi_platform_timer_event_start
 - sbi_platform.h, 265
- sbi_platform_timer_event_stop
 - sbi_platform.h, 265
- sbi_platform_timer_exit
 - sbi_platform.h, 266
- sbi_platform_timer_init
 - sbi_platform.h, 266
- sbi_platform_timer_value
 - sbi_platform.h, 267
- sbi_platform_tlbr_flush_limit
 - sbi_platform.h, 267
- sbi_platform_vendor_ext_check
 - sbi_platform.h, 268
- sbi_platform_vendor_ext_provider
 - sbi_platform.h, 269
- sbi_printf
 - sbi_console.c, 350
- sbi_putc
 - sbi_console.c, 351
 - sbi_console.h, 185
- sbi_puts
 - sbi_console.c, 351

- sbi_console.h, 185
- sbi_scratch, 61
 - fw_size, 62
 - fw_start, 62
 - hartid_to_scratch, 62
 - next_addr, 62
 - next_arg1, 62
 - next_mode, 63
 - options, 63
 - platform_addr, 63
 - tmp0, 63
 - warmboot_addr, 63
- sbi_scratch.c
 - extra_lock, 414
 - extra_offset, 414
 - sbi_scratch_alloc_offset, 413
 - sbi_scratch_free_offset, 413
- sbi_scratch.h
 - __packed, 276
 - SBI_SCRATCH_EXTRA_SPACE_OFFSET, 271
 - SBI_SCRATCH_FW_SIZE_OFFSET, 272
 - SBI_SCRATCH_FW_START_OFFSET, 272
 - SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET, 272
 - SBI_SCRATCH_NEXT_ADDR_OFFSET, 272
 - SBI_SCRATCH_NEXT_ARG1_OFFSET, 272
 - SBI_SCRATCH_NEXT_MODE_OFFSET, 272
 - SBI_SCRATCH_OPTIONS_OFFSET, 273
 - SBI_SCRATCH_PLATFORM_ADDR_OFFSET, 273
 - SBI_SCRATCH_SIZE, 273
 - SBI_SCRATCH_TMP0_OFFSET, 273
 - SBI_SCRATCH_WARMBOOT_ADDR_OFFSET, 274
 - sbi_scratch_alloc_offset, 274
 - sbi_scratch_free_offset, 275
 - sbi_scratch_offset_ptr, 272
 - sbi_scratch_options, 274
 - sbi_scratch_thishart_arg1_ptr, 273
 - sbi_scratch_thishart_offset_ptr, 273
 - sbi_scratch_thishart_ptr, 273
- sbi_scratch_alloc_offset
 - sbi_scratch.c, 413
 - sbi_scratch.h, 274
- sbi_scratch_free_offset
 - sbi_scratch.c, 413
 - sbi_scratch.h, 275
- sbi_scratch_offset_ptr
 - sbi_scratch.h, 272
- sbi_scratch_options
 - sbi_scratch.h, 274
- sbi_scratch_thishart_arg1_ptr
 - sbi_scratch.h, 273
- sbi_scratch_thishart_offset_ptr
 - sbi_scratch.h, 273
- sbi_scratch_thishart_ptr
 - sbi_scratch.h, 273
- sbi_snprintf
 - sbi_console.c, 352
- sbi_sprintf
 - sbi_console.c, 352
- sbi_store_u64
 - sbi_unpriv.c, 446
- sbi_strchr
 - sbi_string.c, 416
 - sbi_string.h, 278
- sbi_strcmp
 - sbi_string.c, 416
 - sbi_string.h, 278
- sbi_strcpy
 - sbi_string.c, 416
 - sbi_string.h, 278
- sbi_string.c
 - sbi_memchr, 415
 - sbi_memcmp, 415
 - sbi_memcpy, 415
 - sbi_memmove, 415
 - sbi_memset, 416
 - sbi_strchr, 416
 - sbi_strcmp, 416
 - sbi_strcpy, 416
 - sbi_strlen, 417
 - sbi_strncpy, 417
 - sbi_strnlen, 417
 - sbi_strrchr, 417
- sbi_string.h
 - sbi_memchr, 277
 - sbi_memcmp, 277
 - sbi_memcpy, 277
 - sbi_memmove, 277
 - sbi_memset, 277
 - sbi_strchr, 278
 - sbi_strcmp, 278
 - sbi_strcpy, 278
 - sbi_strlen, 278
 - sbi_strncpy, 279
 - sbi_strnlen, 279
 - sbi_strrchr, 279
- sbi_strlen
 - sbi_string.c, 417
 - sbi_string.h, 278
- sbi_strncpy
 - sbi_string.c, 417
 - sbi_string.h, 279
- sbi_strnlen
 - sbi_string.c, 417
 - sbi_string.h, 279
- sbi_strrchr
 - sbi_string.c, 417
 - sbi_string.h, 279
- sbi_system.c
 - sbi_system_early_exit, 419
 - sbi_system_early_init, 419
 - sbi_system_final_exit, 419
 - sbi_system_final_init, 420
 - sbi_system_reboot, 420

- sbi_system_shutdown, [421](#)
- sbi_system.h
 - sbi_system_early_exit, [280](#)
 - sbi_system_early_init, [281](#)
 - sbi_system_final_exit, [281](#)
 - sbi_system_final_init, [282](#)
 - sbi_system_reboot, [282](#)
 - sbi_system_shutdown, [283](#)
- sbi_system_early_exit
 - sbi_system.c, [419](#)
 - sbi_system.h, [280](#)
- sbi_system_early_init
 - sbi_system.c, [419](#)
 - sbi_system.h, [281](#)
- sbi_system_final_exit
 - sbi_system.c, [419](#)
 - sbi_system.h, [281](#)
- sbi_system_final_init
 - sbi_system.c, [420](#)
 - sbi_system.h, [282](#)
- sbi_system_reboot
 - sbi_system.c, [420](#)
 - sbi_system.h, [282](#)
- sbi_system_shutdown
 - sbi_system.c, [421](#)
 - sbi_system.h, [283](#)
- sbi_timer.c
 - get_ticks, [423](#)
 - sbi_timer_event_start, [423](#)
 - sbi_timer_exit, [424](#)
 - sbi_timer_get_delta, [424](#)
 - sbi_timer_init, [424](#)
 - sbi_timer_process, [425](#)
 - sbi_timer_set_delta, [425](#)
 - sbi_timer_set_delta_upper, [426](#)
 - sbi_timer_value, [426](#)
 - sbi_timer_virt_value, [427](#)
 - time_delta_off, [427](#)
- sbi_timer.h
 - sbi_timer_event_start, [285](#)
 - sbi_timer_exit, [285](#)
 - sbi_timer_get_delta, [286](#)
 - sbi_timer_init, [286](#)
 - sbi_timer_process, [287](#)
 - sbi_timer_set_delta, [287](#)
 - sbi_timer_set_delta_upper, [288](#)
 - sbi_timer_value, [288](#)
 - sbi_timer_virt_value, [289](#)
- sbi_timer_event_start
 - sbi_timer.c, [423](#)
 - sbi_timer.h, [285](#)
- sbi_timer_exit
 - sbi_timer.c, [424](#)
 - sbi_timer.h, [285](#)
- sbi_timer_get_delta
 - sbi_timer.c, [424](#)
 - sbi_timer.h, [286](#)
- sbi_timer_init
 - sbi_timer.c, [424](#)
 - sbi_timer.h, [286](#)
- sbi_timer.c, [424](#)
- sbi_timer.h, [286](#)
- sbi_timer_process
 - sbi_timer.c, [425](#)
 - sbi_timer.h, [287](#)
- sbi_timer_set_delta
 - sbi_timer.c, [425](#)
 - sbi_timer.h, [287](#)
- sbi_timer_set_delta_upper
 - sbi_timer.c, [426](#)
 - sbi_timer.h, [288](#)
- sbi_timer_value
 - sbi_timer.c, [426](#)
 - sbi_timer.h, [288](#)
- sbi_timer_virt_value
 - sbi_timer.c, [427](#)
 - sbi_timer.h, [289](#)
- sbi_tlb.c
 - __sbi_tlb_range_check, [429](#)
 - sbi_tlb_entry_process, [429](#)
 - sbi_tlb_flush_all, [430](#)
 - sbi_tlb_hfence_gvma, [430](#)
 - sbi_tlb_hfence_gvma_vmid, [430](#)
 - sbi_tlb_hfence_vvma, [431](#)
 - sbi_tlb_hfence_vvma_asid, [432](#)
 - sbi_tlb_init, [432](#)
 - sbi_tlb_local_flush, [433](#)
 - sbi_tlb_process, [434](#)
 - sbi_tlb_process_count, [434](#)
 - sbi_tlb_request, [435](#)
 - sbi_tlb_sfence_vma, [436](#)
 - sbi_tlb_sfence_vma_asid, [437](#)
 - sbi_tlb_sync, [437](#)
 - sbi_tlb_update, [438](#)
 - sbi_tlb_update_cb, [438](#)
 - tlb_event, [439](#)
 - tlb_fifo_mem_off, [439](#)
 - tlb_fifo_off, [439](#)
 - tlb_ops, [440](#)
 - tlb_range_flush_limit, [440](#)
 - tlb_sync_off, [440](#)
- sbi_tlb.h
 - SBI_TLB_FIFO_NUM_ENTRIES, [291](#)
 - SBI_TLB_FLUSH_ALL, [291](#)
 - SBI_TLB_INFO_SIZE, [291](#)
 - sbi_tlb_info_types, [291](#)
 - sbi_tlb_init, [291](#)
 - sbi_tlb_request, [292](#)
- sbi_tlb_entry_process
 - sbi_tlb.c, [429](#)
- sbi_tlb_flush_all
 - sbi_tlb.c, [430](#)
- sbi_tlb_hfence_gvma
 - sbi_tlb.c, [430](#)
- sbi_tlb_hfence_gvma_vmid
 - sbi_tlb.c, [430](#)
- sbi_tlb_hfence_vvma
 - sbi_tlb.c, [431](#)

- sbi_tlb_hfence_vvma_asid
 - sbi_tlb.c, [432](#)
- sbi_tlb_info, [63](#)
 - asid, [64](#)
 - shart_mask, [64](#)
 - size, [64](#)
 - start, [64](#)
 - type, [64](#)
- sbi_tlb_info_types
 - sbi_tlb.h, [291](#)
- sbi_tlb_init
 - sbi_tlb.c, [432](#)
 - sbi_tlb.h, [291](#)
- sbi_tlb_local_flush
 - sbi_tlb.c, [433](#)
- sbi_tlb_process
 - sbi_tlb.c, [434](#)
- sbi_tlb_process_count
 - sbi_tlb.c, [434](#)
- sbi_tlb_request
 - sbi_tlb.c, [435](#)
 - sbi_tlb.h, [292](#)
- sbi_tlb_sfence_vma
 - sbi_tlb.c, [436](#)
- sbi_tlb_sfence_vma_asid
 - sbi_tlb.c, [437](#)
- sbi_tlb_sync
 - sbi_tlb.c, [437](#)
- sbi_tlb_update
 - sbi_tlb.c, [438](#)
- sbi_tlb_update_cb
 - sbi_tlb.c, [438](#)
- sbi_trap.c
 - sbi_trap_error, [441](#)
 - sbi_trap_handler, [441](#)
 - sbi_trap_redirect, [442](#)
- sbi_trap.h
 - __packed, [302](#)
 - SBI_TRAP_REGS_OFFSET, [296](#)
 - SBI_TRAP_REGS_SIZE, [298](#)
 - SBI_TRAP_REGS_a0, [295](#)
 - SBI_TRAP_REGS_a1, [295](#)
 - SBI_TRAP_REGS_a2, [295](#)
 - SBI_TRAP_REGS_a3, [295](#)
 - SBI_TRAP_REGS_a4, [295](#)
 - SBI_TRAP_REGS_a5, [295](#)
 - SBI_TRAP_REGS_a6, [296](#)
 - SBI_TRAP_REGS_a7, [296](#)
 - SBI_TRAP_REGS_gp, [296](#)
 - SBI_TRAP_REGS_last, [296](#)
 - SBI_TRAP_REGS_mepc, [296](#)
 - SBI_TRAP_REGS_mstatus, [296](#)
 - SBI_TRAP_REGS_mstatusH, [296](#)
 - SBI_TRAP_REGS_ra, [297](#)
 - SBI_TRAP_REGS_s0, [297](#)
 - SBI_TRAP_REGS_s1, [297](#)
 - SBI_TRAP_REGS_s10, [297](#)
 - SBI_TRAP_REGS_s11, [297](#)
 - SBI_TRAP_REGS_s2, [297](#)
 - SBI_TRAP_REGS_s3, [297](#)
 - SBI_TRAP_REGS_s4, [298](#)
 - SBI_TRAP_REGS_s5, [298](#)
 - SBI_TRAP_REGS_s6, [298](#)
 - SBI_TRAP_REGS_s7, [298](#)
 - SBI_TRAP_REGS_s8, [298](#)
 - SBI_TRAP_REGS_s9, [298](#)
 - SBI_TRAP_REGS_sp, [298](#)
 - SBI_TRAP_REGS_t0, [299](#)
 - SBI_TRAP_REGS_t1, [299](#)
 - SBI_TRAP_REGS_t2, [299](#)
 - SBI_TRAP_REGS_t3, [299](#)
 - SBI_TRAP_REGS_t4, [299](#)
 - SBI_TRAP_REGS_t5, [299](#)
 - SBI_TRAP_REGS_t6, [299](#)
 - SBI_TRAP_REGS_tp, [299](#)
 - SBI_TRAP_REGS_zero, [300](#)
 - sbi_trap_handler, [300](#)
 - sbi_trap_redirect, [301](#)
- sbi_trap_error
 - sbi_trap.c, [441](#)
- sbi_trap_handler
 - sbi_trap.c, [441](#)
 - sbi_trap.h, [300](#)
- sbi_trap_info, [65](#)
 - cause, [65](#)
 - epc, [65](#)
 - tinst, [65](#)
 - tval, [65](#)
 - tval2, [65](#)
- sbi_trap_redirect
 - sbi_trap.c, [442](#)
 - sbi_trap.h, [301](#)
- sbi_trap_regs, [66](#)
 - a0, [67](#)
 - a1, [67](#)
 - a2, [67](#)
 - a3, [67](#)
 - a4, [67](#)
 - a5, [67](#)
 - a6, [67](#)
 - a7, [68](#)
 - gp, [68](#)
 - mepc, [68](#)
 - mstatus, [68](#)
 - mstatusH, [68](#)
 - ra, [68](#)
 - s0, [68](#)
 - s1, [68](#)
 - s10, [69](#)
 - s11, [69](#)
 - s2, [69](#)
 - s3, [69](#)
 - s4, [69](#)
 - s5, [69](#)
 - s6, [69](#)
 - s7, [69](#)

- s8, [70](#)
- s9, [70](#)
- sp, [70](#)
- t0, [70](#)
- t1, [70](#)
- t2, [70](#)
- t3, [70](#)
- t4, [70](#)
- t5, [71](#)
- t6, [71](#)
- tp, [71](#)
- zero, [71](#)
- sbi_types.h
 - __noreturn, [303](#)
 - __packed, [303](#)
 - bool, [306](#)
 - CLAMP, [303](#)
 - container_of, [304](#)
 - FALSE, [304](#)
 - int16_t, [306](#)
 - int32_t, [306](#)
 - likely, [304](#)
 - MAX, [304](#)
 - MIN, [304](#)
 - NULL, [304](#)
 - offsetof, [305](#)
 - physical_addr_t, [306](#)
 - physical_size_t, [306](#)
 - ROUNDDOWN, [305](#)
 - ROUNDUP, [305](#)
 - s16, [306](#)
 - s32, [306](#)
 - s8, [307](#)
 - STR, [305](#)
 - size_t, [307](#)
 - ssize_t, [307](#)
 - TRUE, [305](#)
 - u16, [307](#)
 - u32, [307](#)
 - u8, [307](#)
 - uint16_t, [307](#)
 - uint32_t, [307](#)
 - uint8_t, [308](#)
 - uintptr_t, [308](#)
 - ulong, [308](#)
 - unlikely, [305](#)
 - virtual_addr_t, [308](#)
 - virtual_size_t, [308](#)
 - XSTR, [305](#)
- sbi_unpriv.c
 - DECLARE_UNPRIVILEGED_LOAD_FUNCTION, [309](#)
 - DECLARE_UNPRIVILEGED_STORE_FUNCTION, [310](#)
 - sbi_get_insn, [310](#)
- sbi_version.h
 - OPENSBI_VERSION_MAJOR, [311](#)
 - OPENSBI_VERSION_MINOR, [312](#)
 - OPENSBI_VERSION, [311](#)
- set_reg
 - sifive-uart.c, [638](#)
 - uart8250.c, [645](#)
- shart_mask
 - sbi_tlb_info, [64](#)
- sifive-uart.c
 - get_reg, [638](#)
 - set_reg, [638](#)
 - sifive_uart_getc, [638](#)
 - sifive_uart_init, [639](#)
 - sifive_uart_putc, [639](#)
 - UART_REG_DIV, [636](#)
 - UART_REG_IE, [636](#)
 - UART_REG_IP, [636](#)
 - UART_REG_RXCTRL, [636](#)
 - UART_REG_RXFIFO, [637](#)
 - UART_REG_TXCTRL, [637](#)
 - UART_REG_TXFIFO, [637](#)
 - UART_RXCTRL_RXEN, [637](#)
 - UART_RXFIFO_DATA, [637](#)
 - UART_RXFIFO_EMPTY, [637](#)
 - UART_TXCTRL_TXEN, [637](#)
 - UART_TXFIFO_FULL, [637](#)
 - uart_base, [640](#)
 - uart_baudrate, [640](#)
 - uart_in_freq, [641](#)
 - uart_min_clk_divisor, [640](#)
- sifive-uart.h
 - sifive_uart_getc, [316](#)
 - sifive_uart_init, [316](#)
 - sifive_uart_putc, [316](#)
- sifive_uart_getc
 - sifive-uart.c, [638](#)
 - sifive-uart.h, [316](#)
- sifive_uart_init
 - sifive-uart.c, [639](#)
 - sifive-uart.h, [316](#)
- sifive_uart_putc
 - sifive-uart.c, [639](#)
 - sifive-uart.h, [316](#)
- size
 - fdt_reserve_entry, [49](#)
 - sbi_tlb_info, [64](#)
- size_dt_strings
 - fdt_header, [47](#)
- size_dt_struct
 - fdt_header, [47](#)
- size_t
 - sbi_types.h, [307](#)

- smp_mb
 - riscv_barrier.h, [92](#)
- smp_rmb
 - riscv_barrier.h, [92](#)
- smp_wmb
 - riscv_barrier.h, [93](#)
- sp
 - sbi_trap_regs, [70](#)
- spin_lock
 - riscv_locks.c, [338](#)
 - riscv_locks.h, [173](#)
- spin_lock_check
 - riscv_locks.c, [339](#)
 - riscv_locks.h, [174](#)
- spin_trylock
 - riscv_locks.c, [340](#)
 - riscv_locks.h, [175](#)
- spin_unlock
 - riscv_locks.c, [341](#)
 - riscv_locks.h, [176](#)
- spinlock_t, [71](#)
 - lock, [71](#)
- ssize_t
 - sbi_types.h, [307](#)
- start
 - sbi_tlb_info, [64](#)
- strchr
 - libfdt_env.h, [624](#)
- strcmp
 - libfdt_env.h, [624](#)
- strcpy
 - libfdt_env.h, [624](#)
- strlen
 - libfdt_env.h, [624](#)
- strnlen
 - libfdt_env.h, [624](#)
- strrchr
 - libfdt_env.h, [624](#)
- sync
 - sbi_ipi_event_ops, [55](#)
- system_opcode_insn
 - sbi_illegal_insn.c, [391](#)
- system_reboot
 - sbi_platform_operations, [60](#)
- system_shutdown
 - sbi_platform_operations, [60](#)
- t0
 - sbi_trap_regs, [70](#)
- t1
 - sbi_trap_regs, [70](#)
- t2
 - sbi_trap_regs, [70](#)
- t3
 - sbi_trap_regs, [70](#)
- t4
 - sbi_trap_regs, [70](#)
- t5
 - sbi_trap_regs, [71](#)
- t6
 - sbi_trap_regs, [71](#)
- TOHOST_CMD
 - htif.c, [658](#)
- TRUE
 - sbi_types.h, [305](#)
- tag
 - fdt_node_header, [48](#)
 - fdt_property, [49](#)
- tail
 - sbi_fifo, [54](#)
- time_delta_off
 - sbi_timer.c, [427](#)
- timer_event_start
 - sbi_platform_operations, [60](#)
- timer_event_stop
 - sbi_platform_operations, [60](#)
- timer_exit
 - sbi_platform_operations, [61](#)
- timer_init
 - sbi_platform_operations, [61](#)
- timer_value
 - sbi_platform_operations, [61](#)
- tinst
 - sbi_trap_info, [65](#)
- tlb_event
 - sbi_tlb.c, [439](#)
- tlb_fifo_mem_off
 - sbi_tlb.c, [439](#)
- tlb_fifo_off
 - sbi_tlb.c, [439](#)
- tlb_ops
 - sbi_tlb.c, [440](#)
- tlb_range_flush_limit
 - sbi_tlb.c, [440](#)
- tlb_sync_off
 - sbi_tlb.c, [440](#)
- tmp0
 - sbi_scratch, [63](#)
- totalsize
 - fdt_header, [47](#)
- tp
 - sbi_trap_regs, [71](#)
- trap_info_offset
 - sbi_hart.c, [389](#)
- truly_illegal_insn
 - sbi_illegal_insn.c, [392](#)
- tval
 - sbi_trap_info, [65](#)
- tval2
 - sbi_trap_info, [65](#)
- type
 - sbi_tlb_info, [64](#)
- u16
 - sbi_types.h, [307](#)
- u32
 - sbi_types.h, [307](#)
- u8

- sbi_types.h, [307](#)
- UART_DLL_OFFSET
 - uart8250.c, [642](#)
- UART_DLM_OFFSET
 - uart8250.c, [642](#)
- UART_FCR_OFFSET
 - uart8250.c, [642](#)
- UART_IER_OFFSET
 - uart8250.c, [642](#)
- UART_IIR_OFFSET
 - uart8250.c, [643](#)
- UART_LCR_OFFSET
 - uart8250.c, [643](#)
- UART_LSR_BRK_ERROR_BITS
 - uart8250.c, [643](#)
- UART_LSR_BI
 - uart8250.c, [643](#)
- UART_LSR_DR
 - uart8250.c, [643](#)
- UART_LSR_FIFOE
 - uart8250.c, [643](#)
- UART_LSR_FE
 - uart8250.c, [643](#)
- UART_LSR_OFFSET
 - uart8250.c, [644](#)
- UART_LSR_OE
 - uart8250.c, [643](#)
- UART_LSR_PE
 - uart8250.c, [644](#)
- UART_LSR_TEMT
 - uart8250.c, [644](#)
- UART_LSR_THRE
 - uart8250.c, [644](#)
- UART_MCR_OFFSET
 - uart8250.c, [644](#)
- UART_MDR1_OFFSET
 - uart8250.c, [644](#)
- UART_MSR_OFFSET
 - uart8250.c, [644](#)
- UART_RBR_OFFSET
 - uart8250.c, [644](#)
- UART_REG_DIV
 - sifive-uart.c, [636](#)
- UART_REG_IE
 - sifive-uart.c, [636](#)
- UART_REG_IP
 - sifive-uart.c, [636](#)
- UART_REG_RXCTRL
 - sifive-uart.c, [636](#)
- UART_REG_RXFIFO
 - sifive-uart.c, [637](#)
- UART_REG_TXCTRL
 - sifive-uart.c, [637](#)
- UART_REG_TXFIFO
 - sifive-uart.c, [637](#)
- UART_RXCTRL_RXEN
 - sifive-uart.c, [637](#)
- UART_RXFIFO_DATA
 - sifive-uart.c, [637](#)
- UART_RXFIFO_EMPTY
 - sifive-uart.c, [637](#)
- UART_SCR_OFFSET
 - uart8250.c, [645](#)
- UART_THR_OFFSET
 - uart8250.c, [645](#)
- UART_TXCTRL_TXEN
 - sifive-uart.c, [637](#)
- UART_TXFIFO_FULL
 - sifive-uart.c, [637](#)
- UINT_MAX
 - libfdt_env.h, [624](#)
- ULL
 - sbi_const.h, [188](#)
- uart8250.c
 - get_reg, [645](#)
 - set_reg, [645](#)
 - UART_DLL_OFFSET, [642](#)
 - UART_DLM_OFFSET, [642](#)
 - UART_FCR_OFFSET, [642](#)
 - UART_IER_OFFSET, [642](#)
 - UART_IIR_OFFSET, [643](#)
 - UART_LCR_OFFSET, [643](#)
 - UART_LSR_BRK_ERROR_BITS, [643](#)
 - UART_LSR_BI, [643](#)
 - UART_LSR_DR, [643](#)
 - UART_LSR_FIFOE, [643](#)
 - UART_LSR_FE, [643](#)
 - UART_LSR_OFFSET, [644](#)
 - UART_LSR_OE, [643](#)
 - UART_LSR_PE, [644](#)
 - UART_LSR_TEMT, [644](#)
 - UART_LSR_THRE, [644](#)
 - UART_MCR_OFFSET, [644](#)
 - UART_MDR1_OFFSET, [644](#)
 - UART_MSR_OFFSET, [644](#)
 - UART_RBR_OFFSET, [644](#)
 - UART_SCR_OFFSET, [645](#)
 - UART_THR_OFFSET, [645](#)
 - uart8250_base, [647](#)
 - uart8250_baudrate, [647](#)
 - uart8250_getc, [646](#)
 - uart8250_in_freq, [647](#)
 - uart8250_init, [646](#)
 - uart8250_putc, [647](#)
 - uart8250_reg_shift, [648](#)
 - uart8250_reg_width, [648](#)
- uart8250.h
 - uart8250_getc, [318](#)
 - uart8250_init, [318](#)
 - uart8250_putc, [319](#)
- uart8250_base
 - uart8250.c, [647](#)
- uart8250_baudrate
 - uart8250.c, [647](#)
- uart8250_getc
 - uart8250.c, [646](#)

- uart8250.h, [318](#)
- uart8250_in_freq
 - uart8250.c, [647](#)
- uart8250_init
 - uart8250.c, [646](#)
 - uart8250.h, [318](#)
- uart8250_putc
 - uart8250.c, [647](#)
 - uart8250.h, [319](#)
- uart8250_reg_shift
 - uart8250.c, [648](#)
- uart8250_reg_width
 - uart8250.c, [648](#)
- uart_base
 - sifive-uart.c, [640](#)
- uart_baudrate
 - sifive-uart.c, [640](#)
- uart_in_freq
 - sifive-uart.c, [641](#)
- uart_min_clk_divisor
 - sifive-uart.c, [640](#)
- uint16_t
 - sbi_types.h, [307](#)
- uint32_t
 - sbi_types.h, [307](#)
- uint8_t
 - sbi_types.h, [308](#)
- uintptr_t
 - sbi_types.h, [308](#)
- UL
 - sbi_const.h, [188](#)
- ulong
 - sbi_types.h, [308](#)
- unlikely
 - sbi_types.h, [305](#)
- update
 - sbi_ipi_event_ops, [55](#)
- va_arg
 - sbi_console.c, [344](#)
- va_end
 - sbi_console.c, [344](#)
- va_list
 - sbi_console.c, [345](#)
- va_start
 - sbi_console.c, [344](#)
- vendor_ext_check
 - sbi_platform_operations, [61](#)
- vendor_ext_provider
 - sbi_platform_operations, [61](#)
- version
 - fdt_header, [47](#)
 - fw_dynamic_info, [51](#)
- virtual_addr_t
 - sbi_types.h, [308](#)
- virtual_size_t
 - sbi_types.h, [308](#)
- warmboot_addr
 - sbi_scratch, [63](#)
- wfi
 - riscv_asm.h, [81](#)
- wmb
 - riscv_barrier.h, [93](#)
- writeb
 - riscv_io.h, [169](#)
- writeb_relaxed
 - riscv_io.h, [169](#)
- writel
 - riscv_io.h, [169](#)
- writel_relaxed
 - riscv_io.h, [169](#)
- writeq
 - riscv_io.h, [169](#)
- writeq_relaxed
 - riscv_io.h, [170](#)
- writew
 - riscv_io.h, [170](#)
- writew_relaxed
 - riscv_io.h, [170](#)
- XSTR
 - sbi_types.h, [305](#)
- xchg
 - riscv_atomic.c, [333](#)
- zero
 - sbi_trap_regs, [71](#)