

Or1ksim: The OpenRISC 1000 Architectural Simulator

Generated by Doxygen 1.5.6

Tue Nov 11 11:50:04 2008

Contents

1	Orksim: the OpenRISC 1000 Architectural Simulator	1
1.1	About	1
1.2	Installation	1
1.3	Documentation	1
1.4	Copying	1
2	Todo List	3
3	Data Structure Index	5
3.1	Data Structures	5
4	File Index	9
4.1	File List	9
5	Data Structure Documentation	13
5.1	_csm_list Struct Reference	13
5.1.1	Field Documentation	14
5.1.1.1	ref	14
5.1.1.2	cnt	14
5.1.1.3	cmovs	14
5.1.1.4	size	14
5.1.1.5	osize	14
5.1.1.6	cmatch	14
5.1.1.7	dead	14
5.1.1.8	ninsn	14
5.1.1.9	from	14
5.1.1.10	next	14
5.2	_cuc_func Struct Reference	15
5.2.1	Field Documentation	16
5.2.1.1	num_bb	16

5.2.1.2	bb	16
5.2.1.3	saved_regs	16
5.2.1.4	lur	16
5.2.1.5	used_regs	16
5.2.1.6	nmsched	16
5.2.1.7	msched	16
5.2.1.8	mtype	16
5.2.1.9	num_init_bb	16
5.2.1.10	init_bb_reloc	16
5.2.1.11	orig_time	16
5.2.1.12	num_runs	16
5.2.1.13	timings	16
5.2.1.14	start_addr	16
5.2.1.15	end_addr	16
5.2.1.16	memory_order	16
5.2.1.17	nfdeps	16
5.2.1.18	fdeps	16
5.2.1.19	tmp	16
5.3	_dep_list_t Struct Reference	17
5.3.1	Field Documentation	17
5.3.1.1	ref	17
5.3.1.2	next	17
5.4	archf Struct Reference	18
5.4.1	Field Documentation	18
5.4.1.1	get_real_func_len	18
5.4.1.2	gen_reloc	18
5.4.1.3	gen_func_reloc	18
5.5	ata_device Struct Reference	19
5.5.1	Field Documentation	22
5.5.1.1	host	22
5.5.1.2	dev	22
5.5.1.3	pio_mode	22
5.5.1.4	dma_mode	22
5.5.1.5	dbuf	22
5.5.1.6	dbuf_ptr	22
5.5.1.7	dbuf_cnt	22

5.5.1.8	state	22
5.5.1.9	heads_per_cylinder	22
5.5.1.10	sectors_per_track	22
5.5.1.11	lba	22
5.5.1.12	nr_sect	22
5.5.1.13	end_t_func	22
5.5.1.14	internals	22
5.5.1.15	command	22
5.5.1.16	cylinder_low	22
5.5.1.17	cylinder_high	22
5.5.1.18	device_control	22
5.5.1.19	device_head	22
5.5.1.20	error	22
5.5.1.21	features	22
5.5.1.22	sector_count	22
5.5.1.23	sector_number	22
5.5.1.24	status	22
5.5.1.25	dataport_i	22
5.5.1.26	regs	22
5.5.1.27	iordy	22
5.5.1.28	intrq	22
5.5.1.29	dmarq	22
5.5.1.30	pdiagi	22
5.5.1.31	pdiago	22
5.5.1.32	daspi	22
5.5.1.33	daspo	22
5.5.1.34	sigs	22
5.5.1.35	file	22
5.5.1.36	stream	22
5.5.1.37	type	22
5.5.1.38	size	22
5.5.1.39	size_sect	22
5.5.1.40	packet	22
5.5.1.41	heads	22
5.5.1.42	sectors	22
5.5.1.43	firmware	22

5.5.1.44	mwdma	22
5.5.1.45	pio	22
5.5.1.46	conf	22
5.6	ata_devices Struct Reference	24
5.6.1	Field Documentation	24
5.6.1.1	device	24
5.7	ata_host Struct Reference	25
5.7.1	Field Documentation	26
5.7.1.1	enabled	26
5.7.1.2	baseaddr	26
5.7.1.3	mem	26
5.7.1.4	irq	26
5.7.1.5	dev_id	26
5.7.1.6	rev	26
5.7.1.7	dev_sel	26
5.7.1.8	pio_mode0_t1	26
5.7.1.9	pio_mode0_t2	26
5.7.1.10	pio_mode0_t4	26
5.7.1.11	pio_mode0_tecoc	26
5.7.1.12	dma_mode0_tm	26
5.7.1.13	dma_mode0_td	26
5.7.1.14	dma_mode0_tecoc	26
5.7.1.15	ctrl	26
5.7.1.16	stat	26
5.7.1.17	pctr	26
5.7.1.18	pftr0	26
5.7.1.19	pftr1	26
5.7.1.20	dtr0	26
5.7.1.21	dtr1	26
5.7.1.22	txb	26
5.7.1.23	rxb	26
5.7.1.24	regs	26
5.7.1.25	devices	26
5.8	bff Struct Reference	28
5.8.1	Field Documentation	28
5.8.1.1	open_obj	28

5.8.1.2	close_obj	28
5.8.1.3	get_func_name	28
5.8.1.4	get_func_start	28
5.8.1.5	get_func_len	28
5.8.1.6	get_func_reloc	28
5.9	BMP_HEADER Struct Reference	29
5.9.1	Field Documentation	29
5.9.1.1	type	29
5.9.1.2	size	29
5.9.1.3	reserved1	29
5.9.1.4	reserved2	29
5.9.1.5	offset	29
5.10	bpb_entry Struct Reference	30
5.10.1	Field Documentation	30
5.10.1.1	addr	30
5.10.1.2	taken	30
5.10.1.3	lru	30
5.10.1.4	way	30
5.11	bpbstat Struct Reference	31
5.11.1	Field Documentation	31
5.11.1.1	hit	31
5.11.1.2	miss	31
5.11.1.3	correct	31
5.11.1.4	incorrect	31
5.12	branchstat Struct Reference	32
5.12.1	Field Documentation	32
5.12.1.1	taken	32
5.12.1.2	nottaken	32
5.12.1.3	forward	32
5.12.1.4	backward	32
5.13	breakpoint_entry Struct Reference	33
5.13.1	Detailed Description	33
5.13.2	Field Documentation	33
5.13.2.1	addr	33
5.13.2.2	next	33
5.14	btic_entry Struct Reference	34

5.14.1	Field Documentation	34
5.14.1.1	addr	34
5.14.1.2	lru	34
5.14.1.3	insn	34
5.14.1.4	way	34
5.15	bticstat Struct Reference	35
5.15.1	Field Documentation	35
5.15.1.1	hit	35
5.15.1.2	miss	35
5.16	cachestats_entry Struct Reference	36
5.16.1	Field Documentation	36
5.16.1.1	readhit	36
5.16.1.2	readmiss	36
5.16.1.3	writehit	36
5.16.1.4	writemiss	36
5.17	channel Struct Reference	37
5.17.1	Detailed Description	37
5.17.2	Field Documentation	37
5.17.2.1	ops	37
5.17.2.2	data	37
5.18	channel_factory Struct Reference	38
5.18.1	Field Documentation	38
5.18.1.1	name	38
5.18.1.2	ops	38
5.18.1.3	next	38
5.19	channel_ops Struct Reference	39
5.19.1	Detailed Description	39
5.19.2	Field Documentation	39
5.19.2.1	init	39
5.19.2.2	open	39
5.19.2.3	close	39
5.19.2.4	read	39
5.19.2.5	write	39
5.19.2.6	free	39
5.19.2.7	isok	39
5.19.2.8	status	39

5.20 COFF_AOUTHDR Struct Reference	40
5.20.1 Field Documentation	40
5.20.1.1 magic	40
5.20.1.2 vstamp	40
5.20.1.3 tsize	40
5.20.1.4 dsize	40
5.20.1.5 bsize	40
5.20.1.6 entry	40
5.20.1.7 text_start	40
5.20.1.8 data_start	40
5.21 COFF_auxent Union Reference	41
5.21.1 Field Documentation	43
5.21.1.1 x_tagndx	43
5.21.1.2 x_lno	43
5.21.1.3 x_size	43
5.21.1.4 x_lnsz	43
5.21.1.5 x_fsize	43
5.21.1.6 x_misc	43
5.21.1.7 x_lnoptr	43
5.21.1.8 x_endndx	43
5.21.1.9 x_fcn	43
5.21.1.10 x_dimen	43
5.21.1.11 x_ary	43
5.21.1.12 x_fcary	43
5.21.1.13 x_tvndx	43
5.21.1.14 x_sym	43
5.21.1.15 x_fname	43
5.21.1.16 x_zeroes	43
5.21.1.17 x_offset	43
5.21.1.18 x_n	43
5.21.1.19 x_file	43
5.21.1.20 x_scrlen	43
5.21.1.21 x_nreloc	43
5.21.1.22 x_nlinno	43
5.21.1.23 x_scn	43
5.21.1.24 x_tvfill	43

5.21.1.25	x_tvlen	43
5.21.1.26	x_tvrn	43
5.21.1.27	x_tv	43
5.22	COFF_filehdr Struct Reference	45
5.22.1	Field Documentation	45
5.22.1.1	f_magic	45
5.22.1.2	f_nsens	45
5.22.1.3	f_timdat	45
5.22.1.4	f_symptr	45
5.22.1.5	f_nsyms	45
5.22.1.6	f_opthdr	45
5.22.1.7	f_flags	45
5.23	COFF_lineno Struct Reference	46
5.23.1	Field Documentation	46
5.23.1.1	l_symndx	46
5.23.1.2	l_paddr	46
5.23.1.3	l_addr	46
5.23.1.4	l_inno	46
5.24	COFF_reloc Struct Reference	47
5.24.1	Field Documentation	47
5.24.1.1	r_vaddr	47
5.24.1.2	r_symndx	47
5.24.1.3	r_type	47
5.25	COFF_scnhdr Struct Reference	48
5.25.1	Field Documentation	48
5.25.1.1	s_name	48
5.25.1.2	s_paddr	48
5.25.1.3	s_vaddr	48
5.25.1.4	s_size	48
5.25.1.5	s_scnptr	48
5.25.1.6	s_relptr	48
5.25.1.7	s_innoptr	48
5.25.1.8	s_nreloc	48
5.25.1.9	s_nlnno	48
5.25.1.10	s_flags	48
5.26	COFF_slb Struct Reference	49

5.26.1	Field Documentation	49
5.26.1.1	sl_entz	49
5.26.1.2	sl_pathndx	49
5.27	COFF_syment Struct Reference	50
5.27.1	Field Documentation	50
5.27.1.1	e_name	50
5.27.1.2	e_zeroes	50
5.27.1.3	e_offset	50
5.27.1.4	e	50
5.27.1.5	e	50
5.27.1.6	e_value	50
5.27.1.7	e_scnnum	50
5.27.1.8	e_type	50
5.27.1.9	e_sclass	50
5.27.1.10	e_numaux	50
5.28	config Struct Reference	51
5.28.1	Detailed Description	53
5.28.2	Field Documentation	55
5.28.2.1	class_ptr	55
5.28.2.2	read_up	55
5.28.2.3	write_up	55
5.28.2.4	ext	55
5.28.2.5	debug	55
5.28.2.6	verbose	55
5.28.2.7	profile	55
5.28.2.8	prof_fn	55
5.28.2.9	mprofile	55
5.28.2.10	mprof_fn	55
5.28.2.11	history	55
5.28.2.12	exe_log	55
5.28.2.13	exe_log_type	55
5.28.2.14	exe_log_start	55
5.28.2.15	exe_log_end	55
5.28.2.16	exe_log_marker	55
5.28.2.17	exe_log_fn	55
5.28.2.18	clkcycle_ps	55

5.28.2.19	sim	55
5.28.2.20	enabled	55
5.28.2.21	server_port	55
5.28.2.22	log_enabled	55
5.28.2.23	hide_device_id	55
5.28.2.24	vapi_fn	55
5.28.2.25	vapi	55
5.28.2.26	timings_fn	55
5.28.2.27	memory_order	55
5.28.2.28	calling_convention	55
5.28.2.29	enable_bursts	55
5.28.2.30	no_multicycle	55
5.28.2.31	cuc	55
5.28.2.32	superscalar	55
5.28.2.33	hazards	55
5.28.2.34	dependstats	55
5.28.2.35	sbuf_len	55
5.28.2.36	cpu	55
5.28.2.37	nways	55
5.28.2.38	nsets	55
5.28.2.39	blocksize	55
5.28.2.40	ustates	55
5.28.2.41	store_missdelay	55
5.28.2.42	store_hitdelay	55
5.28.2.43	load_missdelay	55
5.28.2.44	load_hitdelay	55
5.28.2.45	dc	55
5.28.2.46	pic	55
5.28.2.47	pm	55
5.28.2.48	sbp_bnf_fwd	55
5.28.2.49	sbp_bf_fwd	55
5.28.2.50	btic	55
5.28.2.51	missdelay	55
5.28.2.52	hitdelay	55
5.28.2.53	bpb	55
5.28.2.54	gdb_enabled	55

5.28.2.55	rsp_enabled	55
5.28.2.56	rsp_port	55
5.28.2.57	vapi_id	55
5.28.2.58	debug	55
5.29	config::pic Struct Reference	57
5.29.1	Field Documentation	57
5.29.1.1	enabled	57
5.29.1.2	edge_trigger	57
5.30	config_param Struct Reference	58
5.30.1	Field Documentation	58
5.30.1.1	name	58
5.30.1.2	type	58
5.30.1.3	func	58
5.30.1.4	next	58
5.31	config_section Struct Reference	59
5.31.1	Field Documentation	59
5.31.1.1	name	59
5.31.1.2	sec_start	59
5.31.1.3	sec_end	59
5.31.1.4	dat	59
5.31.1.5	params	59
5.31.1.6	next	59
5.32	cpu_state Struct Reference	60
5.32.1	Detailed Description	60
5.32.2	Field Documentation	60
5.32.2.1	reg	60
5.32.2.2	sprs	60
5.32.2.3	insn_ea	61
5.32.2.4	delay_insn	61
5.32.2.5	pc	61
5.32.2.6	pc_delay	61
5.32.2.7	pic_lines	61
5.32.2.8	iqueue	61
5.32.2.9	icomplet	61
5.33	cuc_bb Struct Reference	62
5.33.1	Field Documentation	63

5.33.1.1	type	63
5.33.1.2	first	63
5.33.1.3	last	63
5.33.1.4	prev	63
5.33.1.5	next	63
5.33.1.6	tmp	63
5.33.1.7	insn	63
5.33.1.8	ninsn	63
5.33.1.9	last_used_reg	63
5.33.1.10	mdep	63
5.33.1.11	nmemory	63
5.33.1.12	cnt	63
5.33.1.13	unrolled	63
5.33.1.14	ntim	63
5.33.1.15	tim	63
5.33.1.16	selected_tim	63
5.34	cuc_conv Struct Reference	64
5.34.1	Field Documentation	64
5.34.1.1	from	64
5.34.1.2	to	64
5.35	cuc_insn Struct Reference	65
5.35.1	Field Documentation	65
5.35.1.1	type	65
5.35.1.2	index	65
5.35.1.3	opt	65
5.35.1.4	op	65
5.35.1.5	dep	65
5.35.1.6	insn	65
5.35.1.7	disasm	65
5.35.1.8	max	65
5.35.1.9	tmp	65
5.36	cuc_known_insn Struct Reference	66
5.36.1	Field Documentation	66
5.36.1.1	name	66
5.36.1.2	comutative	66
5.36.1.3	rtl	66

5.37	cuc_shared_item Struct Reference	67
5.37.1	Field Documentation	67
5.37.1.1	ref	67
5.37.1.2	cmatch	67
5.38	cuc_timing_table Struct Reference	68
5.38.1	Field Documentation	68
5.38.1.1	delay	68
5.38.1.2	size	68
5.38.1.3	delayi	68
5.38.1.4	sizei	68
5.39	cuc_timings Struct Reference	69
5.39.1	Field Documentation	69
5.39.1.1	b	69
5.39.1.2	preroll	69
5.39.1.3	unroll	69
5.39.1.4	nshared	69
5.39.1.5	shared	69
5.39.1.6	new_time	69
5.39.1.7	size	69
5.40	dc_set Struct Reference	70
5.40.1	Field Documentation	70
5.40.1.1	line	70
5.40.1.2	tagaddr	70
5.40.1.3	lru	70
5.40.1.4	way	70
5.41	dev_16450 Struct Reference	71
5.41.1	Field Documentation	74
5.41.1.1	txbuf	74
5.41.1.2	rxbuf	74
5.41.1.3	dll	74
5.41.1.4	dlh	74
5.41.1.5	ier	74
5.41.1.6	iir	74
5.41.1.7	fcr	74
5.41.1.8	lcr	74
5.41.1.9	mcr	74

5.41.1.10	lsr	74
5.41.1.11	msr	74
5.41.1.12	scr	74
5.41.1.13	regs	74
5.41.1.14	txser	74
5.41.1.15	rxser	74
5.41.1.16	loopback	74
5.41.1.17	iregs	74
5.41.1.18	txbuf_head	74
5.41.1.19	txbuf_tail	74
5.41.1.20	rxbuf_head	74
5.41.1.21	rxbuf_tail	74
5.41.1.22	txbuf_full	74
5.41.1.23	rxbuf_full	74
5.41.1.24	receiveing	74
5.41.1.25	recv_break	74
5.41.1.26	ints	74
5.41.1.27	istat	74
5.41.1.28	char_clks	74
5.41.1.29	skew	74
5.41.1.30	vapi	74
5.41.1.31	vapi_buf	74
5.41.1.32	vapi_buf_head_ptr	74
5.41.1.33	vapi_buf_tail_ptr	74
5.41.1.34	fifo_len	74
5.41.1.35	channel	74
5.41.1.36	enabled	74
5.41.1.37	jitter	74
5.41.1.38	baseaddr	74
5.41.1.39	irq	74
5.41.1.40	vapi_id	74
5.41.1.41	uart16550	74
5.41.1.42	channel_str	74
5.42	dev_generic Struct Reference	76
5.42.1	Detailed Description	76
5.42.2	Member Enumeration Documentation	76

5.42.2.1	"@49	76
5.42.2.2	"@50	76
5.42.3	Field Documentation	77
5.42.3.1	trans_direction	77
5.42.3.2	trans_size	77
5.42.3.3	value	77
5.42.3.4	enabled	77
5.42.3.5	byte_enabled	77
5.42.3.6	hw_enabled	77
5.42.3.7	word_enabled	77
5.42.3.8	name	77
5.42.3.9	baseaddr	77
5.42.3.10	size	77
5.43	dev_memarea Struct Reference	78
5.43.1	Detailed Description	78
5.43.2	Field Documentation	79
5.43.2.1	next	79
5.43.2.2	addr_mask	79
5.43.2.3	addr_compare	79
5.43.2.4	size	79
5.43.2.5	size_mask	79
5.43.2.6	valid	79
5.43.2.7	log	79
5.43.2.8	ops	79
5.43.2.9	direct_ops	79
5.44	dma_channel Struct Reference	80
5.44.1	Field Documentation	81
5.44.1.1	controller	81
5.44.1.2	channel_number	81
5.44.1.3	channel_mask	81
5.44.1.4	referenced	81
5.44.1.5	load_next_descriptor_when_done	81
5.44.1.6	current_descriptor	81
5.44.1.7	source	81
5.44.1.8	destination	81
5.44.1.9	source_mask	81

5.44.1.10	destination_mask	81
5.44.1.11	chunk_size	81
5.44.1.12	total_size	81
5.44.1.13	words_transferred	81
5.44.1.14	csr	81
5.44.1.15	sz	81
5.44.1.16	a0	81
5.44.1.17	am0	81
5.44.1.18	a1	81
5.44.1.19	aml	81
5.44.1.20	desc	81
5.44.1.21	swptr	81
5.44.1.22	regs	81
5.44.1.23	dma_req_i	81
5.44.1.24	dma_ack_o	81
5.44.1.25	dma_nd_i	81
5.45	dma_controller Struct Reference	83
5.45.1	Field Documentation	84
5.45.1.1	enabled	84
5.45.1.2	baseaddr	84
5.45.1.3	irq	84
5.45.1.4	vapi_id	84
5.45.1.5	csr	84
5.45.1.6	int_msk_a	84
5.45.1.7	int_msk_b	84
5.45.1.8	int_src_a	84
5.45.1.9	int_src_b	84
5.45.1.10	regs	84
5.45.1.11	ch	84
5.45.1.12	next	84
5.46	dmmu Struct Reference	85
5.46.1	Field Documentation	86
5.46.1.1	enabled	86
5.46.1.2	nways	86
5.46.1.3	nsets	86
5.46.1.4	pagesize	86

5.46.1.5	pagesize_log2	86
5.46.1.6	page_offset_mask	86
5.46.1.7	page_mask	86
5.46.1.8	vpn_mask	86
5.46.1.9	lru_reload	86
5.46.1.10	set_mask	86
5.46.1.11	entrysize	86
5.46.1.12	ustates	86
5.46.1.13	missdelay	86
5.46.1.14	hitdelay	86
5.47	dmmustats_entry Struct Reference	87
5.47.1	Field Documentation	87
5.47.1.1	loads_tlbhit	87
5.47.1.2	loads_tlbmiss	87
5.47.1.3	loads_pagefaults	87
5.47.1.4	stores_tlbhit	87
5.47.1.5	stores_tlbmiss	87
5.47.1.6	stores_pagefaults	87
5.48	dstats_entry Struct Reference	88
5.48.1	Field Documentation	88
5.48.1.1	insn1	88
5.48.1.2	insn2	88
5.48.1.3	cnt_dynamic	88
5.48.1.4	depend	88
5.49	dyn_page Struct Reference	89
5.49.1	Field Documentation	89
5.49.1.1	or_page	89
5.49.1.2	host_page	89
5.49.1.3	host_len	89
5.49.1.4	dirty	89
5.49.1.5	delayr	89
5.49.1.6	ts_bound	89
5.49.1.7	locs	89
5.49.1.8	insns	89
5.49.1.9	insn_indexs	89
5.50	dynamic Struct Reference	90

5.50.1	Field Documentation	90
5.50.1.1	d_tag	90
5.50.1.2	d_val	90
5.50.1.3	d_ptr	90
5.50.1.4	d_un	90
5.51	elf32_hdr Struct Reference	91
5.51.1	Field Documentation	92
5.51.1.1	e_ident	92
5.51.1.2	e_type	92
5.51.1.3	e_machine	92
5.51.1.4	e_version	92
5.51.1.5	e_entry	92
5.51.1.6	e_phoff	92
5.51.1.7	e_shoff	92
5.51.1.8	e_flags	92
5.51.1.9	e_ehsize	92
5.51.1.10	e_phentsize	92
5.51.1.11	e_phnum	92
5.51.1.12	e_shentsize	92
5.51.1.13	e_shnum	92
5.51.1.14	e_shstrndx	92
5.52	elf32_note Struct Reference	93
5.52.1	Field Documentation	93
5.52.1.1	n_namesz	93
5.52.1.2	n_descsz	93
5.52.1.3	n_type	93
5.53	elf32_phdr Struct Reference	94
5.53.1	Field Documentation	94
5.53.1.1	p_type	94
5.53.1.2	p_offset	94
5.53.1.3	p_vaddr	94
5.53.1.4	p_paddr	94
5.53.1.5	p_filesz	94
5.53.1.6	p_memsz	94
5.53.1.7	p_flags	94
5.53.1.8	p_align	94

5.54	elf32_rel Struct Reference	95
5.54.1	Field Documentation	95
5.54.1.1	r_offset	95
5.54.1.2	r_info	95
5.55	elf32_rela Struct Reference	96
5.55.1	Field Documentation	96
5.55.1.1	r_offset	96
5.55.1.2	r_info	96
5.55.1.3	r_addend	96
5.56	elf32_shdr Struct Reference	97
5.56.1	Field Documentation	97
5.56.1.1	sh_name	97
5.56.1.2	sh_type	97
5.56.1.3	sh_flags	97
5.56.1.4	sh_addr	97
5.56.1.5	sh_offset	97
5.56.1.6	sh_size	97
5.56.1.7	sh_link	97
5.56.1.8	sh_info	97
5.56.1.9	sh_addralign	97
5.56.1.10	sh_entsize	97
5.57	elf32_sym Struct Reference	98
5.57.1	Field Documentation	98
5.57.1.1	st_name	98
5.57.1.2	st_value	98
5.57.1.3	st_size	98
5.57.1.4	st_info	98
5.57.1.5	st_other	98
5.57.1.6	st_shndx	98
5.58	Elf64_Dyn Struct Reference	99
5.58.1	Field Documentation	99
5.58.1.1	d_tag	99
5.58.1.2	d_val	99
5.58.1.3	d_ptr	99
5.58.1.4	d_un	99
5.59	elf64_hdr Struct Reference	100

5.59.1	Field Documentation	101
5.59.1.1	e_ident	101
5.59.1.2	e_type	101
5.59.1.3	e_machine	101
5.59.1.4	e_version	101
5.59.1.5	e_entry	101
5.59.1.6	e_phoff	101
5.59.1.7	e_shoff	101
5.59.1.8	e_flags	101
5.59.1.9	e_ehsize	101
5.59.1.10	e_phentsize	101
5.59.1.11	e_phnum	101
5.59.1.12	e_shentsize	101
5.59.1.13	e_shnum	101
5.59.1.14	e_shstrndx	101
5.60	elf64_note Struct Reference	102
5.60.1	Field Documentation	102
5.60.1.1	n_namesz	102
5.60.1.2	n_descsz	102
5.60.1.3	n_type	102
5.61	elf64_phdr Struct Reference	103
5.61.1	Field Documentation	103
5.61.1.1	p_type	103
5.61.1.2	p_flags	103
5.61.1.3	p_offset	103
5.61.1.4	p_vaddr	103
5.61.1.5	p_paddr	103
5.61.1.6	p_filesz	103
5.61.1.7	p_memsz	103
5.61.1.8	p_align	103
5.62	elf64_rel Struct Reference	104
5.62.1	Field Documentation	104
5.62.1.1	r_offset	104
5.62.1.2	r_info	104
5.63	elf64_rela Struct Reference	105
5.63.1	Field Documentation	105

5.63.1.1	r_offset	105
5.63.1.2	r_info	105
5.63.1.3	r_addend	105
5.64	elf64_shdr Struct Reference	106
5.64.1	Field Documentation	106
5.64.1.1	sh_name	106
5.64.1.2	sh_type	106
5.64.1.3	sh_flags	106
5.64.1.4	sh_addr	106
5.64.1.5	sh_offset	106
5.64.1.6	sh_size	106
5.64.1.7	sh_link	106
5.64.1.8	sh_info	106
5.64.1.9	sh_addralign	106
5.64.1.10	sh_entsize	106
5.65	elf64_sym Struct Reference	107
5.65.1	Field Documentation	107
5.65.1.1	st_name	107
5.65.1.2	st_info	107
5.65.1.3	st_other	107
5.65.1.4	st_shndx	107
5.65.1.5	st_value	107
5.65.1.6	st_size	107
5.66	elf_obj Struct Reference	108
5.66.1	Field Documentation	109
5.66.1.1	e_hdr	109
5.66.1.2	e_shdrs	109
5.66.1.3	e_sections	109
5.66.1.4	e_syms	109
5.66.1.5	e_sym_num	109
5.66.1.6	e_sym_str_tab	109
5.66.1.7	e_rels	109
5.66.1.8	e_rel_num	109
5.66.1.9	e_rel_sym	109
5.66.1.10	e_rel_sec	109
5.66.1.11	e_relas	109

5.66.1.12 e_rela_num	109
5.66.1.13 e_rela_sym	109
5.66.1.14 e_rela_sec	109
5.67 eth_device Struct Reference	110
5.67.1 Field Documentation	113
5.67.1.1 enabled	113
5.67.1.2 baseaddr	113
5.67.1.3 dma	113
5.67.1.4 tx_channel	113
5.67.1.5 rx_channel	113
5.67.1.6 mac_address	113
5.67.1.7 mac_int	113
5.67.1.8 base_vapi_id	113
5.67.1.9 rxfile	113
5.67.1.10 txfile	113
5.67.1.11 txfd	113
5.67.1.12 rxfd	113
5.67.1.13 loopback_offset	113
5.67.1.14 sockif	113
5.67.1.15 rtx_sock	113
5.67.1.16 rtx_type	113
5.67.1.17 ifr	113
5.67.1.18 rfd	113
5.67.1.19 wfd	113
5.67.1.20 state	113
5.67.1.21 bd_index	113
5.67.1.22 bd	113
5.67.1.23 bd_addr	113
5.67.1.24 working	113
5.67.1.25 waiting_for_dma	113
5.67.1.26 error	113
5.67.1.27 packet_length	113
5.67.1.28 minimum_length	113
5.67.1.29 maximum_length	113
5.67.1.30 add_crc	113
5.67.1.31 crc_dly	113

5.67.1.32	crc_value	113
5.67.1.33	bytes_left	113
5.67.1.34	bytes_sent	113
5.67.1.35	tx	113
5.67.1.36	fd	113
5.67.1.37	offset	113
5.67.1.38	bytes_read	113
5.67.1.39	rx	113
5.67.1.40	moder	113
5.67.1.41	int_source	113
5.67.1.42	int_mask	113
5.67.1.43	ipgt	113
5.67.1.44	ipgr1	113
5.67.1.45	ipgr2	113
5.67.1.46	packetlen	113
5.67.1.47	collconf	113
5.67.1.48	tx_bd_num	113
5.67.1.49	controlmoder	113
5.67.1.50	miimoder	113
5.67.1.51	miicommand	113
5.67.1.52	miiaddress	113
5.67.1.53	miitx_data	113
5.67.1.54	miirx_data	113
5.67.1.55	miistatus	113
5.67.1.56	hash0	113
5.67.1.57	hash1	113
5.67.1.58	bd_ram	113
5.67.1.59	regs	113
5.67.1.60	rx_buff	113
5.67.1.61	tx_buff	113
5.67.1.62	lo_buff	113
5.68	ether_addr Struct Reference	115
5.68.1	Field Documentation	115
5.68.1.1	ether_addr_octet	115
5.69	ether_header Struct Reference	116
5.69.1	Field Documentation	116

5.69.1.1	ether_dhost	116
5.69.1.2	ether_shost	116
5.69.1.3	ether_type	116
5.70	fb_state Struct Reference	117
5.70.1	Field Documentation	118
5.70.1.1	enabled	118
5.70.1.2	pal	118
5.70.1.3	ctrl	118
5.70.1.4	pic	118
5.70.1.5	in_refresh	118
5.70.1.6	refresh_count	118
5.70.1.7	addr	118
5.70.1.8	cam_addr	118
5.70.1.9	camerax	118
5.70.1.10	cameray	118
5.70.1.11	camera_pos	118
5.70.1.12	baseaddr	118
5.70.1.13	refresh	118
5.70.1.14	refresh_rate	118
5.70.1.15	filename	118
5.71	fd_channel Struct Reference	119
5.71.1	Detailed Description	119
5.71.2	Field Documentation	119
5.71.2.1	fdin	119
5.71.2.2	fdout	119
5.72	file_channel Struct Reference	120
5.72.1	Detailed Description	120
5.72.2	Field Documentation	120
5.72.2.1	fds	120
5.72.2.2	namein	120
5.72.2.3	nameout	120
5.73	fstats_entry Struct Reference	121
5.73.1	Field Documentation	121
5.73.1.1	insn1	121
5.73.1.2	insn2	121
5.73.1.3	cnt_dynamic	121

5.73.1.4	depend	121
5.74	func_struct Struct Reference	122
5.74.1	Detailed Description	122
5.74.2	Field Documentation	122
5.74.2.1	addr	122
5.74.2.2	name	122
5.74.2.3	cum_cycles	122
5.74.2.4	calls	122
5.75	gpio_device Struct Reference	123
5.75.1	Field Documentation	124
5.75.1.1	enabled	124
5.75.1.2	baseaddr	124
5.75.1.3	irq	124
5.75.1.4	gpio_number	124
5.75.1.5	base_vapi_id	124
5.75.1.6	auxiliary_inputs	124
5.75.1.7	in	124
5.75.1.8	out	124
5.75.1.9	oe	124
5.75.1.10	inte	124
5.75.1.11	ptrig	124
5.75.1.12	aux	124
5.75.1.13	ctrl	124
5.75.1.14	ints	124
5.75.1.15	external_clock	124
5.75.1.16	curr	124
5.75.1.17	next	124
5.76	hist_exec Struct Reference	125
5.76.1	Detailed Description	125
5.76.2	Field Documentation	125
5.76.2.1	addr	125
5.76.2.2	prev	125
5.76.2.3	next	125
5.77	ic Struct Reference	126
5.77.1	Field Documentation	127
5.77.1.1	mem	127

5.77.1.2	lrus	127
5.77.1.3	tags	127
5.77.1.4	enabled	127
5.77.1.5	nways	127
5.77.1.6	nsets	127
5.77.1.7	blocksize	127
5.77.1.8	ustates	127
5.77.1.9	missdelay	127
5.77.1.10	hitdelay	127
5.77.1.11	blocksize_log2	127
5.77.1.12	set_mask	127
5.77.1.13	tagaddr_mask	127
5.77.1.14	last_way	127
5.77.1.15	block_offset_mask	127
5.77.1.16	block_mask	127
5.77.1.17	ustates_reload	127
5.78	immu Struct Reference	128
5.78.1	Field Documentation	129
5.78.1.1	enabled	129
5.78.1.2	nways	129
5.78.1.3	nsets	129
5.78.1.4	pagesize	129
5.78.1.5	pagesize_log2	129
5.78.1.6	page_offset_mask	129
5.78.1.7	page_mask	129
5.78.1.8	vpn_mask	129
5.78.1.9	lru_reload	129
5.78.1.10	set_mask	129
5.78.1.11	entrysize	129
5.78.1.12	ustates	129
5.78.1.13	missdelay	129
5.78.1.14	hitdelay	129
5.79	immustats_entry Struct Reference	130
5.79.1	Field Documentation	130
5.79.1.1	fetch_tlbhit	130
5.79.1.2	fetch_tlbmiss	130

5.79.1.3	fetch_pagefaults	130
5.80	INFOHEADER Struct Reference	131
5.80.1	Field Documentation	131
5.80.1.1	size	131
5.80.1.2	width	131
5.80.1.3	height	131
5.80.1.4	planes	131
5.80.1.5	bits	131
5.80.1.6	compression	131
5.80.1.7	imagesize	131
5.80.1.8	xresolution	131
5.80.1.9	yresolution	131
5.80.1.10	ncolours	131
5.80.1.11	importantcolours	131
5.81	iqueue_entry Struct Reference	132
5.81.1	Detailed Description	132
5.81.2	Field Documentation	132
5.81.2.1	insn_index	132
5.81.2.2	insn	132
5.81.2.3	insn_addr	132
5.82	jtr_chain_message Struct Reference	133
5.82.1	Field Documentation	133
5.82.1.1	command	133
5.82.1.2	length	133
5.82.1.3	chain	133
5.83	jtr_chain_response Struct Reference	134
5.83.1	Field Documentation	134
5.83.1.1	status	134
5.84	jtr_failure_response Struct Reference	135
5.84.1	Field Documentation	135
5.84.1.1	status	135
5.85	jtr_read_block_message Struct Reference	136
5.85.1	Field Documentation	136
5.85.1.1	command	136
5.85.1.2	length	136
5.85.1.3	address	136

5.85.1.4	num_regs	136
5.86	jtr_read_block_response Struct Reference	137
5.86.1	Field Documentation	137
5.86.1.1	status	137
5.86.1.2	num_regs	137
5.86.1.3	data	137
5.87	jtr_read_message Struct Reference	138
5.87.1	Field Documentation	138
5.87.1.1	command	138
5.87.1.2	length	138
5.87.1.3	address	138
5.88	jtr_read_response Struct Reference	139
5.88.1	Field Documentation	139
5.88.1.1	status	139
5.88.1.2	data_h	139
5.88.1.3	data_l	139
5.89	jtr_write_block_message Struct Reference	140
5.89.1	Field Documentation	140
5.89.1.1	command	140
5.89.1.2	length	140
5.89.1.3	address	140
5.89.1.4	num_regs	140
5.89.1.5	data	140
5.90	jtr_write_block_response Struct Reference	141
5.90.1	Field Documentation	141
5.90.1.1	status	141
5.91	jtr_write_message Struct Reference	142
5.91.1	Field Documentation	142
5.91.1.1	command	142
5.91.1.2	length	142
5.91.1.3	address	142
5.91.1.4	data_h	142
5.91.1.5	data_l	142
5.92	jtr_write_response Struct Reference	143
5.92.1	Field Documentation	143
5.92.1.1	status	143

5.93	kbd_state Struct Reference	144
5.93.1	Field Documentation	144
5.93.1.1	buf	144
5.93.1.2	buf_count	144
5.93.1.3	buf_head	144
5.93.1.4	buf_tail	144
5.93.1.5	rxfs	144
5.93.1.6	ccmd	144
5.93.1.7	kcmd	144
5.93.1.8	ccmdbyte	144
5.93.1.9	kresp	144
5.93.1.10	slowdown	144
5.93.1.11	enabled	144
5.93.1.12	irq	144
5.93.1.13	baseaddr	144
5.93.1.14	rxfile	144
5.94	label_entry Struct Reference	145
5.94.1	Detailed Description	145
5.94.2	Field Documentation	145
5.94.2.1	name	145
5.94.2.2	addr	145
5.94.2.3	next	145
5.95	mc Struct Reference	146
5.95.1	Field Documentation	147
5.95.1.1	csr	147
5.95.1.2	poc	147
5.95.1.3	ba_mask	147
5.95.1.4	csc	147
5.95.1.5	tms	147
5.95.1.6	baseaddr	147
5.95.1.7	enabled	147
5.95.1.8	index	147
5.95.1.9	mc_areas	147
5.95.1.10	next	147
5.96	mc_area Struct Reference	148
5.96.1	Field Documentation	148

5.96.1.1	mem	148
5.96.1.2	cs	148
5.96.1.3	mc	148
5.96.1.4	next	148
5.97	mem_config Struct Reference	149
5.97.1	Member Enumeration Documentation	149
5.97.1.1	"@53	149
5.97.2	Field Documentation	150
5.97.2.1	ce	150
5.97.2.2	mc	150
5.97.2.3	baseaddr	150
5.97.2.4	size	150
5.97.2.5	name	150
5.97.2.6	log	150
5.97.2.7	delayr	150
5.97.2.8	delayw	150
5.97.2.9	mem	150
5.97.2.10	pattern	150
5.97.2.11	random_seed	150
5.97.2.12	type	150
5.98	mem_ops Struct Reference	151
5.98.1	Detailed Description	151
5.98.2	Field Documentation	152
5.98.2.1	readfunc32	152
5.98.2.2	readfunc16	152
5.98.2.3	readfunc8	152
5.98.2.4	read_dat8	152
5.98.2.5	read_dat16	152
5.98.2.6	read_dat32	152
5.98.2.7	writefunc32	152
5.98.2.8	writefunc16	152
5.98.2.9	writefunc8	152
5.98.2.10	write_dat8	152
5.98.2.11	write_dat16	152
5.98.2.12	write_dat32	152
5.98.2.13	writeprog32	152

5.98.2.14	writeprog8	152
5.98.2.15	writeprog32_dat	152
5.98.2.16	writeprog8_dat	152
5.98.2.17	delayr	152
5.98.2.18	delayw	152
5.98.2.19	log	152
5.99	memory_hash Struct Reference	153
5.99.1	Detailed Description	153
5.99.2	Field Documentation	153
5.99.2.1	next	153
5.99.2.2	addr	153
5.99.2.3	cnt	153
5.100	mp_entry Struct Reference	154
5.100.1	Detailed Description	154
5.100.2	Field Documentation	154
5.100.2.1	type	154
5.100.2.2	addr	154
5.100.2.3	instr	154
5.100.2.4	next	154
5.101	mprofentry_struct Struct Reference	155
5.101.1	Field Documentation	155
5.101.1.1	addr	155
5.101.1.2	type	155
5.102	mstats_entry Struct Reference	156
5.102.1	Field Documentation	156
5.102.1.1	byteadd	156
5.102.1.2	bf	156
5.102.1.3	bnf	156
5.102.1.4	bpb	156
5.102.1.5	btic	156
5.103	op_queue Struct Reference	157
5.103.1	Field Documentation	158
5.103.1.1	num_ops	158
5.103.1.2	ops_len	158
5.103.1.3	ops	158
5.103.1.4	num_ops_param	158

5.103.1.5 ops_param_len	158
5.103.1.6 ops_param	158
5.103.1.7 jump_local	158
5.103.1.8 jump_local_loc	158
5.103.1.9 not_jump_loc	158
5.103.1.10 xref	158
5.103.1.11 linsn_addr	158
5.103.1.12 reg_t	158
5.103.1.13 flags	158
5.103.1.14 linsn_index	158
5.103.1.15 param_type	158
5.103.1.16 param	158
5.103.1.17 param_num	158
5.103.1.18 linsn	158
5.103.1.19 prev	158
5.103.1.20 next	158
5.104 param_val Union Reference	159
5.104.1 Detailed Description	159
5.104.2 Field Documentation	159
5.104.2.1 str_val	159
5.104.2.2 int_val	159
5.104.2.3 longlong_val	159
5.104.2.4 addr_val	159
5.105 raw_stats Struct Reference	160
5.105.1 Field Documentation	160
5.105.1.1 reg	160
5.105.1.2 range	160
5.106 reloc Struct Reference	161
5.106.1 Field Documentation	161
5.106.1.1 func_offset	161
5.106.1.2 addend	161
5.106.1.3 type	161
5.106.1.4 name	161
5.107 rsp_buf Struct Reference	162
5.107.1 Detailed Description	162
5.107.2 Field Documentation	162

5.107.2.1 data	162
5.107.2.2 len	162
5.108 runtime Struct Reference	163
5.108.1 Detailed Description	163
5.108.2 Field Documentation	165
5.108.2.1 fprof	165
5.108.2.2 fmprof	165
5.108.2.3 fexe_log	165
5.108.2.4 fout	165
5.108.2.5 filename	165
5.108.2.6 iprompt	165
5.108.2.7 iprompt_run	165
5.108.2.8 cycles	165
5.108.2.9 end_cycles	165
5.108.2.10 time_point	165
5.108.2.11 ext_int	165
5.108.2.12 mem_cycles	165
5.108.2.13 loadcycles	165
5.108.2.14 storecycles	165
5.108.2.15 reset_cycles	165
5.108.2.16 hush	165
5.108.2.17 sim	165
5.108.2.18 instructions	165
5.108.2.19 reset_instructions	165
5.108.2.20 stalled	165
5.108.2.21 hazardwait	165
5.108.2.22 supercycles	165
5.108.2.23 cpu	165
5.108.2.24 enabled	165
5.108.2.25 vapi_file	165
5.108.2.26 server_port	165
5.108.2.27 vapi	165
5.108.2.28 ndelay	165
5.108.2.29 cycle_duration	165
5.108.2.30 uc	165
5.109 sched_entry Struct Reference	167

5.109.1 Detailed Description	167
5.109.2 Field Documentation	167
5.109.2.1 time	167
5.109.2.2 param	167
5.109.2.3 func	167
5.109.2.4 next	167
5.110 scheduler_struct Struct Reference	168
5.110.1 Detailed Description	168
5.110.2 Field Documentation	168
5.110.2.1 free_job_queue	168
5.110.2.2 job_queue	168
5.111 sim_command Struct Reference	169
5.111.1 Field Documentation	169
5.111.1.1 name	169
5.111.1.2 cmd_handle	169
5.112 sim_reset_hook Struct Reference	170
5.112.1 Detailed Description	170
5.112.2 Field Documentation	170
5.112.2.1 dat	170
5.112.2.2 reset_hook	170
5.112.2.3 next	170
5.113 sim_stat Struct Reference	171
5.113.1 Field Documentation	171
5.113.1.1 stat_func	171
5.113.1.2 dat	171
5.113.1.3 next	171
5.114 spr_bit_def Struct Reference	172
5.114.1 Field Documentation	172
5.114.1.1 name	172
5.114.1.2 mask	172
5.115 spr_def Struct Reference	173
5.115.1 Field Documentation	173
5.115.1.1 from_spr	173
5.115.1.2 to_spr	173
5.115.1.3 name	173
5.115.1.4 bits	173

5.116	stats_entry Struct Reference	174
5.116.1	Field Documentation	174
5.116.1.1	insn	174
5.116.1.2	cnt_dynamic	174
5.117	stack_struct Struct Reference	175
5.117.1	Detailed Description	175
5.117.2	Field Documentation	175
5.117.2.1	addr	175
5.117.2.2	cycles	175
5.117.2.3	raddr	175
5.117.2.4	name	175
5.118	tcp_channel Struct Reference	176
5.118.1	Detailed Description	176
5.118.2	Field Documentation	176
5.118.2.1	fds	176
5.118.2.2	socket_fd	176
5.118.2.3	port_number	176
5.118.2.4	connected	176
5.118.2.5	nonblocking	176
5.119	tty_channel Struct Reference	177
5.119.1	Detailed Description	177
5.119.2	Field Documentation	177
5.119.2.1	fds	177
5.120	vapi_handler Struct Reference	178
5.120.1	Field Documentation	178
5.120.1.1	fd	178
5.120.1.2	base_id	178
5.120.1.3	num_ids	178
5.120.1.4	read_func	178
5.120.1.5	priv_dat	178
5.120.1.6	next	178
5.120.1.7	temp	178
5.121	vga_state Struct Reference	179
5.121.1	Field Documentation	180
5.121.1.1	enabled	180
5.121.1.2	pics	180

5.121.1.3	ctrl	180
5.121.1.4	stat	180
5.121.1.5	htim	180
5.121.1.6	vtim	180
5.121.1.7	vbindex	180
5.121.1.8	vbar	180
5.121.1.9	hlen	180
5.121.1.10	vlen	180
5.121.1.11	lpindex	180
5.121.1.12	palette	180
5.121.1.13	baseaddr	180
5.121.1.14	refresh_rate	180
5.121.1.15	rq	180
5.121.1.16	filename	180
5.122	xterm_channel Struct Reference	181
5.122.1	Detailed Description	181
5.122.2	Field Documentation	181
5.122.2.1	fds	181
5.122.2.2	pid	181
5.122.2.3	argv	181
6	File Documentation	183
6.1	bpb/branch-predict.c File Reference	183
6.1.1	Define Documentation	186
6.1.1.1	BPB_LEN	186
6.1.1.2	BPB_PSTATES	186
6.1.1.3	BPB_USTATES	186
6.1.1.4	BPB_WAYS	186
6.1.1.5	BTIC_BLOCKSIZE	186
6.1.1.6	BTIC_LEN	186
6.1.1.7	BTIC_USTATES	186
6.1.1.8	BTIC_WAYS	186
6.1.2	Function Documentation	186
6.1.2.1	bpb_btic	186
6.1.2.2	bpb_enabled	186
6.1.2.3	bpb_hitdelay	186
6.1.2.4	bpb_info	186

6.1.2.5	bpb_missdelay	186
6.1.2.6	bpb_sbp_bf_fwd	186
6.1.2.7	bpb_sbp_bnf_fwd	186
6.1.2.8	bpb_update	186
6.1.2.9	btic_info	186
6.1.2.10	btic_update	186
6.1.2.11	reg_bpb_sec	186
6.1.3	Variable Documentation	187
6.1.3.1	bpb	187
6.1.3.2	btic	187
6.2	bpb/branch-predict.h File Reference	188
6.2.1	Function Documentation	189
6.2.1.1	bpb_info	189
6.2.1.2	bpb_update	189
6.2.1.3	btic_info	189
6.2.1.4	btic_update	189
6.2.1.5	reg_bpb_sec	189
6.3	cache/dcache-model.c File Reference	190
6.3.1	Function Documentation	191
6.3.1.1	dc_blocksize	191
6.3.1.2	dc_enabled	191
6.3.1.3	dc_info	191
6.3.1.4	dc_inv	191
6.3.1.5	dc_load_hitdelay	191
6.3.1.6	dc_load_missdelay	191
6.3.1.7	dc_nsets	191
6.3.1.8	dc_nways	192
6.3.1.9	dc_simulate_read	192
6.3.1.10	dc_simulate_write	193
6.3.1.11	dc_store_hitdelay	193
6.3.1.12	dc_store_missdelay	193
6.3.1.13	dc_ustates	193
6.3.1.14	reg_dc_sec	194
6.3.2	Variable Documentation	194
6.3.2.1	dc	194
6.4	cache/dcache-model.h File Reference	195

6.4.1	Define Documentation	196
6.4.1.1	MAX_DC_BLOCK_SIZE	196
6.4.1.2	MAX_DC_SETS	196
6.4.1.3	MAX_DC_WAYS	196
6.4.1.4	MIN_DC_BLOCK_SIZE	196
6.4.2	Function Documentation	196
6.4.2.1	dc_info	196
6.4.2.2	dc_inv	196
6.4.2.3	dc_simulate_read	196
6.4.2.4	dc_simulate_write	197
6.4.2.5	reg_dc_sec	197
6.5	cache/icache-model.c File Reference	198
6.5.1	Define Documentation	199
6.5.1.1	MAX_IC_BLOCK_SIZE	199
6.5.1.2	MAX_IC_SETS	199
6.5.1.3	MAX_IC_WAYS	199
6.5.1.4	MIN_IC_BLOCK_SIZE	199
6.5.2	Function Documentation	199
6.5.2.1	ic_blocksize	199
6.5.2.2	ic_enabled	199
6.5.2.3	ic_end_sec	200
6.5.2.4	ic_hitdelay	200
6.5.2.5	ic_info	200
6.5.2.6	ic_inv	200
6.5.2.7	ic_missdelay	200
6.5.2.8	ic_nsets	200
6.5.2.9	ic_nways	200
6.5.2.10	ic_simulate_fetch	201
6.5.2.11	ic_start_sec	201
6.5.2.12	ic_ustates	201
6.5.2.13	reg_ic_sec	202
6.5.3	Variable Documentation	202
6.5.3.1	ic_state	202
6.6	cache/icache-model.h File Reference	203
6.6.1	Function Documentation	204
6.6.1.1	ic_inv	204

6.6.1.2	ic_simulate_fetch	204
6.6.1.3	reg_ic_sec	204
6.6.2	Variable Documentation	205
6.6.2.1	ic_state	205
6.7	cpu-config.c File Reference	206
6.7.1	Define Documentation	207
6.7.1.1	WARNING	207
6.7.2	Function Documentation	207
6.7.2.1	cpu_cfg	207
6.7.2.2	cpu_cfgr	207
6.7.2.3	cpu_dependstats	207
6.7.2.4	cpu_hazards	207
6.7.2.5	cpu_rev	207
6.7.2.6	cpu_sbuf_len	207
6.7.2.7	cpu_sr	207
6.7.2.8	cpu_superscalar	208
6.7.2.9	cpu_upr	208
6.7.2.10	cpu_ver	208
6.7.2.11	reg_cpu_sec	208
6.8	cpu-config.h File Reference	210
6.8.1	Function Documentation	210
6.8.1.1	reg_cpu_sec	210
6.9	cpu/common/abstract.c File Reference	211
6.9.1	Function Documentation	213
6.9.1.1	adjust_rw_delay	213
6.9.1.2	bit_mask	213
6.9.1.3	disassemble_memory	213
6.9.1.4	done_memory_table	213
6.9.1.5	dump_memory	213
6.9.1.6	eval_direct16	214
6.9.1.7	eval_direct32	215
6.9.1.8	eval_direct8	215
6.9.1.9	eval_insn	215
6.9.1.10	eval_mem16	216
6.9.1.11	eval_mem32	216
6.9.1.12	eval_mem8	217

6.9.1.13	eval_mem_16_inv	217
6.9.1.14	eval_mem_16_inv_direct	218
6.9.1.15	eval_mem_32_inv	218
6.9.1.16	eval_mem_32_inv_direct	218
6.9.1.17	eval_mem_8_inv	218
6.9.1.18	eval_mem_8_inv_direct	219
6.9.1.19	evalsim_mem16	219
6.9.1.20	evalsim_mem32	219
6.9.1.21	evalsim_mem8	220
6.9.1.22	generate_time_pretty	220
6.9.1.23	memory_table_status	220
6.9.1.24	reg_mem_area	221
6.9.1.25	register_memoryarea_mask	221
6.9.1.26	set_direct16	222
6.9.1.27	set_direct32	222
6.9.1.28	set_direct8	223
6.9.1.29	set_mem16	223
6.9.1.30	set_mem32	224
6.9.1.31	set_mem8	224
6.9.1.32	set_mem_16_inv	225
6.9.1.33	set_mem_16_inv_direct	225
6.9.1.34	set_mem_32_inv	225
6.9.1.35	set_mem_32_inv_direct	226
6.9.1.36	set_mem_8_inv	226
6.9.1.37	set_mem_8_inv_direct	226
6.9.1.38	set_mem_valid	226
6.9.1.39	set_program32	226
6.9.1.40	set_program8	226
6.9.1.41	setsim_mem16	227
6.9.1.42	setsim_mem32	227
6.9.1.43	setsim_mem8	227
6.9.1.44	verify_memoryarea	228
6.9.2	Variable Documentation	228
6.9.2.1	cur_area	228
6.9.2.2	cur_vadd	228
6.9.2.3	data_ci	228

6.9.2.4	dev_list	228
6.9.2.5	insn_ci	228
6.9.2.6	mc_area	228
6.10	cpu/common/abstract.h File Reference	229
6.10.1	Define Documentation	231
6.10.1.1	CT_NONE	231
6.10.1.2	CT_PHYSICAL	231
6.10.1.3	CT_VIRTUAL	231
6.10.1.4	DEFAULT_MEMORY_LEN	231
6.10.1.5	DEFAULT_MEMORY_START	231
6.10.1.6	HISTEXEC_LEN	231
6.10.1.7	INSNAME_LEN	231
6.10.1.8	LABELNAME_LEN	231
6.10.1.9	LE16	231
6.10.1.10	LONGEST	231
6.10.1.11	MAX_OPERANDS	231
6.10.1.12	OP_MEM_ACCESS	231
6.10.1.13	OPERANDNAME_LEN	231
6.10.1.14	STACK_SIZE	231
6.10.1.15	ULONGEST	231
6.10.2	Function Documentation	231
6.10.2.1	adjust_rw_delay	231
6.10.2.2	disassemble_memory	231
6.10.2.3	done_memory_table	232
6.10.2.4	dump_memory	232
6.10.2.5	eval_direct16	233
6.10.2.6	eval_direct32	233
6.10.2.7	eval_direct8	234
6.10.2.8	eval_insn	234
6.10.2.9	eval_mem16	235
6.10.2.10	eval_mem32	235
6.10.2.11	eval_mem8	236
6.10.2.12	evalsim_mem16	236
6.10.2.13	evalsim_mem32	237
6.10.2.14	evalsim_mem8	237
6.10.2.15	generate_time_pretty	237

6.10.2.16	memory_table_status	237
6.10.2.17	reg_mem_area	238
6.10.2.18	set_direct16	238
6.10.2.19	set_direct32	239
6.10.2.20	set_direct8	239
6.10.2.21	set_mem16	240
6.10.2.22	set_mem32	240
6.10.2.23	set_mem8	241
6.10.2.24	set_mem_valid	241
6.10.2.25	set_program32	241
6.10.2.26	set_program8	241
6.10.2.27	setsim_mem16	242
6.10.2.28	setsim_mem32	242
6.10.2.29	setsim_mem8	242
6.10.2.30	verify_memoryarea	243
6.10.3	Variable Documentation	243
6.10.3.1	cur_area	243
6.10.3.2	data_ci	243
6.10.3.3	hist_exec_tail	243
6.10.3.4	insn_ci	243
6.11	cpu/common/coff.h File Reference	244
6.11.1	Define Documentation	249
6.11.1.1	C_ALIAS	249
6.11.1.2	C_ARG	249
6.11.1.3	C_AUTO	249
6.11.1.4	C_AUTOARG	249
6.11.1.5	C_BCOMM	249
6.11.1.6	C_BINCL	249
6.11.1.7	C_BLOCK	249
6.11.1.8	C_BSTAT	249
6.11.1.9	C_DECL	249
6.11.1.10	C_DEFINE	249
6.11.1.11	C_ECOML	249
6.11.1.12	C_ECOMM	249
6.11.1.13	C_EFCN	249
6.11.1.14	C_EINCL	249

6.11.1.15 C_ENTAG	249
6.11.1.16 C_ENTRY	249
6.11.1.17 C_EOS	249
6.11.1.18 C_ESTAT	249
6.11.1.19 C_EXT	249
6.11.1.20 C_EXTDEF	249
6.11.1.21 C_EXTLAB	249
6.11.1.22 C_FCN	249
6.11.1.23 C_FIELD	249
6.11.1.24 C_FILE	249
6.11.1.25 C_FUN	249
6.11.1.26 C_GSYM	249
6.11.1.27 C_HIDDEN	249
6.11.1.28 C_HIDEXT	249
6.11.1.29 C_LABEL	249
6.11.1.30 C_LASTENT	249
6.11.1.31 C_LEAFEXT	249
6.11.1.32 C_LEAFPROC	249
6.11.1.33 C_LEAFSTAT	249
6.11.1.34 C_LINE	249
6.11.1.35 C_LSYM	249
6.11.1.36 C_MOE	249
6.11.1.37 C_MOS	249
6.11.1.38 C_MOU	249
6.11.1.39 C_NT_WEAK	249
6.11.1.40 C_NULL	249
6.11.1.41 C_OPTVAR	249
6.11.1.42 C_PRAGMA	249
6.11.1.43 C_PSYM	249
6.11.1.44 C_REG	249
6.11.1.45 C_REGPARM	249
6.11.1.46 C_RPSYM	249
6.11.1.47 C_RSYM	249
6.11.1.48 C_SCALL	249
6.11.1.49 C_SECTION	249
6.11.1.50 C_SEGMENT	249

6.11.1.51 C_SHADOW	249
6.11.1.52 C_STAT	249
6.11.1.53 C_STATLAB	249
6.11.1.54 C_STRTAG	249
6.11.1.55 C_STSYM	249
6.11.1.56 C_SYSTEM	249
6.11.1.57 C_TCSYM	249
6.11.1.58 C_THUMBEXT	249
6.11.1.59 C_THUMBEXTFUNC	249
6.11.1.60 C_THUMBLABEL	249
6.11.1.61 C_THUMBSTAT	249
6.11.1.62 C_THUMBSTATFUNC	249
6.11.1.63 C_TPDEF	249
6.11.1.64 C_UEXT	249
6.11.1.65 C_ULABEL	249
6.11.1.66 C_UNTAG	249
6.11.1.67 C_USTATIC	249
6.11.1.68 C_VERSION	249
6.11.1.69 C_WEAKEXT	249
6.11.1.70 COFF_AOUTSZ	249
6.11.1.71 COFF_AUXENT	249
6.11.1.72 COFF_AUXESZ	249
6.11.1.73 COFF_BSS	249
6.11.1.74 COFF_COMMENT	249
6.11.1.75 COFF_DATA	249
6.11.1.76 COFF_DEF_BSS_SECTION_ALIGNMENT	249
6.11.1.77 COFF_DEF_DATA_SECTION_ALIGNMENT	249
6.11.1.78 COFF_DEF_SECTION_ALIGNMENT	249
6.11.1.79 COFF_DEF_TEXT_SECTION_ALIGNMENT	249
6.11.1.80 COFF_DMAGIC	249
6.11.1.81 COFF_E_DIMNUM	249
6.11.1.82 COFF_E_FILNMLEN	249
6.11.1.83 COFF_E_SYMNMLEN	249
6.11.1.84 COFF_ETEXT	249
6.11.1.85 COFF_F_AR16WR	249
6.11.1.86 COFF_F_AR32W	249

6.11.1.87	COFF_F_AR32WR	249
6.11.1.88	COFF_F_EXEC	249
6.11.1.89	COFF_F_LNNO	249
6.11.1.90	COFF_F_LSYMS	249
6.11.1.91	COFF_F_MINMAL	249
6.11.1.92	COFF_F_NODF	249
6.11.1.93	COFF_F_PATCH	249
6.11.1.94	COFF_F_RELFLG	249
6.11.1.95	COFF_F_SWABD	249
6.11.1.96	COFF_F_UPDATE	249
6.11.1.97	COFF_FILHDR	249
6.11.1.98	COFF_FILHSZ	249
6.11.1.99	COFF_I386BADMAG	249
6.11.1.100	COFF_I386MAGIC	249
6.11.1.101	COFF_JMAGIC	249
6.11.1.102	COFF_LIB	249
6.11.1.103	COFF_LINENO	249
6.11.1.104	COFF_LINESZ	249
6.11.1.105	COFF_LONG	249
6.11.1.106	COFF_LONG_H	249
6.11.1.107	COFF_LONG_L	249
6.11.1.108	COFF_N_BTMASK	249
6.11.1.109	COFF_N_BTSHFT	249
6.11.1.110	COFF_N_TMASK	249
6.11.1.111	COFF_N_TSHIFT	249
6.11.1.112	COFF_OMAGIC	249
6.11.1.113	COFF_RELOC	249
6.11.1.114	COFF_RELSZ	249
6.11.1.115	COFF_SCNHDR	249
6.11.1.116	COFF_SCNHSZ	249
6.11.1.117	COFF_SECT_BSS	249
6.11.1.118	COFF_SECT_DATA	249
6.11.1.119	COFF_SECT_REQD	249
6.11.1.120	COFF_SECT_TEXT	249
6.11.1.121	COFF_SHMAGIC	249
6.11.1.122	COFF_SHORT	249

6.11.1.123	COFF_SHORT_H	249
6.11.1.124	COFF_SHORT_L	249
6.11.1.125	COFF_SLIBHD	249
6.11.1.126	COFF_SLIBSZ	249
6.11.1.127	COFF_STMAGIC	249
6.11.1.128	COFF_STYP_BSS	249
6.11.1.129	COFF_STYP_COPY	249
6.11.1.130	COFF_STYP_DATA	249
6.11.1.131	COFF_STYP_DSECT	249
6.11.1.132	COFF_STYP_GROUP	249
6.11.1.133	COFF_STYP_INFO	249
6.11.1.134	COFF_STYP_LIB	249
6.11.1.135	COFF_STYP_NOLOAD	249
6.11.1.136	COFF_STYP_OVER	249
6.11.1.137	COFF_STYP_PAD	249
6.11.1.138	COFF_STYP_REG	249
6.11.1.139	COFF_STYP_TEXT	249
6.11.1.140	COFF_SYMENT	249
6.11.1.141	COFF_SYMESZ	249
6.11.1.142	COFF_TEXT	249
6.11.1.143	COFF_ZMAGIC	249
6.11.1.144	E_DIMNUM	249
6.11.1.145	E_FILNMLEN	249
6.11.1.146	E_SYMNMLEN	249
6.11.1.147	KEEP_ENDIAN_LONG	249
6.11.1.148	KEEP_ENDIAN_SHORT	250
6.11.1.149	SWAP_ENDIAN_LONG	250
6.11.1.150	SWAP_ENDIAN_SHORT	250
6.12	cpu/common/elf.h File Reference	251
6.12.1	Define Documentation	257
6.12.1.1	AT_BASE	257
6.12.1.2	AT_EGID	257
6.12.1.3	AT_ENTRY	257
6.12.1.4	AT_EUID	257
6.12.1.5	AT_EXECFD	257
6.12.1.6	AT_FLAGS	257

6.12.1.7	AT_GID	257
6.12.1.8	AT_IGNORE	257
6.12.1.9	AT_NOTELF	257
6.12.1.10	AT_NULL	257
6.12.1.11	AT_PAGESZ	257
6.12.1.12	AT_PHDR	257
6.12.1.13	AT_PHENT	257
6.12.1.14	AT_PHNUM	257
6.12.1.15	AT_UID	257
6.12.1.16	DT_DEBUG	257
6.12.1.17	DT_FINI	257
6.12.1.18	DT_HASH	257
6.12.1.19	DT_HIPROC	257
6.12.1.20	DT_INIT	257
6.12.1.21	DT_JMPREL	257
6.12.1.22	DT_LOPROC	257
6.12.1.23	DT_NEEDED	257
6.12.1.24	DT_NULL	257
6.12.1.25	DT_PLTGOT	257
6.12.1.26	DT_PLTREL	257
6.12.1.27	DT_PLTRELSZ	257
6.12.1.28	DT_REL	257
6.12.1.29	DT_RELA	257
6.12.1.30	DT_RELAENT	257
6.12.1.31	DT_RELASZ	257
6.12.1.32	DT_RELENT	257
6.12.1.33	DT_RELSZ	257
6.12.1.34	DT_RPATH	257
6.12.1.35	DT_SONAME	257
6.12.1.36	DT_STRSZ	257
6.12.1.37	DT_STRTAB	257
6.12.1.38	DT_SYMBOLIC	257
6.12.1.39	DT_SYMENT	257
6.12.1.40	DT_SYMTAB	257
6.12.1.41	DT_TEXTREL	257
6.12.1.42	EI_CLASS	257

6.12.1.43	EL_DATA	257
6.12.1.44	EL_MAG0	257
6.12.1.45	EL_MAG1	257
6.12.1.46	EL_MAG2	257
6.12.1.47	EL_MAG3	257
6.12.1.48	EL_NIDENT	257
6.12.1.49	EL_PAD	257
6.12.1.50	EL_VERSION	257
6.12.1.51	ELF32_R_SYM	257
6.12.1.52	ELF32_R_TYPE	257
6.12.1.53	ELF32_ST_BIND	257
6.12.1.54	ELF32_ST_TYPE	257
6.12.1.55	ELF_LONG_H	257
6.12.1.56	elf_note	258
6.12.1.57	elf_phdr	258
6.12.1.58	ELF_SHORT_H	258
6.12.1.59	ELFCLASS32	260
6.12.1.60	ELFCLASS64	260
6.12.1.61	ELFCLASSNONE	260
6.12.1.62	ELFCLASSNUM	260
6.12.1.63	ELFDATA2LSB	260
6.12.1.64	ELFDATA2MSB	260
6.12.1.65	ELFDATANONE	260
6.12.1.66	elfhdr	260
6.12.1.67	ELFMAG	260
6.12.1.68	ELFMAG0	260
6.12.1.69	ELFMAG1	260
6.12.1.70	ELFMAG2	260
6.12.1.71	ELFMAG3	260
6.12.1.72	EM_386	260
6.12.1.73	EM_486	260
6.12.1.74	EM_68K	260
6.12.1.75	EM_860	260
6.12.1.76	EM_88K	260
6.12.1.77	EM_ALPHA	260
6.12.1.78	EM_M32	260

6.12.1.79 EM_MIPS	260
6.12.1.80 EM_MIPS_RS4_BE	260
6.12.1.81 EM_NONE	260
6.12.1.82 EM_PARISC	260
6.12.1.83 EM_PPC	260
6.12.1.84 EM_SPARC	260
6.12.1.85 EM_SPARC32PLUS	260
6.12.1.86 EM_SPARC64	260
6.12.1.87 ET_CORE	260
6.12.1.88 ET_DYN	260
6.12.1.89 ET_EXEC	260
6.12.1.90 ET_HIPROC	260
6.12.1.91 ET_LOPROC	260
6.12.1.92 ET_NONE	260
6.12.1.93 ET_REL	260
6.12.1.94 EV_CURRENT	260
6.12.1.95 EV_NONE	260
6.12.1.96 EV_NUM	260
6.12.1.97 NT_PRFPREG	260
6.12.1.98 NT_PRPSINFO	260
6.12.1.99 NT_PRSTATUS	260
6.12.1.100 NT_TASKSTRUCT	260
6.12.1.101 PF_R	260
6.12.1.102 PF_W	260
6.12.1.103 PF_X	260
6.12.1.104 PT_DYNAMIC	260
6.12.1.105 PT_HIPROC	260
6.12.1.106 PT_INTERP	260
6.12.1.107 PT_LOAD	260
6.12.1.108 PT_LOPROC	260
6.12.1.109 PT_NOTE	260
6.12.1.110 PT_NULL	260
6.12.1.111 PT_PHDR	260
6.12.1.112 PT_SHLIB	260
6.12.1.113 R_386_32	260
6.12.1.114 R_386_COPY	260

6.12.1.115R_386_GLOB_DAT	260
6.12.1.116R_386_GOT32	260
6.12.1.117R_386_GOTOFF	260
6.12.1.118R_386_GOTPC	260
6.12.1.119R_386_JMP_SLOT	260
6.12.1.120R_386_NONE	260
6.12.1.121R_386_NUM	260
6.12.1.122R_386_PC32	260
6.12.1.123R_386_PLT32	260
6.12.1.124R_386_RELATIVE	260
6.12.1.125R_68K_16	260
6.12.1.126R_68K_32	260
6.12.1.127R_68K_8	260
6.12.1.128R_68K_COPY	260
6.12.1.129R_68K_GLOB_DAT	260
6.12.1.130R_68K_GOT16	260
6.12.1.131R_68K_GOT16O	260
6.12.1.132R_68K_GOT32	260
6.12.1.133R_68K_GOT32O	260
6.12.1.134R_68K_GOT8	260
6.12.1.135R_68K_GOT8O	260
6.12.1.136R_68K_JMP_SLOT	260
6.12.1.137R_68K_NONE	260
6.12.1.138R_68K_PC16	260
6.12.1.139R_68K_PC32	260
6.12.1.140R_68K_PC8	260
6.12.1.141R_68K_PLT16	260
6.12.1.142R_68K_PLT16O	260
6.12.1.143R_68K_PLT32	260
6.12.1.144R_68K_PLT32O	260
6.12.1.145R_68K_PLT8	260
6.12.1.146R_68K_PLT8O	260
6.12.1.147R_68K_RELATIVE	260
6.12.1.148SELMAG	260
6.12.1.149SHF_ALLOC	260
6.12.1.150SHF_EXECINSTR	260

6.12.1.151SHF_MASKPROC	260
6.12.1.152SHF_WRITE	260
6.12.1.153SHN_ABS	260
6.12.1.154SHN_COMMON	260
6.12.1.155SHN_HIPROC	260
6.12.1.156SHN_HIRESERVE	260
6.12.1.157SHN_LOPROC	260
6.12.1.158SHN_LORESERVE	260
6.12.1.159SHN_UNDEF	260
6.12.1.160SHT_DYNAMIC	260
6.12.1.161SHT_DYNSYM	260
6.12.1.162SHT_HASH	260
6.12.1.163SHT_HIPROC	260
6.12.1.164SHT_HIUSER	260
6.12.1.165SHT_LOPROC	260
6.12.1.166SHT_LOUSER	260
6.12.1.167SHT_NOBITS	260
6.12.1.168SHT_NOTE	260
6.12.1.169SHT_NULL	260
6.12.1.170SHT_NUM	260
6.12.1.171SHT_PROGBITS	260
6.12.1.172SHT_REL	260
6.12.1.173SHT_RELA	260
6.12.1.174SHT_SHLIB	260
6.12.1.175SHT_STRTAB	260
6.12.1.176SHT_SYMTAB	260
6.12.1.177STB_GLOBAL	260
6.12.1.178STB_LOCAL	260
6.12.1.179STB_WEAK	260
6.12.1.180STT_FILE	260
6.12.1.181STT_FUNC	260
6.12.1.182STT_NOTYPE	260
6.12.1.183STT_OBJECT	260
6.12.1.184STT_SECTION	260
6.12.2 Typedef Documentation	260
6.12.2.1 Elf32_Addr	260

6.12.2.2	Elf32_Dyn	260
6.12.2.3	Elf32_Ehdr	260
6.12.2.4	Elf32_Half	260
6.12.2.5	Elf32_Nhdr	260
6.12.2.6	Elf32_Off	260
6.12.2.7	Elf32_Phdr	260
6.12.2.8	Elf32_Rel	260
6.12.2.9	Elf32_Rela	260
6.12.2.10	Elf32_Shdr	260
6.12.2.11	Elf32_Sword	260
6.12.2.12	Elf32_Sym	260
6.12.2.13	Elf32_Word	260
6.12.2.14	Elf64_Ehdr	260
6.12.2.15	Elf64_Nhdr	260
6.12.2.16	Elf64_Phdr	260
6.12.2.17	Elf64_Rel	260
6.12.2.18	Elf64_Rela	260
6.12.2.19	Elf64_Shdr	260
6.12.2.20	Elf64_Sym	260
6.12.3	Variable Documentation	260
6.12.3.1	_DYNAMIC	260
6.13	cpu/common/execute.h File Reference	261
6.13.1	Define Documentation	262
6.13.1.1	CURINSN	262
6.13.1.2	SET_OV_FLAG_FN	262
6.13.2	Function Documentation	262
6.13.2.1	analysis	262
6.13.2.2	cpu_clock	263
6.13.2.3	cpu_reset	263
6.13.2.4	depend_operands	263
6.13.2.5	dump_exe_log	263
6.13.2.6	dumpreg	264
6.13.2.7	eval_operand_val	264
6.13.2.8	evalsim_reg	265
6.13.2.9	exec_main	265
6.13.2.10	setsim_reg	267

6.13.3	Variable Documentation	267
6.13.3.1	cpu_state	267
6.13.3.2	do_stats	267
6.13.3.3	hist_exec_tail	267
6.13.3.4	pcnext	267
6.13.3.5	sbuf_total_cyc	267
6.13.3.6	sbuf_wait_cyc	267
6.14	cpu/common/labels.c File Reference	268
6.14.1	Define Documentation	269
6.14.1.1	LABELS_HASH_SIZE	269
6.14.2	Function Documentation	269
6.14.2.1	add_breakpoint	269
6.14.2.2	add_label	269
6.14.2.3	eval_label	269
6.14.2.4	find_label	269
6.14.2.5	get_label	269
6.14.2.6	has_breakpoint	269
6.14.2.7	init_breakpoints	269
6.14.2.8	init_labels	269
6.14.2.9	print_breakpoints	269
6.14.2.10	remove_breakpoint	269
6.14.3	Variable Documentation	269
6.14.3.1	breakpoints	269
6.14.3.2	label_hash	269
6.15	cpu/common/labels.h File Reference	270
6.15.1	Function Documentation	271
6.15.1.1	add_breakpoint	271
6.15.1.2	add_label	271
6.15.1.3	eval_label	271
6.15.1.4	find_label	271
6.15.1.5	get_label	271
6.15.1.6	has_breakpoint	271
6.15.1.7	init_breakpoints	271
6.15.1.8	init_labels	271
6.15.1.9	print_breakpoints	271
6.15.1.10	remove_breakpoint	271

6.15.2	Variable Documentation	271
6.15.2.1	breakpoints	271
6.16	cpu/common/parse.c File Reference	272
6.16.1	Define Documentation	273
6.16.1.1	IMM_STATS	273
6.16.1.2	MEMORY_LEN	273
6.16.2	Function Documentation	273
6.16.2.1	addprogram	273
6.16.2.2	identifyfile	274
6.16.2.3	loadcode	274
6.16.2.4	readfile_coff	275
6.16.2.5	readfile_elf	276
6.16.2.6	readsyms_coff	276
6.16.2.7	strup	276
6.16.2.8	translate	276
6.16.3	Variable Documentation	277
6.16.3.1	freemem	277
6.16.3.2	transl_error	277
6.16.3.3	transl_table	277
6.17	cpu/common/parse.h File Reference	278
6.17.1	Function Documentation	278
6.17.1.1	loadcode	278
6.17.1.2	strup	279
6.18	cpu/common/stats.c File Reference	280
6.18.1	Define Documentation	281
6.18.1.1	DSTATS_LEN	281
6.18.1.2	FSTATS_LEN	281
6.18.1.3	SD	281
6.18.1.4	SSTATS_LEN	281
6.18.2	Function Documentation	281
6.18.2.1	addstats	281
6.18.2.2	addfstats	281
6.18.2.3	addsstats	281
6.18.2.4	initstats	281
6.18.2.5	printotherstats	281
6.18.2.6	printstats	281

6.18.3	Variable Documentation	282
6.18.3.1	dc_stats	282
6.18.3.2	dmmu_stats	282
6.18.3.3	dstats	282
6.18.3.4	fstats	282
6.18.3.5	func_unit_str	282
6.18.3.6	ic_stats	282
6.18.3.7	immu_stats	282
6.18.3.8	or1k_mstats	283
6.18.3.9	raw_stats	283
6.18.3.10	sstats	283
6.19	cpu/common/stats.h File Reference	284
6.19.1	Define Documentation	285
6.19.1.1	RAW_RANGE	285
6.19.2	Function Documentation	285
6.19.2.1	addstats	285
6.19.2.2	addfstats	285
6.19.2.3	addsstats	285
6.19.2.4	initstats	285
6.19.2.5	printstats	285
6.19.3	Variable Documentation	285
6.19.3.1	dc_stats	285
6.19.3.2	dmmu_stats	285
6.19.3.3	ic_stats	285
6.19.3.4	immu_stats	285
6.19.3.5	or1k_mstats	285
6.19.3.6	raw_stats	285
6.20	cpu/common/trace.c File Reference	286
6.20.1	Function Documentation	286
6.20.1.1	set_insnbrkpoint	286
6.21	cpu/common/trace.h File Reference	288
6.21.1	Function Documentation	288
6.21.1.1	set_insnbrkpoint	288
6.22	cpu/or1k/arch.h File Reference	290
6.22.1	Define Documentation	290
6.22.1.1	ADDR_C	290

6.22.1.2	LINK_REGNO	290
6.22.1.3	PRIdREG	290
6.22.1.4	PRIxADDR	291
6.22.1.5	PRIxREG	291
6.22.1.6	REG_C	291
6.22.2	Typedef Documentation	291
6.22.2.1	oraddr_t	291
6.22.2.2	orreg_t	291
6.22.2.3	uorreg_t	291
6.23	cpu/or1k/except.c File Reference	292
6.23.1	Function Documentation	293
6.23.1.1	except_handle	293
6.23.1.2	op_join_mem_cycles	293
6.23.2	Variable Documentation	293
6.23.2.1	except_pending	293
6.24	cpu/or1k/except.h File Reference	294
6.24.1	Define Documentation	296
6.24.1.1	EXCEPT_ALIGN	296
6.24.1.2	EXCEPT_BUSERR	296
6.24.1.3	EXCEPT_DPF	296
6.24.1.4	EXCEPT_DTLBMISS	296
6.24.1.5	EXCEPT_FPE	296
6.24.1.6	EXCEPT_ILLEGAL	296
6.24.1.7	EXCEPT_INT	296
6.24.1.8	EXCEPT_IPF	296
6.24.1.9	EXCEPT_ITLBMIS	296
6.24.1.10	EXCEPT_NONE	296
6.24.1.11	EXCEPT_RANGE	296
6.24.1.12	EXCEPT_RESET	296
6.24.1.13	EXCEPT_SYSCALL	296
6.24.1.14	EXCEPT_TICK	296
6.24.1.15	EXCEPT_TRAP	296
6.24.2	Function Documentation	296
6.24.2.1	except_handle	296
6.24.3	Variable Documentation	297
6.24.3.1	except_pending	297

6.25	cpu/or1k/spr-defs.h File Reference	298
6.25.1	Define Documentation	305
6.25.1.1	MATCHPOINTS_TO_NDP	305
6.25.1.2	MAX_GRPS	307
6.25.1.3	MAX_MATCHPOINTS	307
6.25.1.4	MAX_SPRS	307
6.25.1.5	MAX_SPRS_PER_GRP	307
6.25.1.6	MAX_SPRS_PER_GRP_BITS	307
6.25.1.7	MAX_WATCHPOINTS	307
6.25.1.8	NOP_CNT_RESET	307
6.25.1.9	NOP_EXIT	307
6.25.1.10	NOP_NOP	307
6.25.1.11	NOP_PRINTF	307
6.25.1.12	NOP_PUTC	307
6.25.1.13	NOP_REPORT	307
6.25.1.14	NOP_REPORT_FIRST	307
6.25.1.15	NOP_REPORT_LAST	307
6.25.1.16	SPR_CPUCFGR	307
6.25.1.17	SPR_CPUCFGR_CGF	307
6.25.1.18	SPR_CPUCFGR_NSGF	307
6.25.1.19	SPR_CPUCFGR_OB32S	307
6.25.1.20	SPR_CPUCFGR_OB64S	307
6.25.1.21	SPR_CPUCFGR_OF32S	307
6.25.1.22	SPR_CPUCFGR_OF64S	307
6.25.1.23	SPR_CPUCFGR_OV64S	307
6.25.1.24	SPR_CPUCFGR_RES	307
6.25.1.25	SPR_DCBFR	307
6.25.1.26	SPR_DCBIR	307
6.25.1.27	SPR_DCBLR	307
6.25.1.28	SPR_DCBPR	307
6.25.1.29	SPR_DCBWR	307
6.25.1.30	SPR_DCCFGR	307
6.25.1.31	SPR_DCCFGR_CBFRI	307
6.25.1.32	SPR_DCCFGR_CBIRI	307
6.25.1.33	SPR_DCCFGR_CBLRI	307
6.25.1.34	SPR_DCCFGR_CBPRI	307

6.25.1.35 SPR_DCCFGR_CBS	307
6.25.1.36 SPR_DCCFGR_CBS_OFF	307
6.25.1.37 SPR_DCCFGR_CBWBRI	307
6.25.1.38 SPR_DCCFGR_CCRI	307
6.25.1.39 SPR_DCCFGR_CWS	307
6.25.1.40 SPR_DCCFGR_NCS	307
6.25.1.41 SPR_DCCFGR_NCS_OFF	307
6.25.1.42 SPR_DCCFGR_NCW	307
6.25.1.43 SPR_DCCFGR_NCW_OFF	307
6.25.1.44 SPR_DCCR	307
6.25.1.45 SPR_DCCR_EW	307
6.25.1.46 SPR_DCFGR	307
6.25.1.47 SPR_DCFGR_NDP	307
6.25.1.48 SPR_DCFGR_NDP1	307
6.25.1.49 SPR_DCFGR_NDP2	307
6.25.1.50 SPR_DCFGR_NDP3	307
6.25.1.51 SPR_DCFGR_NDP4	307
6.25.1.52 SPR_DCFGR_NDP5	307
6.25.1.53 SPR_DCFGR_NDP6	307
6.25.1.54 SPR_DCFGR_NDP7	307
6.25.1.55 SPR_DCFGR_NDP8	307
6.25.1.56 SPR_DCFGR_WPCI	307
6.25.1.57 SPR_DCR	307
6.25.1.58 SPR_DCR_BASE	307
6.25.1.59 SPR_DCR_CC	307
6.25.1.60 SPR_DCR_CC_EQUAL	307
6.25.1.61 SPR_DCR_CC_GREAT	307
6.25.1.62 SPR_DCR_CC_GREATE	307
6.25.1.63 SPR_DCR_CC_LESS	307
6.25.1.64 SPR_DCR_CC_LESSE	307
6.25.1.65 SPR_DCR_CC_MASKED	307
6.25.1.66 SPR_DCR_CC_NEQUAL	307
6.25.1.67 SPR_DCR_CT	307
6.25.1.68 SPR_DCR_CT_DISABLED	307
6.25.1.69 SPR_DCR_CT_IFEA	307
6.25.1.70 SPR_DCR_CT_LD	307

6.25.1.71	SPR_DCR_CT_LEA	307
6.25.1.72	SPR_DCR_CT_LSD	307
6.25.1.73	SPR_DCR_CT_LSEA	307
6.25.1.74	SPR_DCR_CT_SD	307
6.25.1.75	SPR_DCR_CT_SEA	307
6.25.1.76	SPR_DCR_DP	307
6.25.1.77	SPR_DCR_LAST	307
6.25.1.78	SPR_DCR_SC	307
6.25.1.79	SPR_DMMUCFGR	307
6.25.1.80	SPR_DMMUCFGR_CRI	307
6.25.1.81	SPR_DMMUCFGR_HTR	307
6.25.1.82	SPR_DMMUCFGR_NAE	307
6.25.1.83	SPR_DMMUCFGR_NTS	307
6.25.1.84	SPR_DMMUCFGR_NTS_OFF	307
6.25.1.85	SPR_DMMUCFGR_NTW	307
6.25.1.86	SPR_DMMUCFGR_NTW_OFF	307
6.25.1.87	SPR_DMMUCFGR_PRI	307
6.25.1.88	SPR_DMMUCFGR_TEIRI	307
6.25.1.89	SPR_DMMUCR	307
6.25.1.90	SPR_DMMUCR_P1S	307
6.25.1.91	SPR_DMMUCR_P2S	307
6.25.1.92	SPR_DMMUCR_PADDR_WIDTH	307
6.25.1.93	SPR_DMMUCR_VADDR_WIDTH	307
6.25.1.94	SPR_DMR1	307
6.25.1.95	SPR_DMR1_BT	307
6.25.1.96	SPR_DMR1_CW	307
6.25.1.97	SPR_DMR1_CW0	307
6.25.1.98	SPR_DMR1_CW0_AND	307
6.25.1.99	SPR_DMR1_CW0_OR	307
6.25.1.100	SPR_DMR1_CW1	307
6.25.1.101	SPR_DMR1_CW1_AND	307
6.25.1.102	SPR_DMR1_CW1_OR	307
6.25.1.103	SPR_DMR1_CW2	307
6.25.1.104	SPR_DMR1_CW2_AND	307
6.25.1.105	SPR_DMR1_CW2_OR	307
6.25.1.106	SPR_DMR1_CW3	307

6.25.1.107	SPR_DMR1_CW3_AND	307
6.25.1.108	SPR_DMR1_CW3_OR	307
6.25.1.109	SPR_DMR1_CW4	307
6.25.1.110	SPR_DMR1_CW4_AND	307
6.25.1.111	SPR_DMR1_CW4_OR	307
6.25.1.112	SPR_DMR1_CW5	307
6.25.1.113	SPR_DMR1_CW5_AND	307
6.25.1.114	SPR_DMR1_CW5_OR	307
6.25.1.115	SPR_DMR1_CW6	307
6.25.1.116	SPR_DMR1_CW6_AND	307
6.25.1.117	SPR_DMR1_CW6_OR	307
6.25.1.118	SPR_DMR1_CW7	307
6.25.1.119	SPR_DMR1_CW7_AND	307
6.25.1.120	SPR_DMR1_CW7_OR	307
6.25.1.121	SPR_DMR1_CW8	307
6.25.1.122	SPR_DMR1_CW8_AND	307
6.25.1.123	SPR_DMR1_CW8_OR	307
6.25.1.124	SPR_DMR1_CW9	307
6.25.1.125	SPR_DMR1_CW9_AND	307
6.25.1.126	SPR_DMR1_CW9_OR	307
6.25.1.127	SPR_DMR1_RES1	307
6.25.1.128	SPR_DMR1_RES2	307
6.25.1.129	SPR_DMR1_ST	307
6.25.1.130	SPR_DMR2	307
6.25.1.131	SPR_DMR2_AWTC	307
6.25.1.132	SPR_DMR2_AWTC_OFF	307
6.25.1.133	SPR_DMR2_WBS	307
6.25.1.134	SPR_DMR2_WBS_OFF	307
6.25.1.135	SPR_DMR2_WCE0	307
6.25.1.136	SPR_DMR2_WCE1	307
6.25.1.137	SPR_DMR2_WGB	307
6.25.1.138	SPR_DMR2_WGB_OFF	307
6.25.1.139	SPR_DRR	307
6.25.1.140	SPR_DRR_AE	307
6.25.1.141	SPR_DRR_BUSEE	307
6.25.1.142	SPR_DRR_DME	307

6.25.1.143	SPR_DRR_DPFE	307
6.25.1.144	SPR_DRR_IE	307
6.25.1.145	SPR_DRR_IIE	307
6.25.1.146	SPR_DRR_IME	307
6.25.1.147	SPR_DRR_IPFE	307
6.25.1.148	SPR_DRR_RE	307
6.25.1.149	SPR_DRR_RSTE	307
6.25.1.150	SPR_DRR_SCE	307
6.25.1.151	SPR_DRR_TE	307
6.25.1.152	SPR_DRR_TTE	307
6.25.1.153	SPR_DSR	307
6.25.1.154	SPR_DSR_AE	307
6.25.1.155	SPR_DSR_BUSEE	307
6.25.1.156	SPR_DSR_DME	307
6.25.1.157	SPR_DSR_DPFE	307
6.25.1.158	SPR_DSR_IE	307
6.25.1.159	SPR_DSR_IIE	307
6.25.1.160	SPR_DSR_IME	307
6.25.1.161	SPR_DSR_IPFE	307
6.25.1.162	SPR_DSR_RE	307
6.25.1.163	SPR_DSR_RSTE	307
6.25.1.164	SPR_DSR_SCE	307
6.25.1.165	SPR_DSR_SSE	307
6.25.1.166	SPR_DSR_TE	307
6.25.1.167	SPR_DSR_TTE	307
6.25.1.168	SPR_DTLBMR_BASE	307
6.25.1.169	SPR_DTLBMR_CID	307
6.25.1.170	SPR_DTLBMR_LAST	307
6.25.1.171	SPR_DTLBMR_LRU	307
6.25.1.172	SPR_DTLBMR_PL1	307
6.25.1.173	SPR_DTLBMR_V	307
6.25.1.174	SPR_DTLBMR_VPN	307
6.25.1.175	SPR_DTLBTR_A	307
6.25.1.176	SPR_DTLBTR_BASE	307
6.25.1.177	SPR_DTLBTR_CC	307
6.25.1.178	SPR_DTLBTR_CI	307

6.25.1.179	SPR_DTLBTR_D	307
6.25.1.180	SPR_DTLBTR_LAST	307
6.25.1.181	SPR_DTLBTR_PPN	307
6.25.1.182	SPR_DTLBTR_SRE	307
6.25.1.183	SPR_DTLBTR_SWE	307
6.25.1.184	SPR_DTLBTR_URE	307
6.25.1.185	SPR_DTLBTR_UWE	307
6.25.1.186	SPR_DTLBTR_WBC	307
6.25.1.187	SPR_DTLBTR_WOM	307
6.25.1.188	SPR_DVR	307
6.25.1.189	SPR_DWCR0	307
6.25.1.190	SPR_DWCR1	307
6.25.1.191	SPR_DWCR_COUNT	307
6.25.1.192	SPR_DWCR_MATCH	307
6.25.1.193	SPR_DWCR_MATCH_OFF	307
6.25.1.194	SPR_EEAR_BASE	307
6.25.1.195	SPR_EEAR_LAST	307
6.25.1.196	SPR_EPCR_BASE	307
6.25.1.197	SPR_EPCR_LAST	307
6.25.1.198	SPR_ESR_BASE	307
6.25.1.199	SPR_ESR_LAST	307
6.25.1.200	SPR_ICBIR	307
6.25.1.201	SPR_ICBLR	307
6.25.1.202	SPR_ICBPR	307
6.25.1.203	SPR_ICCFGR	307
6.25.1.204	SPR_ICCFGR_CBIRI	307
6.25.1.205	SPR_ICCFGR_CBLRI	307
6.25.1.206	SPR_ICCFGR_CBPRI	307
6.25.1.207	SPR_ICCFGR_CBS	307
6.25.1.208	SPR_ICCFGR_CBS_OFF	307
6.25.1.209	SPR_ICCFGR_CCRI	307
6.25.1.210	SPR_ICCFGR_NCS	307
6.25.1.211	SPR_ICCFGR_NCS_OFF	307
6.25.1.212	SPR_ICCFGR_NCW	307
6.25.1.213	SPR_ICCFGR_NCW_OFF	307
6.25.1.214	SPR_ICCR	307

6.25.1.215	SPR_ICCR_EW	307
6.25.1.216	SPR_ICR_BASE	307
6.25.1.217	SPR_ICR_LAST	307
6.25.1.218	SPR_IMMUCFGR	307
6.25.1.219	SPR_IMMUCFGR_CRI	307
6.25.1.220	SPR_IMMUCFGR_HTR	307
6.25.1.221	SPR_IMMUCFGR_NAE	307
6.25.1.222	SPR_IMMUCFGR_NTS	307
6.25.1.223	SPR_IMMUCFGR_NTS_OFF	307
6.25.1.224	SPR_IMMUCFGR_NTW	307
6.25.1.225	SPR_IMMUCFGR_NTW_OFF	307
6.25.1.226	SPR_IMMUCFGR_PRI	307
6.25.1.227	SPR_IMMUCFGR_TEIRI	307
6.25.1.228	SPR_IMMUCR	307
6.25.1.229	SPR_IMMUCR_P1S	307
6.25.1.230	SPR_IMMUCR_P2S	307
6.25.1.231	SPR_IMMUCR_PADDR_WIDTH	307
6.25.1.232	SPR_IMMUCR_VADDR_WIDTH	307
6.25.1.233	SPR_ITLBMR_BASE	307
6.25.1.234	SPR_ITLBMR_CID	307
6.25.1.235	SPR_ITLBMR_LAST	307
6.25.1.236	SPR_ITLBMR_LRU	307
6.25.1.237	SPR_ITLBMR_PL1	307
6.25.1.238	SPR_ITLBMR_V	307
6.25.1.239	SPR_ITLBMR_VPN	307
6.25.1.240	SPR_ITLBTR_A	307
6.25.1.241	SPR_ITLBTR_BASE	307
6.25.1.242	SPR_ITLBTR_CC	307
6.25.1.243	SPR_ITLBTR_CI	307
6.25.1.244	SPR_ITLBTR_D	307
6.25.1.245	SPR_ITLBTR_LAST	307
6.25.1.246	SPR_ITLBTR_PPN	307
6.25.1.247	SPR_ITLBTR_SXE	307
6.25.1.248	SPR_ITLBTR_UXE	307
6.25.1.249	SPR_ITLBTR_WBC	307
6.25.1.250	SPR_ITLBTR_WOM	307

6.25.1.251	SPR_MACHI	307
6.25.1.252	SPR_MACLO	307
6.25.1.253	SPR_NPC	307
6.25.1.254	SPR_PCCFGR	307
6.25.1.255	SPR_PCCR	307
6.25.1.256	SPR_PCMR	307
6.25.1.257	SPR_PCMR_BS	307
6.25.1.258	SPR_PCMR_CISM	307
6.25.1.259	SPR_PCMR_CIUM	307
6.25.1.260	SPR_PCMR_CP	307
6.25.1.261	SPR_PCMR_DCM	307
6.25.1.262	SPR_PCMR_DDS	307
6.25.1.263	SPR_PCMR_DTLBM	307
6.25.1.264	SPR_PCMR_ICM	307
6.25.1.265	SPR_PCMR_IF	307
6.25.1.266	SPR_PCMR_IFS	307
6.25.1.267	SPR_PCMR_ITLBM	307
6.25.1.268	SPR_PCMR_LA	307
6.25.1.269	SPR_PCMR_LSUS	307
6.25.1.270	SPR_PCMR_SA	307
6.25.1.271	SPR_PCMR_UMRA	307
6.25.1.272	SPR_PCMR_WPE	307
6.25.1.273	SPR_PICMR	307
6.25.1.274	SPR_PICMR_IUM	307
6.25.1.275	SPR_PICPR	307
6.25.1.276	SPR_PICPR_IPRIO	307
6.25.1.277	SPR_PICSR	307
6.25.1.278	SPR_PICSR_IS	307
6.25.1.279	SPR_PMR	307
6.25.1.280	SPR_PMR_DCGE	307
6.25.1.281	SPR_PMR_DME	307
6.25.1.282	SPR_PMR_SDF	307
6.25.1.283	SPR_PMR_SME	307
6.25.1.284	SPR_PMR_SUME	307
6.25.1.285	SPR_PPC	307
6.25.1.286	SPR_SR	307

6.25.1.287	SPR_SR_CE	307
6.25.1.288	SPR_SR_CID	307
6.25.1.289	SPR_SR_CY	307
6.25.1.290	SPR_SR_DCE	307
6.25.1.291	SPR_SR_DME	307
6.25.1.292	SPR_SR_DSX	307
6.25.1.293	SPR_SR_EPH	307
6.25.1.294	SPR_SR_F	307
6.25.1.295	SPR_SR_FO	307
6.25.1.296	SPR_SR_ICE	307
6.25.1.297	SPR_SR_IEE	307
6.25.1.298	SPR_SR_IME	307
6.25.1.299	SPR_SR_LEE	307
6.25.1.300	SPR_SR_OV	307
6.25.1.301	SPR_SR_OVE	307
6.25.1.302	SPR_SR_RES	307
6.25.1.303	SPR_SR_SM	307
6.25.1.304	SPR_SR_SUMRA	307
6.25.1.305	SPR_SR_TEE	307
6.25.1.306	SPR_TTCR	307
6.25.1.307	SPR_TTCR_PERIOD	307
6.25.1.308	SPR_TTMR	307
6.25.1.309	SPR_TTMR_CR	307
6.25.1.310	SPR_TTMR_IE	307
6.25.1.311	SPR_TTMR_IP	307
6.25.1.312	SPR_TTMR_M	307
6.25.1.313	SPR_TTMR_PERIOD	307
6.25.1.314	SPR_TTMR_RT	307
6.25.1.315	SPR_TTMR_SR	307
6.25.1.316	SPR_UPR	307
6.25.1.317	SPR_UPR_CUP	307
6.25.1.318	SPR_UPR_DCP	307
6.25.1.319	SPR_UPR_DMP	307
6.25.1.320	SPR_UPR_DUP	307
6.25.1.321	SPR_UPR_ICP	307
6.25.1.322	SPR_UPR_IMP	307

6.25.1.323	SPR_UPR_MP	307
6.25.1.324	SPR_UPR_PCUP	307
6.25.1.325	SPR_UPR_PICP	307
6.25.1.326	SPR_UPR_PMP	307
6.25.1.327	SPR_UPR_RES	307
6.25.1.328	SPR_UPR_TTP	307
6.25.1.329	SPR_UPR_UP	307
6.25.1.330	SPR_VR	307
6.25.1.331	SPR_VR_CFG	307
6.25.1.332	SPR_VR_CFG_OFF	307
6.25.1.333	SPR_VR_RES	307
6.25.1.334	SPR_VR_REV	307
6.25.1.335	SPR_VR_REV_OFF	307
6.25.1.336	SPR_VR_VER	307
6.25.1.337	SPR_VR_VER_OFF	307
6.25.1.338	SPRGROUP_D	307
6.25.1.339	SPRGROUP_DC	307
6.25.1.340	SPRGROUP_DMMU	307
6.25.1.341	SPRGROUP_IC	307
6.25.1.342	SPRGROUP_IMMU	307
6.25.1.343	SPRGROUP_MAC	307
6.25.1.344	SPRGROUP_PC	307
6.25.1.345	SPRGROUP_PIC	307
6.25.1.346	SPRGROUP_PM	307
6.25.1.347	SPRGROUP_SYS	307
6.25.1.348	SPRGROUP_TT	307
6.26	cpu/or1k/spr-dump.c File Reference	308
6.26.1	Function Documentation	309
6.26.1.1	dump_spr	309
6.26.2	Variable Documentation	309
6.26.2.1	ret_spr	309
6.26.2.2	spr_cpufgr	309
6.26.2.3	spr_d_group	310
6.26.2.4	spr_dc_group	310
6.26.2.5	spr_dccfgr	310
6.26.2.6	spr_dccr	310

6.26.2.7	spr_dcfgr	311
6.26.2.8	spr_dmmu_group	311
6.26.2.9	spr_dmmucfgr	311
6.26.2.10	spr_dmmucr	311
6.26.2.11	spr_dmr1	312
6.26.2.12	spr_dmr2	312
6.26.2.13	spr_drr	312
6.26.2.14	spr_dsr	312
6.26.2.15	spr_dtlbmr	313
6.26.2.16	spr_dtlbtr	313
6.26.2.17	spr_dwcr	313
6.26.2.18	spr_groups	314
6.26.2.19	spr_ic_group	314
6.26.2.20	spr_icefgr	314
6.26.2.21	spr_iccr	314
6.26.2.22	spr_immu_group	315
6.26.2.23	spr_immucfgr	315
6.26.2.24	spr_immucr	315
6.26.2.25	spr_itlbmr	315
6.26.2.26	spr_itlbtr	316
6.26.2.27	spr_mac_group	316
6.26.2.28	spr_one_val	316
6.26.2.29	spr_pc_group	316
6.26.2.30	spr_pcmr	316
6.26.2.31	spr_pic_group	317
6.26.2.32	spr_picmr	317
6.26.2.33	spr_pm_group	317
6.26.2.34	spr_pmr	317
6.26.2.35	spr_sr	318
6.26.2.36	spr_sys_group	318
6.26.2.37	spr_tt_group	318
6.26.2.38	spr_ttmr	319
6.26.2.39	spr_upr	319
6.26.2.40	spr_vr	319
6.27	cpu/or1k/spr-dump.h File Reference	320
6.27.1	Function Documentation	320

6.27.1.1	dump_spr	320
6.28	cpu/or1k/sprs.c File Reference	321
6.28.1	Function Documentation	322
6.28.1.1	DECLARE_DEBUG_CHANNEL	322
6.28.1.2	mfspr	322
6.28.1.3	mtspr	322
6.28.1.4	sprs_status	322
6.28.2	Variable Documentation	322
6.28.2.1	audio_cnt	322
6.28.2.2	fo	322
6.29	cpu/or1k/sprs.h File Reference	323
6.29.1	Function Documentation	323
6.29.1.1	dump_spr	323
6.29.1.2	mfspr	324
6.29.1.3	mtspr	324
6.29.1.4	sprs_status	324
6.30	cpu/or32/common-i386.h File Reference	325
6.30.1	Function Documentation	325
6.30.1.1	get_pc	325
6.30.1.2	set_pc	325
6.30.1.3	upd_sim_cycles	325
6.30.2	Variable Documentation	325
6.30.2.1	high32	325
6.30.2.2	low32	325
6.30.2.3	useless_x86	325
6.30.2.4	val3232	325
6.30.2.5	val64	325
6.31	cpu/or32/def-op-t.h File Reference	326
6.31.1	Define Documentation	326
6.31.1.1	DEF_1T_OP	326
6.31.1.2	DEF_2T_OP	326
6.31.1.3	DEF_2T_OP_NEQ	326
6.31.1.4	DEF_3T_OP	327
6.31.1.5	DEF_3T_OP_NEQ	327
6.31.1.6	DEF_GPR_OP	327
6.31.1.7	GPR_T	327

6.32	cpu/or32/dyn-rec.c File Reference	328
6.32.1	Define Documentation	335
6.32.1.1	OPS_ENLARGE_BY	335
6.32.1.2	RECED_PAGE_ENLARGE_BY	335
6.32.1.3	T_NONE	335
6.32.1.4	TFLAG_DST	335
6.32.1.5	TFLAG_SAVED	335
6.32.1.6	TFLAG_SOURCED	335
6.32.1.7	TFLAG_SRC	335
6.32.2	Typedef Documentation	335
6.32.2.1	generic_gen_op	335
6.32.2.2	imm_gen_op	335
6.32.3	Function Documentation	335
6.32.3.1	add_to_op_params	335
6.32.3.2	add_to_opq	335
6.32.3.3	DEF_1T_OP	335
6.32.3.4	DEF_1T_OP	335
6.32.3.5	DEF_1T_OP	335
6.32.3.6	DEF_1T_OP	335
6.32.3.7	DEF_1T_OP	335
6.32.3.8	DEF_1T_OP	335
6.32.3.9	DEF_1T_OP	335
6.32.3.10	DEF_1T_OP	335
6.32.3.11	DEF_1T_OP	335
6.32.3.12	DEF_1T_OP	335
6.32.3.13	DEF_1T_OP	335
6.32.3.14	DEF_1T_OP	335
6.32.3.15	DEF_1T_OP	335
6.32.3.16	DEF_1T_OP	335
6.32.3.17	DEF_1T_OP	335
6.32.3.18	DEF_1T_OP	335
6.32.3.19	DEF_1T_OP	335
6.32.3.20	DEF_1T_OP	335
6.32.3.21	DEF_1T_OP	335
6.32.3.22	DEF_1T_OP	335
6.32.3.23	DEF_1T_OP	335

6.32.3.24 DEF_1T_OP	335
6.32.3.25 DEF_1T_OP	335
6.32.3.26 DEF_1T_OP	335
6.32.3.27 DEF_1T_OP	335
6.32.3.28 DEF_1T_OP	335
6.32.3.29 DEF_1T_OP	335
6.32.3.30 DEF_1T_OP	335
6.32.3.31 DEF_1T_OP	335
6.32.3.32 DEF_1T_OP	335
6.32.3.33 DEF_1T_OP	335
6.32.3.34 DEF_1T_OP	335
6.32.3.35 DEF_1T_OP	335
6.32.3.36 DEF_1T_OP	335
6.32.3.37 DEF_1T_OP	335
6.32.3.38 DEF_1T_OP	335
6.32.3.39 DEF_1T_OP	335
6.32.3.40 DEF_1T_OP	335
6.32.3.41 DEF_1T_OP	335
6.32.3.42 DEF_1T_OP	335
6.32.3.43 DEF_1T_OP	335
6.32.3.44 DEF_1T_OP	335
6.32.3.45 DEF_2T_OP	335
6.32.3.46 DEF_2T_OP	335
6.32.3.47 DEF_2T_OP	335
6.32.3.48 DEF_2T_OP	335
6.32.3.49 DEF_2T_OP	335
6.32.3.50 DEF_2T_OP	335
6.32.3.51 DEF_2T_OP	335
6.32.3.52 DEF_2T_OP	335
6.32.3.53 DEF_2T_OP	335
6.32.3.54 DEF_2T_OP	335
6.32.3.55 DEF_2T_OP	335
6.32.3.56 DEF_2T_OP	335
6.32.3.57 DEF_2T_OP	335
6.32.3.58 DEF_2T_OP	335
6.32.3.59 DEF_2T_OP	335

6.32.3.60 DEF_2T_OP	335
6.32.3.61 DEF_2T_OP	335
6.32.3.62 DEF_2T_OP	335
6.32.3.63 DEF_2T_OP	335
6.32.3.64 DEF_2T_OP	335
6.32.3.65 DEF_2T_OP	335
6.32.3.66 DEF_2T_OP	335
6.32.3.67 DEF_2T_OP	335
6.32.3.68 DEF_2T_OP	335
6.32.3.69 DEF_2T_OP	335
6.32.3.70 DEF_2T_OP	335
6.32.3.71 DEF_2T_OP	335
6.32.3.72 DEF_2T_OP	335
6.32.3.73 DEF_2T_OP	335
6.32.3.74 DEF_2T_OP	335
6.32.3.75 DEF_2T_OP	335
6.32.3.76 DEF_2T_OP	335
6.32.3.77 DEF_2T_OP	335
6.32.3.78 DEF_2T_OP	335
6.32.3.79 DEF_2T_OP	335
6.32.3.80 DEF_2T_OP	335
6.32.3.81 DEF_2T_OP	335
6.32.3.82 DEF_2T_OP_NEQ	335
6.32.3.83 DEF_3T_OP	335
6.32.3.84 DEF_3T_OP	335
6.32.3.85 DEF_3T_OP	335
6.32.3.86 DEF_3T_OP	335
6.32.3.87 DEF_3T_OP	335
6.32.3.88 DEF_3T_OP	335
6.32.3.89 DEF_3T_OP	335
6.32.3.90 DEF_3T_OP	335
6.32.3.91 DEF_3T_OP	335
6.32.3.92 DEF_3T_OP	335
6.32.3.93 DEF_3T_OP_NEQ	335
6.32.3.94 DEF_3T_OP_NEQ	335
6.32.3.95 DEF_3T_OP_NEQ	335

6.32.3.96 DEF_3T_OP_NEQ	335
6.32.3.97 DEF_GPR_OP	335
6.32.3.98 DEF_GPR_OP	335
6.32.3.99 dirtyfy_page	335
6.32.3.100 dyn_checkwrite	336
6.32.3.101 dyn_main	336
6.32.3.102 dyn_ret_stack_prot	337
6.32.3.103 dyn_sigsegv_debug	337
6.32.3.104 enough_host_page	337
6.32.3.105 eval_insn_ops	337
6.32.3.106 find_jump_loc	337
6.32.3.107 find_t	338
6.32.3.108 gen_j_imm	338
6.32.3.109 gen_j_reg	338
6.32.3.110 gen_l_add	338
6.32.3.111 gen_l_addc	338
6.32.3.112 gen_l_and	338
6.32.3.113 gen_l_bf	338
6.32.3.114 gen_l_bnf	339
6.32.3.115 gen_l_cmov	340
6.32.3.116 gen_l_cust1	340
6.32.3.117 gen_l_cust2	340
6.32.3.118 gen_l_cust3	340
6.32.3.119 gen_l_cust4	340
6.32.3.120 gen_l_cust5	340
6.32.3.121 gen_l_cust6	340
6.32.3.122 gen_l_cust7	340
6.32.3.123 gen_l_cust8	340
6.32.3.124 gen_l_div	340
6.32.3.125 gen_l_divu	340
6.32.3.126 gen_l_extbs	340
6.32.3.127 gen_l_extbz	340
6.32.3.128 gen_l_exths	340
6.32.3.129 gen_l_exthz	340
6.32.3.130 gen_l_extws	340
6.32.3.131 gen_l_extwz	340

6.32.3.132	gen_l_ffl	340
6.32.3.133	gen_l_invalid	340
6.32.3.134	gen_l_j	340
6.32.3.135	gen_l_jal	341
6.32.3.136	gen_l_jalr	341
6.32.3.137	gen_l_jr	341
6.32.3.138	gen_l_lbs	341
6.32.3.139	gen_l_lbz	342
6.32.3.140	gen_l_lhs	342
6.32.3.141	gen_l_lhz	342
6.32.3.142	gen_l_lws	342
6.32.3.143	gen_l_lwz	342
6.32.3.144	gen_l_mac	343
6.32.3.145	gen_l_macrc	343
6.32.3.146	gen_l_mfspr	343
6.32.3.147	gen_l_movhi	343
6.32.3.148	gen_l_msb	343
6.32.3.149	gen_l_mtspr	343
6.32.3.150	gen_l_mul	343
6.32.3.151	gen_l_mulu	343
6.32.3.152	gen_l_nop	343
6.32.3.153	gen_l_or	343
6.32.3.154	gen_l_rfe	343
6.32.3.155	gen_l_sb	343
6.32.3.156	gen_l_sfeq	344
6.32.3.157	gen_l_sfges	344
6.32.3.158	gen_l_sfgeu	344
6.32.3.159	gen_l_sfgts	344
6.32.3.160	gen_l_sfgtu	344
6.32.3.161	gen_l_sfles	344
6.32.3.162	gen_l_sfleu	344
6.32.3.163	gen_l_sftts	344
6.32.3.164	gen_l_sfttu	344
6.32.3.165	gen_l_sfne	344
6.32.3.166	gen_l_sh	344
6.32.3.167	gen_l_sll	344

6.32.3.168	gen_l_sra	344
6.32.3.169	gen_l_srl	344
6.32.3.170	gen_l_sub	344
6.32.3.171	gen_l_sw	344
6.32.3.172	gen_l_sys	345
6.32.3.173	gen_l_trap	345
6.32.3.174	gen_l_xor	345
6.32.3.175	gen_lf_add_s	345
6.32.3.176	gen_lf_div_s	345
6.32.3.177	gen_lf_ftoi_s	345
6.32.3.178	gen_lf_itof_s	345
6.32.3.179	gen_lf_madd_s	345
6.32.3.180	gen_lf_mul_s	346
6.32.3.181	gen_lf_rem_s	346
6.32.3.182	gen_lf_sfreq_s	346
6.32.3.183	gen_lf_sfge_s	346
6.32.3.184	gen_lf_sfgt_s	346
6.32.3.185	gen_lf_sfle_s	346
6.32.3.186	gen_lf_sflt_s	347
6.32.3.187	gen_lf_sfne_s	347
6.32.3.188	gen_lf_sub_s	347
6.32.3.189	gen_op_mark_loc	347
6.32.3.190	immu_retranslate	348
6.32.3.191	init_dyn_recomp	348
6.32.3.192	new_dp	348
6.32.3.193	recheck_immu	348
6.32.3.194	recompile_delay_insn	349
6.32.3.195	recompile_insn	349
6.32.3.196	recompile_page	349
6.32.3.197	run_sched_out_of_line	350
6.32.3.198	ship_gprs_out_t	350
6.32.3.199	ship_t_out	350
6.32.4	Variable Documentation	350
6.32.4.1	__op_param1	350
6.32.4.2	__op_param2	350
6.32.4.3	__op_param3	350

6.32.4.4	do_stats	350
6.32.4.5	set_pc_delay_gpr	351
6.32.4.6	sigsegv_addr	351
6.32.4.7	sigsegv_state	351
6.33	cpu/or32/dyn-rec.h File Reference	352
6.33.1	Define Documentation	353
6.33.1.1	glue	353
6.33.1.2	IMMU_GOT_DISABLED	353
6.33.1.3	IMMU_GOT_ENABLED	353
6.33.1.4	xglue	353
6.33.2	Typedef Documentation	353
6.33.2.1	gen_code_ent	353
6.33.3	Function Documentation	353
6.33.3.1	add_to_op_params	353
6.33.3.2	add_to_opq	353
6.33.3.3	dyn_checkwrite	353
6.33.3.4	dyn_main	354
6.33.3.5	enough_host_page	354
6.33.3.6	enter_dyn_code	354
6.33.3.7	init_dyn_recomp	354
6.33.3.8	new_dp	355
6.33.3.9	recheck_immu	355
6.33.3.10	recompile_page	355
6.33.3.11	run_sched_out_of_line	356
6.33.4	Variable Documentation	356
6.33.4.1	rec_stack_base	356
6.34	cpu/or32/dyn32-defs.h File Reference	357
6.34.1	Function Documentation	357
6.34.1.1	l_none	357
6.34.1.2	PARAMS	357
6.35	cpu/or32/dyngen-elf.c File Reference	358
6.35.1	Function Documentation	358
6.35.1.1	elf_close_obj	358
6.35.1.2	elf_find_func	358
6.35.1.3	elf_get_func_len	358
6.35.1.4	elf_get_func_name	359

6.35.1.5	elf_get_func_reloc	359
6.35.1.6	elf_get_func_start	359
6.35.1.7	elf_get_sym_name	359
6.35.1.8	elf_open_obj	359
6.35.2	Variable Documentation	359
6.35.2.1	bffs	359
6.36	cpu/or32/dyngen-i386.c File Reference	360
6.36.1	Define Documentation	360
6.36.1.1	RET_OPCODE	360
6.36.2	Function Documentation	360
6.36.2.1	i386_gen_func_reloc	360
6.36.2.2	i386_gen_reloc	360
6.36.2.3	i386_get_real_func_len	360
6.36.3	Variable Documentation	360
6.36.3.1	archfs	360
6.37	cpu/or32/dyngen.c File Reference	362
6.37.1	Define Documentation	363
6.37.1.1	MAX_PARAMS	363
6.37.1.2	OP_FUNC_PARAM_PREFIX	363
6.37.1.3	OP_FUNC_PREFIX	363
6.37.2	Function Documentation	363
6.37.2.1	gen_func_proto	363
6.37.2.2	main	363
6.37.3	Variable Documentation	363
6.37.3.1	c_file_head	363
6.37.3.2	c_rel_file_head	363
6.37.3.3	c_rel_file_tail	363
6.37.3.4	c_sw_file_head	364
6.37.3.5	c_sw_file_tail	364
6.37.3.6	gen_code_proto	364
6.38	cpu/or32/dyngen.h File Reference	365
6.38.1	Variable Documentation	365
6.38.1.1	archfs	365
6.38.1.2	bffs	365
6.39	cpu/or32/execute.c File Reference	366
6.39.1	Function Documentation	367

6.39.1.1	analysis	367
6.39.1.2	check_depend	368
6.39.1.3	cpu_clock	368
6.39.1.4	cpu_reset	369
6.39.1.5	decode_execute	369
6.39.1.6	decode_execute_wrapper	369
6.39.1.7	dump_exe_log	370
6.39.1.8	dumpreg	370
6.39.1.9	eval_operand_val	371
6.39.1.10	evalsim_reg	371
6.39.1.11	exec_main	372
6.39.1.12	fetch	374
6.39.1.13	l_invalid	374
6.39.1.14	sbuf_load	375
6.39.1.15	sbuf_store	375
6.39.1.16	setsim_reg	375
6.39.1.17	update_pc	375
6.39.2	Variable Documentation	375
6.39.2.1	breakpoint	375
6.39.2.2	cpu_state	375
6.39.2.3	do_stats	375
6.39.2.4	hist_exec_tail	376
6.39.2.5	issued_per_cycle	376
6.39.2.6	multissue	376
6.39.2.7	next_delay_insn	376
6.39.2.8	pcnext	376
6.39.2.9	sbuf_buf	376
6.39.2.10	sbuf_count	376
6.39.2.11	sbuf_head	376
6.39.2.12	sbuf_prev_cycles	376
6.39.2.13	sbuf_tail	376
6.39.2.14	sbuf_total_cyc	376
6.39.2.15	sbuf_wait_cyc	376
6.40	cpu/or32/generate.c File Reference	377
6.40.1	Function Documentation	377
6.40.1.1	gen_eval_operands	377

6.40.1.2	generate_body	378
6.40.1.3	generate_footer	378
6.40.1.4	generate_header	378
6.40.1.5	main	378
6.40.1.6	output_call	379
6.40.1.7	output_function	379
6.40.1.8	shift_fprintf	379
6.40.2	Variable Documentation	379
6.40.2.1	in_file	379
6.40.2.2	out_file	379
6.40.2.3	out_lines	379
6.40.2.4	write_to_reg	379
6.41	cpu/or32/i386-regs.h File Reference	380
6.41.1	Define Documentation	380
6.41.1.1	CPU_STATE_REG	380
6.41.1.2	NUM_T_REGS	380
6.41.1.3	T0_REG	380
6.41.1.4	T1_REG	380
6.41.1.5	T2_REG	380
6.42	cpu/or32/insnset.c File Reference	381
6.42.1	Function Documentation	384
6.42.1.1	INSTRUCTION	384
6.42.1.2	INSTRUCTION	384
6.42.1.3	INSTRUCTION	384
6.42.1.4	INSTRUCTION	384
6.42.1.5	INSTRUCTION	384
6.42.1.6	INSTRUCTION	384
6.42.1.7	INSTRUCTION	384
6.42.1.8	INSTRUCTION	384
6.42.1.9	INSTRUCTION	384
6.42.1.10	INSTRUCTION	384
6.42.1.11	INSTRUCTION	384
6.42.1.12	INSTRUCTION	384
6.42.1.13	INSTRUCTION	384
6.42.1.14	INSTRUCTION	384
6.42.1.15	INSTRUCTION	384

6.42.1.16 INSTRUCTION	384
6.42.1.17 INSTRUCTION	384
6.42.1.18 INSTRUCTION	384
6.42.1.19 INSTRUCTION	384
6.42.1.20 INSTRUCTION	384
6.42.1.21 INSTRUCTION	384
6.42.1.22 INSTRUCTION	384
6.42.1.23 INSTRUCTION	384
6.42.1.24 INSTRUCTION	384
6.42.1.25 INSTRUCTION	385
6.42.1.26 INSTRUCTION	385
6.42.1.27 INSTRUCTION	385
6.42.1.28 INSTRUCTION	386
6.42.1.29 INSTRUCTION	386
6.42.1.30 INSTRUCTION	386
6.42.1.31 INSTRUCTION	386
6.42.1.32 INSTRUCTION	386
6.42.1.33 INSTRUCTION	386
6.42.1.34 INSTRUCTION	386
6.42.1.35 INSTRUCTION	386
6.42.1.36 INSTRUCTION	386
6.42.1.37 INSTRUCTION	386
6.42.1.38 INSTRUCTION	386
6.42.1.39 INSTRUCTION	386
6.42.1.40 INSTRUCTION	386
6.42.1.41 INSTRUCTION	386
6.42.1.42 INSTRUCTION	386
6.42.1.43 INSTRUCTION	386
6.42.1.44 INSTRUCTION	386
6.42.1.45 INSTRUCTION	387
6.42.1.46 INSTRUCTION	387
6.42.1.47 INSTRUCTION	387
6.42.1.48 INSTRUCTION	387
6.42.1.49 INSTRUCTION	388
6.42.1.50 INSTRUCTION	388
6.42.1.51 INSTRUCTION	388

6.42.1.52 INSTRUCTION	388
6.42.1.53 INSTRUCTION	388
6.42.1.54 INSTRUCTION	388
6.42.1.55 INSTRUCTION	388
6.42.1.56 INSTRUCTION	389
6.42.1.57 INSTRUCTION	389
6.42.1.58 INSTRUCTION	389
6.42.1.59 INSTRUCTION	389
6.42.1.60 INSTRUCTION	389
6.42.1.61 INSTRUCTION	389
6.42.1.62 INSTRUCTION	389
6.42.1.63 INSTRUCTION	389
6.42.1.64 INSTRUCTION	390
6.42.1.65 INSTRUCTION	390
6.42.1.66 INSTRUCTION	390
6.42.1.67 INSTRUCTION	391
6.42.1.68 INSTRUCTION	391
6.42.1.69 INSTRUCTION	392
6.42.1.70 INSTRUCTION	392
6.42.1.71 INSTRUCTION	392
6.42.1.72 INSTRUCTION	392
6.43 cpu/or32/op-1t-op.h File Reference	393
6.43.1 Function Documentation	393
6.43.1.1 glue	393
6.43.1.2 glue	393
6.43.1.3 glue	393
6.43.1.4 glue	393
6.43.1.5 glue	393
6.43.1.6 glue	393
6.43.1.7 glue	393
6.44 cpu/or32/op-1t.h File Reference	394
6.44.1 Define Documentation	394
6.44.1.1 OP_1T	394
6.44.1.2 T	394
6.45 cpu/or32/op-2t-op.h File Reference	395
6.45.1 Function Documentation	395

6.45.1.1	glue	395
6.45.1.2	glue	395
6.45.1.3	glue	395
6.46	cpu/or32/op-2t.h File Reference	396
6.46.1	Define Documentation	396
6.46.1.1	OP_2T	396
6.46.1.2	T	396
6.47	cpu/or32/op-3t-op.h File Reference	397
6.47.1	Function Documentation	397
6.47.1.1	glue	397
6.48	cpu/or32/op-3t.h File Reference	398
6.48.1	Define Documentation	398
6.48.1.1	OP_3T	398
6.48.1.2	T	398
6.49	cpu/or32/op-arith-op.h File Reference	399
6.50	cpu/or32/op-comp-op.h File Reference	400
6.51	cpu/or32/op-extend-op.h File Reference	401
6.51.1	Function Documentation	401
6.51.1.1	glue	401
6.52	cpu/or32/op-ff1-op.h File Reference	402
6.52.1	Function Documentation	402
6.52.1.1	glue	402
6.53	cpu/or32/op-i386.h File Reference	403
6.53.1	Define Documentation	403
6.53.1.1	FORCE_RET	403
6.53.1.2	OP_JUMP	403
6.53.1.3	SPEEDY_CALL	403
6.53.2	Function Documentation	403
6.53.2.1	asm	403
6.54	cpu/or32/op-lwhb-op.h File Reference	404
6.55	cpu/or32/op-mac-op.h File Reference	405
6.55.1	Function Documentation	405
6.55.1.1	glue	405
6.56	cpu/or32/op-mftspr-op.h File Reference	406
6.56.1	Function Documentation	406
6.56.1.1	op_mtspr_imm_clear	406

6.57	cpu/or32/op-support.c File Reference	407
6.57.1	Function Documentation	408
6.57.1.1	do_jump	408
6.57.1.2	op_support_analysis	408
6.57.1.3	op_support_nop_exit	408
6.57.1.4	op_support_nop_printf	408
6.57.1.5	op_support_nop_report	408
6.57.1.6	op_support_nop_report_imm	408
6.57.1.7	op_support_nop_reset	408
6.58	cpu/or32/op-support.h File Reference	409
6.58.1	Function Documentation	409
6.58.1.1	do_jump	409
6.58.1.2	op_support_analysis	409
6.58.1.3	op_support_nop_exit	410
6.58.1.4	op_support_nop_printf	410
6.58.1.5	op_support_nop_report	410
6.58.1.6	op_support_nop_report_imm	410
6.58.1.7	op_support_nop_reset	410
6.58.1.8	upd_reg_from_t	410
6.59	cpu/or32/op-swhb-op.h File Reference	411
6.59.1	Function Documentation	411
6.59.1.1	glue	411
6.60	cpu/or32/op-t-reg-mov-op.h File Reference	412
6.60.1	Function Documentation	415
6.60.1.1	glue	415
6.60.1.2	glue	415
6.60.1.3	glue	415
6.60.1.4	glue	415
6.60.1.5	glue	415
6.60.1.6	glue	415
6.60.1.7	glue	415
6.60.1.8	glue	415
6.60.1.9	glue	415
6.60.1.10	glue	415
6.60.1.11	glue	415
6.60.1.12	glue	415

6.60.1.13 glue	415
6.60.1.14 glue	415
6.60.1.15 glue	415
6.60.1.16 glue	415
6.60.1.17 glue	415
6.60.1.18 glue	415
6.60.1.19 glue	415
6.60.1.20 glue	415
6.60.1.21 glue	415
6.60.1.22 glue	415
6.60.1.23 glue	415
6.60.1.24 glue	415
6.60.1.25 glue	415
6.60.1.26 glue	415
6.60.1.27 glue	415
6.60.1.28 glue	415
6.60.1.29 glue	415
6.60.1.30 glue	415
6.60.1.31 glue	415
6.60.1.32 glue	415
6.60.1.33 glue	415
6.60.1.34 glue	415
6.60.1.35 glue	415
6.60.1.36 glue	415
6.60.1.37 glue	415
6.60.1.38 glue	415
6.60.1.39 glue	415
6.60.1.40 glue	415
6.60.1.41 glue	415
6.60.1.42 glue	415
6.60.1.43 glue	415
6.60.1.44 glue	415
6.60.1.45 glue	415
6.60.1.46 glue	415
6.60.1.47 glue	415
6.60.1.48 glue	415

6.60.1.49 glue	415
6.60.1.50 glue	415
6.60.1.51 glue	415
6.60.1.52 glue	415
6.60.1.53 glue	415
6.60.1.54 glue	415
6.60.1.55 glue	415
6.60.1.56 glue	415
6.60.1.57 glue	415
6.60.1.58 glue	415
6.60.1.59 glue	415
6.60.1.60 glue	415
6.60.1.61 glue	415
6.60.1.62 glue	415
6.61 cpu/or32/op.c File Reference	416
6.61.1 Define Documentation	423
6.61.1.1 __or_dynop	423
6.61.1.2 COMP	423
6.61.1.3 COMP	423
6.61.1.4 COMP	423
6.61.1.5 COMP	423
6.61.1.6 COMP	423
6.61.1.7 COMP	423
6.61.1.8 COMP	423
6.61.1.9 COMP	423
6.61.1.10 COMP	423
6.61.1.11 COMP	423
6.61.1.12 COMP_CAST	423
6.61.1.13 COMP_CAST	423
6.61.1.14 COMP_CAST	423
6.61.1.15 COMP_CAST	423
6.61.1.16 COMP_CAST	423
6.61.1.17 COMP_CAST	423
6.61.1.18 COMP_CAST	423
6.61.1.19 COMP_CAST	423
6.61.1.20 COMP_CAST	423

6.61.1.21 COMP_CAST	423
6.61.1.22 COMP_NAME	423
6.61.1.23 COMP_NAME	423
6.61.1.24 COMP_NAME	423
6.61.1.25 COMP_NAME	423
6.61.1.26 COMP_NAME	423
6.61.1.27 COMP_NAME	423
6.61.1.28 COMP_NAME	423
6.61.1.29 COMP_NAME	423
6.61.1.30 COMP_NAME	423
6.61.1.31 COMP_NAME	423
6.61.1.32 EXT_CAST	423
6.61.1.33 EXT_CAST	423
6.61.1.34 EXT_CAST	423
6.61.1.35 EXT_CAST	423
6.61.1.36 EXT_NAME	423
6.61.1.37 EXT_NAME	423
6.61.1.38 EXT_NAME	423
6.61.1.39 EXT_NAME	423
6.61.1.40 EXT_TYPE	423
6.61.1.41 EXT_TYPE	423
6.61.1.42 EXT_TYPE	423
6.61.1.43 EXT_TYPE	423
6.61.1.44 LS_OP_CAST	423
6.61.1.45 LS_OP_CAST	423
6.61.1.46 LS_OP_CAST	423
6.61.1.47 LS_OP_CAST	423
6.61.1.48 LS_OP_CAST	423
6.61.1.49 LS_OP_CAST	423
6.61.1.50 LS_OP_FUNC	423
6.61.1.51 LS_OP_FUNC	423
6.61.1.52 LS_OP_FUNC	423
6.61.1.53 LS_OP_FUNC	423
6.61.1.54 LS_OP_FUNC	423
6.61.1.55 LS_OP_FUNC	423
6.61.1.56 LS_OP_NAME	423

6.61.1.57 LS_OP_NAME	423
6.61.1.58 LS_OP_NAME	423
6.61.1.59 LS_OP_NAME	423
6.61.1.60 LS_OP_NAME	423
6.61.1.61 LS_OP_NAME	423
6.61.1.62 OP	423
6.61.1.63 OP	423
6.61.1.64 OP	423
6.61.1.65 OP	423
6.61.1.66 OP	423
6.61.1.67 OP	423
6.61.1.68 OP	423
6.61.1.69 OP	423
6.61.1.70 OP	423
6.61.1.71 OP	423
6.61.1.72 OP	423
6.61.1.73 OP	423
6.61.1.74 OP	423
6.61.1.75 OP	423
6.61.1.76 OP	423
6.61.1.77 OP_CAST	423
6.61.1.78 OP_CAST	423
6.61.1.79 OP_CAST	423
6.61.1.80 OP_CAST	423
6.61.1.81 OP_CAST	423
6.61.1.82 OP_CAST	423
6.61.1.83 OP_CAST	423
6.61.1.84 OP_CAST	423
6.61.1.85 OP_CAST	423
6.61.1.86 OP_CAST	423
6.61.1.87 OP_CAST	423
6.61.1.88 OP_CAST	423
6.61.1.89 OP_CAST	423
6.61.1.90 OP_EXTRA	423
6.61.1.91 OP_EXTRA	423
6.61.1.92 OP_EXTRA	423

6.61.1.93 OP_FILE	423
6.61.1.94 OP_FILE	423
6.61.1.95 OP_FILE	423
6.61.1.96 OP_FILE	423
6.61.1.97 OP_FILE	423
6.61.1.98 OP_FILE	423
6.61.1.99 OP_FILE	423
6.61.1.100OP_FILE	423
6.61.1.101OP_FILE	423
6.61.1.102OP_FILE	423
6.61.1.103OP_FILE	423
6.61.1.104OP_NAME	423
6.61.1.105OP_NAME	423
6.61.1.106OP_NAME	423
6.61.1.107OP_NAME	423
6.61.1.108OP_NAME	423
6.61.1.109OP_NAME	423
6.61.1.110OP_NAME	423
6.61.1.111OP_NAME	423
6.61.1.112OP_NAME	423
6.61.1.113OP_NAME	423
6.61.1.114OP_NAME	423
6.61.1.115OP_NAME	423
6.61.1.116OP_NAME	423
6.61.1.117OP_NAME	423
6.61.1.118OP_NAME	423
6.61.1.119OP_PARAM1	423
6.61.1.120OP_PARAM2	423
6.61.1.121OP_PARAM3	423
6.61.1.122S_FUNC	423
6.61.1.123S_FUNC	423
6.61.1.124S_FUNC	423
6.61.1.125S_OP_NAME	423
6.61.1.126S_OP_NAME	423
6.61.1.127S_OP_NAME	423
6.61.2 Function Documentation	423

6.61.2.1	asm	423
6.61.2.2	asm	423
6.61.2.3	asm	423
6.61.2.4	asm	423
6.61.2.5	do_sched_wrap	423
6.61.2.6	do_sched_wrap_delay	424
6.61.2.7	enter_dyn_code	424
6.61.2.8	op_add_pc	424
6.61.2.9	op_analysis	424
6.61.2.10	op_check_delay_slot	424
6.61.2.11	op_check_flag	424
6.61.2.12	op_check_flag_delay	425
6.61.2.13	op_check_not_flag	425
6.61.2.14	op_check_not_flag_delay	425
6.61.2.15	op_clear_delay_insn	425
6.61.2.16	op_clear_flag	425
6.61.2.17	op_clear_pc_delay	425
6.61.2.18	op_do_jump_delay	425
6.61.2.19	op_do_sched	425
6.61.2.20	op_do_sched_delay	425
6.61.2.21	op_illegal	427
6.61.2.22	op_illegal_delay	427
6.61.2.23	op_jump_imm	427
6.61.2.24	op_join_mem_cycles	427
6.61.2.25	op_macc	427
6.61.2.26	op_move_gpr10_pc_delay	427
6.61.2.27	op_move_gpr11_pc_delay	427
6.61.2.28	op_move_gpr12_pc_delay	427
6.61.2.29	op_move_gpr13_pc_delay	427
6.61.2.30	op_move_gpr14_pc_delay	427
6.61.2.31	op_move_gpr15_pc_delay	427
6.61.2.32	op_move_gpr16_pc_delay	427
6.61.2.33	op_move_gpr17_pc_delay	427
6.61.2.34	op_move_gpr18_pc_delay	427
6.61.2.35	op_move_gpr19_pc_delay	427
6.61.2.36	op_move_gpr1_pc_delay	427

6.61.2.37 op_move_gpr20_pc_delay	427
6.61.2.38 op_move_gpr21_pc_delay	427
6.61.2.39 op_move_gpr22_pc_delay	427
6.61.2.40 op_move_gpr23_pc_delay	427
6.61.2.41 op_move_gpr24_pc_delay	427
6.61.2.42 op_move_gpr25_pc_delay	427
6.61.2.43 op_move_gpr26_pc_delay	427
6.61.2.44 op_move_gpr27_pc_delay	427
6.61.2.45 op_move_gpr28_pc_delay	427
6.61.2.46 op_move_gpr29_pc_delay	427
6.61.2.47 op_move_gpr2_pc_delay	427
6.61.2.48 op_move_gpr30_pc_delay	427
6.61.2.49 op_move_gpr31_pc_delay	427
6.61.2.50 op_move_gpr3_pc_delay	427
6.61.2.51 op_move_gpr4_pc_delay	427
6.61.2.52 op_move_gpr5_pc_delay	427
6.61.2.53 op_move_gpr6_pc_delay	427
6.61.2.54 op_move_gpr7_pc_delay	427
6.61.2.55 op_move_gpr8_pc_delay	427
6.61.2.56 op_move_gpr9_pc_delay	427
6.61.2.57 op_nop_exit	427
6.61.2.58 op_nop_printf	428
6.61.2.59 op_nop_report	428
6.61.2.60 op_nop_report_imm	428
6.61.2.61 op_nop_reset	428
6.61.2.62 op_prep_rfe	428
6.61.2.63 op_prep_sys	428
6.61.2.64 op_prep_sys_delay	429
6.61.2.65 op_prep_trap	429
6.61.2.66 op_prep_trap_delay	429
6.61.2.67 op_set_delay_insn	429
6.61.2.68 op_set_flag	429
6.61.2.69 op_set_pc_delay_imm	429
6.61.2.70 op_set_pc_delay_pc	429
6.61.2.71 op_set_pc_pc_delay	429
6.61.2.72 op_store_insn_ea	429

6.61.2.73	op_store_link_addr_gpr	429
6.61.2.74	prep_except	429
6.61.2.75	save_t_bound	429
6.61.3	Variable Documentation	429
6.61.3.1	__op_param1	429
6.61.3.2	__op_param2	429
6.61.3.3	__op_param3	429
6.62	cpu/or32/or32.c File Reference	430
6.62.1	Define Documentation	432
6.62.1.1	EF	432
6.62.1.2	EFI	432
6.62.1.3	EFN	432
6.62.1.4	MAX_AUTOMATA_SIZE	432
6.62.1.5	MAX_LEN	432
6.62.1.6	MAX_OP_TABLE_SIZE	432
6.62.1.7	MIN	432
6.62.2	Function Documentation	432
6.62.2.1	build_automata	432
6.62.2.2	cover_insn	432
6.62.2.3	destruct_automata	433
6.62.2.4	disassemble_index	433
6.62.2.5	disassemble_insn	433
6.62.2.6	extend_imm	433
6.62.2.7	insn_decode	434
6.62.2.8	insn_extract	434
6.62.2.9	insn_index	434
6.62.2.10	insn_len	434
6.62.2.11	insn_name	434
6.62.2.12	l_none	434
6.62.2.13	letter_range	434
6.62.2.14	letter_signed	434
6.62.2.15	num_ones	434
6.62.2.16	or32_debug	434
6.62.2.17	or32_extract	434
6.62.2.18	or32_print_immediate	434
6.62.2.19	or32_print_register	434

6.62.2.20	parse_params	435
6.62.3	Variable Documentation	435
6.62.3.1	automata	435
6.62.3.2	curpass	435
6.62.3.3	disassembled	435
6.62.3.4	disassembled_str	435
6.62.3.5	num_opcodes	435
6.62.3.6	nuncovered	435
6.62.3.7	op_data	435
6.62.3.8	op_start	435
6.62.3.9	or32_letters	435
6.62.3.10	or32_opcodes	436
6.62.3.11	range_cache	436
6.62.3.12	ti	436
6.63	cpu/or32/rec-i386.h File Reference	437
6.63.1	Function Documentation	437
6.63.1.1	get_sp	437
6.64	cpu/or32/sched-i386.h File Reference	438
6.64.1	Function Documentation	438
6.64.1.1	set_sched_cycle	438
6.65	cpu/or32/simpl32-defs.h File Reference	439
6.65.1	Function Documentation	439
6.65.1.1	PARAMS	439
6.65.1.2	PARAMS	439
6.66	cuc/adv.c File Reference	440
6.66.1	Function Documentation	441
6.66.1.1	calc_max	441
6.66.1.2	detect_max_values	441
6.66.1.3	insert_conditional_facts	441
6.66.1.4	mark_successors	441
6.66.1.5	mask	441
6.66.1.6	max_op	441
6.67	cuc/bb.c File Reference	442
6.67.1	Function Documentation	443
6.67.1.1	build_bb	443
6.67.1.2	count_bb_seq	443

6.67.1.3	cpy_bb	443
6.67.1.4	cuc_check	443
6.67.1.5	detect_bb	444
6.67.1.6	dup_func	444
6.67.1.7	expand_bb	444
6.67.1.8	free_func	444
6.67.1.9	generate_bb_seq	444
6.67.1.10	join_bb	445
6.67.1.11	optimize_bb	445
6.67.1.12	preunroll_loop	445
6.67.1.13	print_bb_num	446
6.67.1.14	print_cuc_bb	446
6.67.1.15	recalc_last_used_reg	446
6.67.1.16	reg_dep	446
6.67.1.17	reg_dep_rec	446
6.67.1.18	relocate_bb	446
6.67.1.19	remove_dead_bb	446
6.67.1.20	roll_loop	446
6.67.1.21	simplify_bb	447
6.68	cuc/cuc.c File Reference	448
6.68.1	Function Documentation	451
6.68.1.1	analyse_function	451
6.68.1.2	calc_cycles	452
6.68.1.3	calc_size	452
6.68.1.4	cuc_calling_conv	452
6.68.1.5	cuc_enable_bursts	452
6.68.1.6	cuc_memory_order	452
6.68.1.7	cuc_no_multicycle	453
6.68.1.8	cuc_optimize	453
6.68.1.9	cuc_timings_fn	454
6.68.1.10	extract_function	454
6.68.1.11	format_func_options	454
6.68.1.12	gen_option	456
6.68.1.13	generate_function	456
6.68.1.14	main_cuc	458
6.68.1.15	options_cmd	459

6.68.1.16	preunroll_bb	461
6.68.1.17	print_option	462
6.68.1.18	reg_cuc_sec	462
6.68.1.19	set_func_deps	462
6.68.1.20	tim_comp	462
6.68.2	Variable Documentation	462
6.68.2.1	caller_saved	462
6.68.2.2	cuc_debug	463
6.68.2.3	flog	463
6.68.2.4	func	463
6.68.2.5	func_v	463
6.68.2.6	option_char	463
6.69	cuc/cuc.h File Reference	464
6.69.1	Define Documentation	469
6.69.1.1	BB_DEAD	469
6.69.1.2	BB_INLOOP	469
6.69.1.3	BB_OPTIONAL	469
6.69.1.4	BBID_END	469
6.69.1.5	BBID_START	469
6.69.1.6	CUC_MAX_STACK	469
6.69.1.7	CUC_WIDTH_ITERATIONS	469
6.69.1.8	cucdebug	469
6.69.1.9	FLAG_REG	469
6.69.1.10	INSN	469
6.69.1.11	IT_BBEND	469
6.69.1.12	IT_BBSTART	469
6.69.1.13	IT_BRANCH	469
6.69.1.14	IT_COND	469
6.69.1.15	IT_CUT	469
6.69.1.16	IT_FLAG1	469
6.69.1.17	IT_FLAG2	469
6.69.1.18	IT_INDELAY	469
6.69.1.19	IT_LATCHED	469
6.69.1.20	IT_MEMADD	469
6.69.1.21	IT_MEMORY	469
6.69.1.22	IT_OUTPUT	469

6.69.1.23	IT_SIGNED	469
6.69.1.24	IT_UNUSED	469
6.69.1.25	IT_VOLATILE	469
6.69.1.26	log	469
6.69.1.27	LRBB_REG	469
6.69.1.28	MAX	469
6.69.1.29	MAX_BB	469
6.69.1.30	MAX_INSNS	469
6.69.1.31	MAX_PREROLL	469
6.69.1.32	MAX_REGS	469
6.69.1.33	MAX_UNROLL	469
6.69.1.34	MIN	469
6.69.1.35	MO_EXACT	469
6.69.1.36	MO_NONE	469
6.69.1.37	MO_STRONG	469
6.69.1.38	MO_WEAK	469
6.69.1.39	MT_BURST	469
6.69.1.40	MT_BURSTE	469
6.69.1.41	MT_CALL	469
6.69.1.42	MT_LOAD	469
6.69.1.43	MT_SIGNED	469
6.69.1.44	MT_STORE	469
6.69.1.45	MT_WIDTH	469
6.69.1.46	OPT_BB	469
6.69.1.47	OPT_CONST	469
6.69.1.48	OPT_DEST	469
6.69.1.49	OPT_JUMP	469
6.69.1.50	OPT_LRBB	469
6.69.1.51	OPT_NONE	469
6.69.1.52	OPT_REF	469
6.69.1.53	OPT_REGISTER	469
6.69.1.54	REF	469
6.69.1.55	REF_BB	469
6.69.1.56	REF_I	469
6.69.2	Typedef Documentation	469
6.69.2.1	cuc_func	469

6.69.2.2	cuc_shared_list	469
6.69.2.3	dep_list	469
6.69.3	Function Documentation	469
6.69.3.1	add_data_dep	469
6.69.3.2	add_dep	470
6.69.3.3	add_latches	470
6.69.3.4	add_memory_dep	470
6.69.3.5	analyse_timings	470
6.69.3.6	build_bb	470
6.69.3.7	clean_deps	471
6.69.3.8	cse	471
6.69.3.9	csm	471
6.69.3.10	csm_gen	471
6.69.3.11	cuc_check	471
6.69.3.12	cuc_load	472
6.69.3.13	detect_bb	472
6.69.3.14	detect_max_values	472
6.69.3.15	dispose_list	473
6.69.3.16	dup_func	473
6.69.3.17	expand_bb	473
6.69.3.18	free_func	473
6.69.3.19	generate_bb_seq	473
6.69.3.20	insert_conditional_facts	474
6.69.3.21	insert_insns	474
6.69.3.22	main_cuc	476
6.69.3.23	mark_cut	477
6.69.3.24	negate_conditional	477
6.69.3.25	optimize_bb	477
6.69.3.26	optimize_cmovs	477
6.69.3.27	optimize_tree	478
6.69.3.28	preunroll_loop	478
6.69.3.29	print_bb_num	478
6.69.3.30	print_cuc_bb	478
6.69.3.31	print_cuc_insns	479
6.69.3.32	print_insns	479
6.69.3.33	recalc_cnts	479

6.69.3.34	reg_cuc_sec	479
6.69.3.35	reg_dep	480
6.69.3.36	remove_dead	480
6.69.3.37	remove_dead_bb	480
6.69.3.38	remove_nops	480
6.69.3.39	remove_trivial_regs	480
6.69.3.40	schedule_memory	481
6.69.3.41	set_io	481
6.69.4	Variable Documentation	481
6.69.4.1	caller_saved	481
6.69.4.2	cuc_debug	481
6.69.4.3	flog	481
6.69.4.4	insn	481
6.69.4.5	num_insn	481
6.69.4.6	reloc	481
6.70	cuc/insn.c File Reference	482
6.70.1	Function Documentation	483
6.70.1.1	add_data_dep	483
6.70.1.2	add_dep	483
6.70.1.3	add_latches	483
6.70.1.4	apply_edge_condition	484
6.70.1.5	change_insn_type	484
6.70.1.6	cmov_needed	484
6.70.1.7	count_cmovs	484
6.70.1.8	cse	484
6.70.1.9	csm	484
6.70.1.10	csm_gen	484
6.70.1.11	cuc_insn_name	485
6.70.1.12	dispose_list	485
6.70.1.13	insert_insns	485
6.70.1.14	insn_uses	485
6.70.1.15	optimize_cmov_more	485
6.70.1.16	optimize_cmovs	485
6.70.1.17	optimize_tree	486
6.70.1.18	print_insns	486
6.70.1.19	print_shared	486

6.70.1.20	remove_dead	486
6.70.1.21	remove_nops	487
6.70.1.22	remove_trivial_regs	487
6.70.1.23	search_csm	487
6.70.1.24	set_io	487
6.70.1.25	unmark_tree	487
6.70.2	Variable Documentation	487
6.70.2.1	iteration	487
6.70.2.2	known	487
6.70.2.3	main_list	487
6.70.2.4	tmp_op	487
6.70.2.5	tmp_opt	487
6.71	cuc/insn.h File Reference	488
6.71.1	Define Documentation	490
6.71.1.1	II_ADD	490
6.71.1.2	II_AND	490
6.71.1.3	II_BF	490
6.71.1.4	II_CALL	490
6.71.1.5	II_CMOV	490
6.71.1.6	II_IS_LOAD	490
6.71.1.7	II_IS_STORE	490
6.71.1.8	II_LAST	490
6.71.1.9	II_LB	490
6.71.1.10	II_LH	490
6.71.1.11	II_LRBB	490
6.71.1.12	II_LW	490
6.71.1.13	II_MASK	490
6.71.1.14	II_MEM	490
6.71.1.15	II_MEM_WIDTH	490
6.71.1.16	II_MUL	492
6.71.1.17	II_NOP	492
6.71.1.18	II_OR	492
6.71.1.19	II_REG	492
6.71.1.20	II_SB	492
6.71.1.21	II_SFEQ	492
6.71.1.22	II_SFGE	492

6.71.1.23	II_SFGT	492
6.71.1.24	II_SFLE	492
6.71.1.25	II_SFLT	492
6.71.1.26	II_SFNE	492
6.71.1.27	II_SH	492
6.71.1.28	II_SIGNED	492
6.71.1.29	II_SLL	492
6.71.1.30	II_SRA	492
6.71.1.31	II_SRL	492
6.71.1.32	II_SUB	492
6.71.1.33	II_SW	492
6.71.1.34	II_XOR	492
6.71.2	Function Documentation	492
6.71.2.1	change_insn_type	492
6.71.2.2	cuc_insn_name	492
6.71.2.3	ii_size	492
6.71.2.4	insn_size	492
6.71.2.5	insn_time	492
6.71.2.6	load_timing_table	492
6.71.2.7	print_shared	492
6.71.3	Variable Documentation	493
6.71.3.1	known	493
6.72	cuc/load.c File Reference	494
6.72.1	Function Documentation	495
6.72.1.1	build_insn	495
6.72.1.2	cuc_load	496
6.72.1.3	detect_locals	496
6.72.1.4	expand_branch	496
6.72.1.5	expand_calls	496
6.72.1.6	expand_memory	497
6.72.1.7	expand_signed	497
6.72.1.8	negate_conditional	497
6.72.1.9	print_cuc_insns	497
6.72.1.10	remove_dslots	497
6.72.1.11	xchg_insn	498
6.72.2	Variable Documentation	498

6.72.2.1	conv	498
6.72.2.2	insn	498
6.72.2.3	num_insn	498
6.72.2.4	reloc	498
6.73	cuc/memory.c File Reference	499
6.73.1	Function Documentation	500
6.73.1.1	add_memory_dep	500
6.73.1.2	check_memory_conflict	500
6.73.1.3	clean_deps	500
6.73.1.4	join_transfers	500
6.73.1.5	mem_ordering_cmp	500
6.73.1.6	same_transfers	500
6.73.1.7	schedule_memory	501
6.74	peripheral/memory.c File Reference	502
6.74.1	Function Documentation	503
6.74.1.1	mem_reset	503
6.74.1.2	memory_baseaddr	503
6.74.1.3	memory_ce	503
6.74.1.4	memory_delayr	503
6.74.1.5	memory_delayw	503
6.74.1.6	memory_log	503
6.74.1.7	memory_mc	503
6.74.1.8	memory_name	503
6.74.1.9	memory_pattern	503
6.74.1.10	memory_random_seed	504
6.74.1.11	memory_sec_end	504
6.74.1.12	memory_sec_start	504
6.74.1.13	memory_size	504
6.74.1.14	memory_type	504
6.74.1.15	reg_memory_sec	505
6.74.1.16	simmem_read16	506
6.74.1.17	simmem_read32	506
6.74.1.18	simmem_read8	506
6.74.1.19	simmem_read_zero16	506
6.74.1.20	simmem_read_zero32	506
6.74.1.21	simmem_read_zero8	506

6.74.1.22	<code>simmem_write16</code>	506
6.74.1.23	<code>simmem_write32</code>	506
6.74.1.24	<code>simmem_write8</code>	506
6.74.1.25	<code>simmem_write_null16</code>	506
6.74.1.26	<code>simmem_write_null32</code>	506
6.74.1.27	<code>simmem_write_null8</code>	506
6.75	<code>cuc/timings.c</code> File Reference	507
6.75.1	Function Documentation	508
6.75.1.1	<code>analyse_timings</code>	508
6.75.1.2	<code>bb_size</code>	508
6.75.1.3	<code>cut_tree</code>	508
6.75.1.4	<code>ii_size</code>	509
6.75.1.5	<code>insn_size</code>	509
6.75.1.6	<code>insn_time</code>	509
6.75.1.7	<code>load_timing_table</code>	509
6.75.1.8	<code>mark_cut</code>	509
6.75.1.9	<code>max_delay</code>	509
6.75.1.10	<code>memory_delay</code>	509
6.75.1.11	<code>new_bb_cycles</code>	509
6.75.1.12	<code>recalc_cnts</code>	509
6.75.2	Variable Documentation	509
6.75.2.1	<code>max_bb_delay</code>	509
6.75.2.2	<code>timing_table</code>	509
6.76	<code>cuc/verilog.c</code> File Reference	510
6.76.1	Define Documentation	511
6.76.1.1	<code>GEN</code>	511
6.76.2	Function Documentation	511
6.76.2.1	<code>branch_index</code>	511
6.76.2.2	<code>find_lsc_index</code>	511
6.76.2.3	<code>func_index</code>	511
6.76.2.4	<code>generate_main</code>	511
6.76.2.5	<code>output_verilog</code>	511
6.76.2.6	<code>print_deps</code>	512
6.76.2.7	<code>print_insn_v</code>	512
6.76.2.8	<code>print_op_v</code>	512
6.76.2.9	<code>print_turn_off_dep</code>	512

6.77	cuc/verilog.h File Reference	513
6.77.1	Function Documentation	513
6.77.1.1	generate_main	513
6.77.1.2	output_verilog	513
6.78	debug/debug-unit.c File Reference	514
6.78.1	Define Documentation	515
6.78.1.1	RISCOP_RESET	515
6.78.1.2	RISCOP_STALL	515
6.78.2	Enumeration Type Documentation	516
6.78.2.1	development_interface_address_space	516
6.78.3	Function Documentation	516
6.78.3.1	calculate_watchpoints	516
6.78.3.2	check_debug_unit	516
6.78.3.3	debug_enabled	517
6.78.3.4	debug_gdb_enabled	517
6.78.3.5	debug_get_mem	517
6.78.3.6	debug_get_register	518
6.78.3.7	debug_ignore_exception	518
6.78.3.8	debug_rsp_enabled	518
6.78.3.9	debug_rsp_port	519
6.78.3.10	debug_server_port	519
6.78.3.11	debug_set_chain	519
6.78.3.12	debug_set_mem	520
6.78.3.13	debug_set_register	520
6.78.3.14	debug_vapi_id	521
6.78.3.15	du_reset	521
6.78.3.16	get_devint_reg	521
6.78.3.17	get_devint_reg	521
6.78.3.18	reg_debug_sec	521
6.78.3.19	set_devint_reg	522
6.78.3.20	set_devint_reg	523
6.78.3.21	set_stall_state	523
6.78.4	Variable Documentation	523
6.78.4.1	current_scan_chain	523
6.78.4.2	development	523
6.78.4.3	in_reset	523

6.79	debug/debug-unit.h File Reference	524
6.79.1	Enumeration Type Documentation	524
6.79.1.1	debug_scan_chain_ids	524
6.79.1.2	debug_unit_action	525
6.79.2	Function Documentation	525
6.79.2.1	check_debug_unit	525
6.79.2.2	debug_get_register	525
6.79.2.3	debug_ignore_exception	526
6.79.2.4	debug_set_chain	526
6.79.2.5	debug_set_register	526
6.79.2.6	du_reset	527
6.79.2.7	reg_debug_sec	527
6.79.2.8	set_stall_state	528
6.80	debug/gdb.h File Reference	529
6.80.1	Enumeration Type Documentation	529
6.80.1.1	or1k_jtag_errors	529
6.80.1.2	or1k_jtag_proxy_protocol_commands	530
6.81	debug/gdbcomm.c File Reference	531
6.81.1	Function Documentation	532
6.81.1.1	block_jtag	532
6.81.1.2	gdb_read	532
6.81.1.3	gdb_request	532
6.81.1.4	gdb_write	533
6.81.1.5	gdbcomm_init	533
6.81.1.6	get_server_socket	533
6.81.1.7	handle_server_socket	533
6.81.1.8	jtag_request	534
6.81.1.9	protocol_clean	534
6.81.2	Variable Documentation	534
6.81.2.1	gdb_fd	534
6.81.2.2	server_fd	534
6.81.2.3	server_ip	534
6.81.2.4	server_port	534
6.81.2.5	tcp_level	534
6.82	debug/gdbcomm.h File Reference	535
6.82.1	Enumeration Type Documentation	535

6.82.1.1	boolean	535
6.82.2	Function Documentation	535
6.82.2.1	block_jtag	535
6.82.2.2	gdbcomm_init	535
6.82.2.3	handle_server_socket	536
6.83	debug/rsp-server.c File Reference	537
6.83.1	Define Documentation	539
6.83.1.1	GDB_BUF_MAX	539
6.83.1.2	MP_HASH_SIZE	539
6.83.1.3	NPC_REGNUM	540
6.83.1.4	NUM_REGS	540
6.83.1.5	OR1K_TRAP_INSTR	540
6.83.1.6	OR1KSIM_RSP_PROTOCOL	540
6.83.1.7	OR1KSIM_RSP_SERVICE	540
6.83.1.8	PPC_REGNUM	540
6.83.1.9	RSP_TRACE	540
6.83.1.10	SR_REGNUM	540
6.83.2	Enumeration Type Documentation	540
6.83.2.1	mp_type	540
6.83.2.2	target_signal	541
6.83.3	Function Documentation	541
6.83.3.1	ascii2hex	541
6.83.3.2	get_packet	541
6.83.3.3	get_rsp_char	542
6.83.3.4	handle_rsp	542
6.83.3.5	hex	543
6.83.3.6	hex2ascii	544
6.83.3.7	hex2reg	544
6.83.3.8	mp_hash_add	544
6.83.3.9	mp_hash_delete	545
6.83.3.10	mp_hash_init	545
6.83.3.11	mp_hash_lookup	545
6.83.3.12	put_packet	546
6.83.3.13	put_rsp_char	546
6.83.3.14	put_str_packet	546
6.83.3.15	reg2hex	547

6.83.3.16	<code>rsp_client_close</code>	547
6.83.3.17	<code>rsp_client_request</code>	547
6.83.3.18	<code>rsp_command</code>	548
6.83.3.19	<code>rsp_continue</code>	549
6.83.3.20	<code>rsp_continue_generic</code>	550
6.83.3.21	<code>rsp_continue_with_signal</code>	550
6.83.3.22	<code>rsp_exception</code>	550
6.83.3.23	<code>rsp_init</code>	550
6.83.3.24	<code>rsp_insert_matchpoint</code>	551
6.83.3.25	<code>rsp_query</code>	551
6.83.3.26	<code>rsp_read_all_regs</code>	552
6.83.3.27	<code>rsp_read_mem</code>	552
6.83.3.28	<code>rsp_read_reg</code>	553
6.83.3.29	<code>rsp_remove_matchpoint</code>	553
6.83.3.30	<code>rsp_report_exception</code>	554
6.83.3.31	<code>rsp_restart</code>	554
6.83.3.32	<code>rsp_server_close</code>	554
6.83.3.33	<code>rsp_server_request</code>	554
6.83.3.34	<code>rsp_set</code>	555
6.83.3.35	<code>rsp_step</code>	555
6.83.3.36	<code>rsp_step_generic</code>	555
6.83.3.37	<code>rsp_step_with_signal</code>	556
6.83.3.38	<code>rsp_unescape</code>	556
6.83.3.39	<code>rsp_vpkt</code>	556
6.83.3.40	<code>rsp_write_all_regs</code>	557
6.83.3.41	<code>rsp_write_mem</code>	557
6.83.3.42	<code>rsp_write_mem_bin</code>	558
6.83.3.43	<code>rsp_write_reg</code>	558
6.83.3.44	<code>set_npc</code>	558
6.83.4	Variable Documentation	559
6.83.4.1	<code>client_fd</code>	559
6.83.4.2	<code>client_waiting</code>	559
6.83.4.3	<code>hexchars</code>	559
6.83.4.4	<code>mp_hash</code>	559
6.83.4.5	<code>proto_num</code>	559
6.83.4.6	<code>rsp</code>	559

6.83.4.7	server_fd	559
6.83.4.8	sigval	559
6.83.4.9	start_addr	559
6.84	debug/rsp-server.h File Reference	560
6.84.1	Function Documentation	560
6.84.1.1	handle_rsp	560
6.84.1.2	rsp_exception	561
6.84.1.3	rsp_init	562
6.85	libtoplevel.c File Reference	563
6.85.1	Function Documentation	564
6.85.1.1	internal_or1ksim_time	564
6.85.1.2	or1ksim_clock_rate	564
6.85.1.3	or1ksim_get_time_period	564
6.85.1.4	or1ksim_init	564
6.85.1.5	or1ksim_interrupt	567
6.85.1.6	or1ksim_is_le	567
6.85.1.7	or1ksim_reset_duration	567
6.85.1.8	or1ksim_run	567
6.85.1.9	or1ksim_set_time_point	568
6.86	mainpage File Reference	569
6.86.1	Detailed Description	569
6.87	mmu/dmmu.c File Reference	570
6.87.1	Function Documentation	571
6.87.1.1	dmmu_enabled	571
6.87.1.2	dmmu_end_sec	571
6.87.1.3	dmmu_entsize	571
6.87.1.4	dmmu_find_tlbmr	572
6.87.1.5	dmmu_hitdelay	572
6.87.1.6	dmmu_missdelay	572
6.87.1.7	dmmu_nsets	572
6.87.1.8	dmmu_nways	572
6.87.1.9	dmmu_pagesize	572
6.87.1.10	dmmu_start_sec	573
6.87.1.11	dmmu_translate	573
6.87.1.12	dmmu_ustates	573
6.87.1.13	dtlb_status	574

6.87.1.14	peek_into_dtlb	574
6.87.1.15	reg_dmmu_sec	574
6.87.2	Variable Documentation	575
6.87.2.1	dmmu_state	575
6.88	mmu/dmmu.h File Reference	576
6.88.1	Define Documentation	577
6.88.1.1	DADDR_PAGE	577
6.88.2	Function Documentation	577
6.88.2.1	dmmu_simulate_tlb	577
6.88.2.2	dmmu_translate	577
6.88.2.3	peek_into_dtlb	577
6.88.2.4	reg_dmmu_sec	578
6.88.3	Variable Documentation	578
6.88.3.1	dmmu_state	578
6.89	mmu/immu.c File Reference	579
6.89.1	Function Documentation	580
6.89.1.1	immu_enabled	580
6.89.1.2	immu_end_sec	580
6.89.1.3	immu_entrysize	580
6.89.1.4	immu_find_tlbmr	581
6.89.1.5	immu_hitdelay	581
6.89.1.6	immu_missdelay	581
6.89.1.7	immu_nsets	581
6.89.1.8	immu_nways	581
6.89.1.9	immu_pagesize	581
6.89.1.10	immu_start_sec	582
6.89.1.11	immu_translate	582
6.89.1.12	immu_ustates	582
6.89.1.13	itlb_status	583
6.89.1.14	peek_into_itlb	583
6.89.1.15	reg_immu_sec	583
6.89.2	Variable Documentation	583
6.89.2.1	immu_state	583
6.90	mmu/immu.h File Reference	584
6.90.1	Define Documentation	585
6.90.1.1	IADDR_PAGE	585

6.90.2	Function Documentation	585
6.90.2.1	immu_simulate_tlb	585
6.90.2.2	immu_translate	585
6.90.2.3	peek_into_itlb	585
6.90.2.4	reg_immu_sec	586
6.90.3	Variable Documentation	586
6.90.3.1	immu_state	586
6.91	mprofiler.c File Reference	587
6.91.1	Define Documentation	588
6.91.1.1	BUF_SIZE	588
6.91.1.2	HASH_FUNC	588
6.91.1.3	HASH_SIZE	588
6.91.1.4	MODE_ACCESS	588
6.91.1.5	MODE_DETAIL	588
6.91.1.6	MODE_PRETTY	588
6.91.1.7	MODE_WIDTH	588
6.91.2	Function Documentation	588
6.91.2.1	hash_add	588
6.91.2.2	hash_get	588
6.91.2.3	init	588
6.91.2.4	main_mprofiler	588
6.91.2.5	nbits	589
6.91.2.6	printout	589
6.91.2.7	read_file	589
6.91.3	Variable Documentation	589
6.91.3.1	end_addr	589
6.91.3.2	fprof	589
6.91.3.3	group_bits	589
6.91.3.4	hash	589
6.91.3.5	start_addr	590
6.92	mprofiler.h File Reference	591
6.92.1	Function Documentation	591
6.92.1.1	main_mprofiler	591
6.93	or1ksim.h File Reference	592
6.93.1	Enumeration Type Documentation	592
6.93.1.1	or1ksim_rc	592

6.93.2	Function Documentation	592
6.93.2.1	or1ksim_clock_rate	592
6.93.2.2	or1ksim_get_time_period	593
6.93.2.3	or1ksim_init	593
6.93.2.4	or1ksim_interrupt	593
6.93.2.5	or1ksim_is_le	593
6.93.2.6	or1ksim_reset_duration	593
6.93.2.7	or1ksim_run	594
6.93.2.8	or1ksim_set_time_point	594
6.94	peripheral/16450.c File Reference	595
6.94.1	Define Documentation	598
6.94.1.1	MAX_SKEW	598
6.94.1.2	MIN	598
6.94.1.3	UART_ADDR_SPACE	598
6.94.1.4	UART_BREAK_COUNT	598
6.94.1.5	UART_CHAR_TIMEOUT	598
6.94.1.6	UART_CLOCK_DIVIDER	598
6.94.1.7	UART_DLH	598
6.94.1.8	UART_DLL	598
6.94.1.9	UART_FCR	598
6.94.1.10	UART_FCR_FIE	598
6.94.1.11	UART_FCR_RRXFI	598
6.94.1.12	UART_FCR_RTXFI	598
6.94.1.13	UART_FGETC_SLOWDOWN	598
6.94.1.14	UART_FIFO_TRIGGER	598
6.94.1.15	UART_IER	601
6.94.1.16	UART_IER_MSI	601
6.94.1.17	UART_IER_RDI	601
6.94.1.18	UART_IER_RLSI	601
6.94.1.19	UART_IER_THRI	601
6.94.1.20	UART_IIR	601
6.94.1.21	UART_IIR_CTI	601
6.94.1.22	UART_IIR_ID	601
6.94.1.23	UART_IIR_MSI	601
6.94.1.24	UART_IIR_NO_INT	601
6.94.1.25	UART_IIR_RDI	601

6.94.1.26	UART_IIR_RLSI	601
6.94.1.27	UART_IIR_THRI	601
6.94.1.28	UART_LCR	601
6.94.1.29	UART_LCR_DLAB	601
6.94.1.30	UART_LCR_EPAR	601
6.94.1.31	UART_LCR_PARITY	601
6.94.1.32	UART_LCR_RESET	601
6.94.1.33	UART_LCR_SBC	601
6.94.1.34	UART_LCR_SPAR	601
6.94.1.35	UART_LCR_STOP	601
6.94.1.36	UART_LCR_WLEN5	601
6.94.1.37	UART_LCR_WLEN6	601
6.94.1.38	UART_LCR_WLEN7	601
6.94.1.39	UART_LCR_WLEN8	601
6.94.1.40	UART_LSR	601
6.94.1.41	UART_LSR_BREAK	601
6.94.1.42	UART_LSR_FRAME	601
6.94.1.43	UART_LSR_OVRRUN	601
6.94.1.44	UART_LSR_PARITY	601
6.94.1.45	UART_LSR_RDRDY	601
6.94.1.46	UART_LSR_RXERR	601
6.94.1.47	UART_LSR_TXBUFE	601
6.94.1.48	UART_LSR_TXSERE	601
6.94.1.49	UART_MAX_FIFO_LEN	601
6.94.1.50	UART_MCR	602
6.94.1.51	UART_MCR_AUX1	602
6.94.1.52	UART_MCR_AUX2	602
6.94.1.53	UART_MCR_DTR	602
6.94.1.54	UART_MCR_LOOP	602
6.94.1.55	UART_MCR_RTS	602
6.94.1.56	UART_MSR	602
6.94.1.57	UART_MSR_CTS	602
6.94.1.58	UART_MSR_DCD	602
6.94.1.59	UART_MSR_DCTS	602
6.94.1.60	UART_MSR_DDCD	602
6.94.1.61	UART_MSR_DDSR	602

6.94.1.62	UART_MSR_DSR	602
6.94.1.63	UART_MSR_RI	602
6.94.1.64	UART_MSR_TERI	602
6.94.1.65	UART_RXBUF	602
6.94.1.66	UART_SCR	602
6.94.1.67	UART_TXBUF	602
6.94.1.68	UART_VALID_FCR	602
6.94.1.69	UART_VALID_IER	602
6.94.1.70	UART_VALID_IIR	602
6.94.1.71	UART_VALID_LCR	602
6.94.1.72	UART_VALID_LSR	602
6.94.1.73	UART_VALID_MCR	602
6.94.1.74	UART_VALID_MSR	602
6.94.1.75	UART_VAPI_BUF_LEN	602
6.94.2	Function Documentation	604
6.94.2.1	char_clks	604
6.94.2.2	reg_uart_sec	604
6.94.2.3	send_char	605
6.94.2.4	uart_16550	605
6.94.2.5	uart_add_char	605
6.94.2.6	uart_baseaddr	605
6.94.2.7	uart_channel	605
6.94.2.8	uart_char_clock	606
6.94.2.9	uart_check_char	606
6.94.2.10	uart_check_rdi	606
6.94.2.11	uart_check_rlsi	607
6.94.2.12	uart_clear_int	607
6.94.2.13	uart_enabled	608
6.94.2.14	uart_int_cti	608
6.94.2.15	uart_int_msi	608
6.94.2.16	uart_int_rdi	609
6.94.2.17	uart_int_rlsi	609
6.94.2.18	uart_int_thri	610
6.94.2.19	uart_irq	610
6.94.2.20	uart_jitter	610
6.94.2.21	uart_loopback	610

6.94.2.22	uart_newway	611
6.94.2.23	uart_next_int	611
6.94.2.24	uart_read_byte	611
6.94.2.25	uart_recv_break	611
6.94.2.26	uart_recv_break_start	612
6.94.2.27	uart_recv_break_stop	612
6.94.2.28	uart_recv_char	612
6.94.2.29	uart_reset	613
6.94.2.30	uart_sched_recv_check	613
6.94.2.31	uart_sec_end	614
6.94.2.32	uart_sec_start	614
6.94.2.33	uart_send_break	615
6.94.2.34	uart_status	615
6.94.2.35	uart_tx_send	615
6.94.2.36	uart_vapi_cmd	616
6.94.2.37	uart_vapi_id	616
6.94.2.38	uart_vapi_read	616
6.94.2.39	uart_write_byte	616
6.95	peripheral/16450.h File Reference	617
6.95.1	Function Documentation	619
6.95.1.1	reg_uart_sec	619
6.95.1.2	uart_reset	620
6.95.1.3	uart_status	620
6.96	peripheral/atacmd.h File Reference	621
6.96.1	Define Documentation	624
6.96.1.1	CFA_DISABLE_8BIT_PIO_TRANSFER_MODE	624
6.96.1.2	CFA_DISABLE_POWER_MODE1	624
6.96.1.3	CFA_ENABLE_8BIT_PIO_TRANSFER_MODE	624
6.96.1.4	CFA_ENABLE_POWER_MODE1	624
6.96.1.5	CFA_ERASE_SECTORS	624
6.96.1.6	CFA_REQUEST_EXTENDED_ERROR_CODE	624
6.96.1.7	CFA_TRANSLATE_SECTOR	624
6.96.1.8	CFA_WRITE_MULTIPLE_WITHOUT_ERASE	624
6.96.1.9	CFA_WRITE_SECTORS_WITHOUT_ERASE	624
6.96.1.10	CHECK_POWER_MODE	624
6.96.1.11	DEVICE_RESET	624

6.96.1.12	DISABLE_ADVANCED_POWER_MANAGEMENT	624
6.96.1.13	DISABLE_MEDIA_STATUS_NOTIFICATION	624
6.96.1.14	DISABLE_POWERUP_IN_STANDBY_FEATURE_SET	624
6.96.1.15	DISABLE_READ_LOOKAHEAD	624
6.96.1.16	DISABLE_RELEASE_INTERRUPT	624
6.96.1.17	DISABLE_REVERTING_TO_POWERON_DEFAULTS	624
6.96.1.18	DISABLE_SERVICE_INTERRUPT	624
6.96.1.19	DISABLE_WRITE_CACHE	624
6.96.1.20	DOWNLOAD_MICROCODE	624
6.96.1.21	ENABLE_ADVANCED_POWER_MANAGEMENT	624
6.96.1.22	ENABLE_MEDIA_STATUS_NOTIFICATION	624
6.96.1.23	ENABLE_POWERUP_IN_STANDBY_FEATURE_SET	624
6.96.1.24	ENABLE_READ_LOOKAHEAD_FEATURE	624
6.96.1.25	ENABLE_RELEASE_INTERRUPT	624
6.96.1.26	ENABLE_REVERTING_TO_POWERON_DEFAULTS	624
6.96.1.27	ENABLE_SERVICE_INTERRUPT	624
6.96.1.28	ENABLE_WRITE_CACHE	624
6.96.1.29	EXECUTE_DEVICE_DIAGNOSTICS	624
6.96.1.30	FLUSH_CACHE	624
6.96.1.31	GET_MEDIA_STATUS	624
6.96.1.32	IDENTIFY_DEVICE	624
6.96.1.33	IDENTIFY_PACKET_DEVICE	624
6.96.1.34	IDLE	624
6.96.1.35	IDLE_IMMEDIATE	624
6.96.1.36	INITIALIZE_DEVICE_PARAMETERS	624
6.96.1.37	MEDIA_EJECT	624
6.96.1.38	MEDIA_LOCK	624
6.96.1.39	MEDIA_UNLOCK	624
6.96.1.40	NOP	624
6.96.1.41	PACKET	624
6.96.1.42	POWERUP_IN_STANDBY_FEATURE_SET_SPINUP	624
6.96.1.43	READ_BUFFER	624
6.96.1.44	READ_DMA	624
6.96.1.45	READ_DMA_QUEUED	624
6.96.1.46	READ_MULTIPLE	624
6.96.1.47	READ_NATIVE_MAX_ADDRESS	624

6.96.1.48 READ_SECTOR	624
6.96.1.49 READ_SECTORS	624
6.96.1.50 READ_VERIFY_SECTOR	624
6.96.1.51 READ_VERIFY_SECTORS	624
6.96.1.52 SECURITY_DISABLE_PASSWORD	624
6.96.1.53 SECURITY_ERASE_PREPARE	624
6.96.1.54 SECURITY_ERASE_UNIT	624
6.96.1.55 SECURITY_FREEZE_LOCK	624
6.96.1.56 SECURITY_SET_PASSWORD	624
6.96.1.57 SECURITY_UNLOCK	624
6.96.1.58 SEEK	624
6.96.1.59 SERVICE	624
6.96.1.60 SET_FEATURES	624
6.96.1.61 SET_MAX	624
6.96.1.62 SET_MAX_ADDRESS	624
6.96.1.63 SET_MAX_FREEZE_LOCK	624
6.96.1.64 SET_MAX_LOCK	624
6.96.1.65 SET_MAX_SET_PASSWORD	624
6.96.1.66 SET_MAX_UNLOCK	624
6.96.1.67 SET_MULTIPLE_MODE	624
6.96.1.68 SET_TRANSFER_MODE_SECTOR_COUNT_REG	624
6.96.1.69 SLEEP	624
6.96.1.70 SMART	624
6.96.1.71 SMART_ATTRIBITE_AUTOSAVE	624
6.96.1.72 SMART_DISABLE_OPERATIONS	624
6.96.1.73 SMART_ENABLE_OPERATIONS	624
6.96.1.74 SMART_EXECUTE_OFFLINE_IMMEDIATE	624
6.96.1.75 SMART_READ_DATA	624
6.96.1.76 SMART_READ_LOG	624
6.96.1.77 SMART_RETURN_STATUS	624
6.96.1.78 SMART_SAVE_ATTRIBUTE_VALUES	624
6.96.1.79 SMART_WRITE_LOG	624
6.96.1.80 STANDBY	624
6.96.1.81 STANDBY_IMMEDIATE	624
6.96.1.82 WRITE_BUFFER	624
6.96.1.83 WRITE_DMA	624

6.96.1.84	WRITE_DMA_QUEUED	624
6.96.1.85	WRITE_MULTIPLE	624
6.96.1.86	WRITE_SECTOR	624
6.96.1.87	WRITE_SECTORS	624
6.97	peripheral/atadevice-cmdi.c File Reference	625
6.97.1	Function Documentation	626
6.97.1.1	ata_calc_lba	626
6.97.1.2	ata_cmd_complete	626
6.97.1.3	ata_device_execute_cmd	626
6.97.1.4	ata_device_reset_cmd	626
6.97.1.5	ata_execute_device_diagnostics_cmd	627
6.97.1.6	ata_identify_device_cmd	627
6.97.1.7	ata_initialize_device_parameters_cmd	627
6.97.1.8	ata_read_native_max_addr	627
6.97.1.9	ata_read_sect	627
6.97.1.10	ata_read_sectors_cmd	628
6.97.1.11	ata_set_device_signature	628
6.97.1.12	ata_set_features	628
6.97.1.13	ata_set_sect	628
6.97.1.14	ata_write_sect	628
6.97.1.15	ata_write_sectors	629
6.98	peripheral/atadevice-cmdi.h File Reference	630
6.98.1	Define Documentation	633
6.98.1.1	BYTES_PER_SECTOR	633
6.98.1.2	MIN_MWDMA_CYCLE_TIME	633
6.98.1.3	MIN_PIO_CYCLE_TIME_IORDY	633
6.98.1.4	MIN_PIO_CYCLE_TIME_NO_IORDY	633
6.98.1.5	QUEUE_DEPTH	633
6.98.1.6	RECOMMENDED_MWDMA_CYCLE_TIME	633
6.98.1.7	SET_FEATURES_REQUIRED_AFTER_POWER_UP	633
6.98.1.8	SUPPORT_APM	633
6.98.1.9	SUPPORT_CFA	633
6.98.1.10	SUPPORT_DEVICE_RESET_CMD	633
6.98.1.11	SUPPORT_DOWNLOAD_MICROCODE	633
6.98.1.12	SUPPORT_HOST_PROTECTED_AREA	633
6.98.1.13	SUPPORT_LOOKAHEAD	633

6.98.1.14	SUPPORT_NOP_CMD	633
6.98.1.15	SUPPORT_POWER_MANAGEMENT	633
6.98.1.16	SUPPORT_POWER_UP_IN_STANDBY_MODE	633
6.98.1.17	SUPPORT_READ_BUFFER_CMD	633
6.98.1.18	SUPPORT_READ_WRITE_DMA_QUEUED	633
6.98.1.19	SUPPORT_RELEASE_INTERRUPT	633
6.98.1.20	SUPPORT_REMOVABLE_MEDIA	633
6.98.1.21	SUPPORT_REMOVABLE_MEDIA_NOTIFICATION	633
6.98.1.22	SUPPORT_SECURITY_MODE	633
6.98.1.23	SUPPORT_SERVICE_INTERRUPT	633
6.98.1.24	SUPPORT_SET_MAX	633
6.98.1.25	SUPPORT_SMART	633
6.98.1.26	SUPPORT_WRITE_BUFFER_CMD	633
6.98.1.27	SUPPORT_WRITE_CACHE	633
6.98.2	Function Documentation	633
6.98.2.1	ata_device_execute_cmd	633
6.98.2.2	ata_execute_device_diagnostics_cmd	634
6.99	peripheral/atadevice.c File Reference	635
6.99.1	Function Documentation	636
6.99.1.1	ata_device_do_command_register	636
6.99.1.2	ata_device_do_control_register	636
6.99.1.3	ata_device_hw_reset	636
6.99.1.4	ata_device_init	636
6.99.1.5	ata_device_write	637
6.99.1.6	ata_devices_hw_reset	637
6.99.1.7	ata_devices_init	637
6.99.1.8	ata_devices_read	637
6.99.1.9	ata_devices_write	637
6.99.1.10	open_file	638
6.99.1.11	open_local	638
6.100	peripheral/atadevice.h File Reference	639
6.100.1	Define Documentation	642
6.100.1.1	ATA_ASR	642
6.100.1.2	ATA_CHR	642
6.100.1.3	ATA_CLR	642
6.100.1.4	ATA_CR	642

6.100.1.5	ATA_DA	642
6.100.1.6	ATA_DAR_DS0	642
6.100.1.7	ATA_DAR_DS1	642
6.100.1.8	ATA_DAR_H	642
6.100.1.9	ATA_DAR_WTG	642
6.100.1.10	ATA_DCR	642
6.100.1.11	ATA_DCR_IEN	642
6.100.1.12	ATA_DCR_RST	642
6.100.1.13	ATA_DHR	642
6.100.1.14	ATA_DHR_DEV	642
6.100.1.15	ATA_DHR_H	642
6.100.1.16	ATA_DHR_LBA	642
6.100.1.17	ATA_DR	642
6.100.1.18	ATA_ERR	642
6.100.1.19	ATA_ERR_ABT	642
6.100.1.20	ATA_ERR_AMN	642
6.100.1.21	ATA_ERR_BBK	642
6.100.1.22	ATA_ERR_IDNF	642
6.100.1.23	ATA_ERR_TON	642
6.100.1.24	ATA_ERR_UNC	642
6.100.1.25	ATA_FR	642
6.100.1.26	ATA_SCR	642
6.100.1.27	ATA_SNR	642
6.100.1.28	ATA_SR	642
6.100.1.29	ATA_SR_BSY	642
6.100.1.30	ATA_SR_COR	642
6.100.1.31	ATA_SR_DF	642
6.100.1.32	ATA_SR_DRDY	642
6.100.1.33	ATA_SR_DRQ	642
6.100.1.34	ATA_SR_DSC	642
6.100.1.35	ATA_SR_ERR	642
6.100.1.36	ATA_SR_IDX	642
6.100.1.37	ATA_STATE_HW_RST	642
6.100.1.38	ATA_STATE_IDLE	642
6.100.1.39	ATA_STATE_SW_RST	642
6.100.1.40	TYPE_FILE	642

6.100.1.4	ITYPE_LOCAL	642
6.100.1.42	TYPE_NO_CONNECT	642
6.100.2	Function Documentation	642
6.100.2.1	ata_devices_hw_reset	642
6.100.2.2	ata_devices_init	643
6.100.2.3	ata_devices_read	643
6.100.2.4	ata_devices_write	643
6.101	peripheral/atahost.c File Reference	644
6.101.1	Define Documentation	646
6.101.1.1	DMA_MODE0_TD	646
6.101.1.2	DMA_MODE0_TEOC	646
6.101.1.3	DMA_MODE0_TM	646
6.101.1.4	PIO_MODE0_T1	646
6.101.1.5	PIO_MODE0_T2	646
6.101.1.6	PIO_MODE0_T4	646
6.101.1.7	PIO_MODE0_TEOC	646
6.101.2	Function Documentation	646
6.101.2.1	ata_baseaddr	646
6.101.2.2	ata_dev_id	646
6.101.2.3	ata_dma_mode0_td	646
6.101.2.4	ata_dma_mode0_tecoc	646
6.101.2.5	ata_dma_mode0_tm	646
6.101.2.6	ata_enabled	646
6.101.2.7	ata_enddevice	646
6.101.2.8	ata_file	646
6.101.2.9	ata_firmware	647
6.101.2.10	ata_heads	647
6.101.2.11	ata_int	647
6.101.2.12	ata_irq	647
6.101.2.13	ata_mwdma	647
6.101.2.14	ata_packet	647
6.101.2.15	ata_pio	647
6.101.2.16	ata_pio_mode0_t1	648
6.101.2.17	ata_pio_mode0_t2	648
6.101.2.18	ata_pio_mode0_t4	648
6.101.2.19	ata_pio_mode0_tecoc	648

6.101.2.20	ata_read32	648
6.101.2.21	ata_reset	648
6.101.2.22	ata_rev	648
6.101.2.23	ata_sec_end	649
6.101.2.24	ata_sec_start	649
6.101.2.25	ata_sectors	650
6.101.2.26	ata_size	650
6.101.2.27	ata_start_device	650
6.101.2.28	ata_status	650
6.101.2.29	ata_type	650
6.101.2.30	ata_write32	650
6.101.2.31	reg_ata_sec	652
6.101.3	Variable Documentation	653
6.101.3.1	conf_dev	653
6.102	peripheral/atahost.h File Reference	654
6.102.1	Define Documentation	657
6.102.1.1	ATA_ADDR_SPACE	657
6.102.1.2	ATA_BELECO	657
6.102.1.3	ATA_BELEC1	657
6.102.1.4	ATA_CTRL	657
6.102.1.5	ATA_DEVID	657
6.102.1.6	ata_dma_delay	657
6.102.1.7	ATA_DMA_EN	657
6.102.1.8	ATA_DMA_RD	657
6.102.1.9	ATA_DMA_TIP	657
6.102.1.10	ATA_DMA_WR	657
6.102.1.11	ATA_DMARQ	657
6.102.1.12	ATA_DRBE	657
6.102.1.13	ATA_DTBF	657
6.102.1.14	ATA_DTR0	657
6.102.1.15	ATA_DTR1	657
6.102.1.16	ATA_FTE0	657
6.102.1.17	ATA_FTE1	657
6.102.1.18	ATA_IDE_EN	657
6.102.1.19	ATA_IDEIS	657
6.102.1.20	ATA_IORDY	657

6.102.1.21	ATA_IORDY_FTE0	657
6.102.1.22	ATA_IORDY_FTE1	657
6.102.1.23	ATA_PCTR	657
6.102.1.24	ATA_PFTR0	657
6.102.1.25	ATA_PFTR1	657
6.102.1.26	ata_pio_delay	657
6.102.1.27	ATA_PIO_TIP	657
6.102.1.28	ATA_PWPP	657
6.102.1.29	ATA_PWPPF	657
6.102.1.30	ATA_REVNO	657
6.102.1.31	ATA_RST	657
6.102.1.32	ATA_RXB	657
6.102.1.33	ATA_STAT	657
6.102.1.34	ATA_T1	657
6.102.1.35	ATA_T2	657
6.102.1.36	ATA_T4	657
6.102.1.37	ATA_TD	657
6.102.1.38	ATA_TEOC	657
6.102.1.39	ATA_TM	657
6.102.1.40	ATA_TXB	657
6.102.1.4	lis_ata_hostadr	657
6.102.2	Function Documentation	657
6.102.2.1	ata_int	657
6.102.2.2	reg_ata_sec	659
6.103	peripheral/channels/channel.c File Reference	660
6.103.1	Function Documentation	661
6.103.1.1	channel_close	661
6.103.1.2	channel_init	661
6.103.1.3	channel_open	661
6.103.1.4	channel_read	661
6.103.1.5	channel_write	661
6.103.1.6	find_channel_factory	661
6.103.2	Variable Documentation	661
6.103.2.1	head	661
6.103.2.2	preloaded	661
6.104	peripheral/channels/channel.h File Reference	662

6.104.1 Function Documentation	662
6.104.1.1 channel_close	662
6.104.1.2 channel_init	662
6.104.1.3 channel_open	662
6.104.1.4 channel_read	662
6.104.1.5 channel_write	662
6.105peripheral/channels/fd.c File Reference	663
6.105.1 Function Documentation	664
6.105.1.1 fd_init	664
6.105.1.2 fd_isok	664
6.105.1.3 fd_read	664
6.105.1.4 fd_status	664
6.105.1.5 fd_status_fd	664
6.105.1.6 fd_write	664
6.105.2 Variable Documentation	664
6.105.2.1 fd_channel_ops	664
6.106peripheral/channels/fd.h File Reference	665
6.106.1 Function Documentation	665
6.106.1.1 fd_read	665
6.106.1.2 fd_write	665
6.106.2 Variable Documentation	665
6.106.2.1 fd_channel_ops	665
6.107peripheral/channels/file.c File Reference	666
6.107.1 Function Documentation	667
6.107.1.1 file_close	667
6.107.1.2 file_free	667
6.107.1.3 file_init	667
6.107.1.4 file_open	667
6.107.2 Variable Documentation	667
6.107.2.1 file_channel_ops	667
6.108peripheral/channels/file.h File Reference	668
6.108.1 Variable Documentation	668
6.108.1.1 file_channel_ops	668
6.109peripheral/channels/tcp.c File Reference	669
6.109.1 Function Documentation	670
6.109.1.1 tcp_init	670

6.109.1.2	tcp_open	670
6.109.1.3	tcp_read	670
6.109.1.4	tcp_write	670
6.109.1.5	wait_for_tcp_connect	670
6.109.2	Variable Documentation	670
6.109.2.1	tcp_channel_ops	670
6.110	peripheral/channels/tcp.h File Reference	671
6.110.1	Variable Documentation	671
6.110.1.1	tcp_channel_ops	671
6.111	peripheral/channels/tty.c File Reference	672
6.111.1	Define Documentation	673
6.111.1.1	DEFAULT_BAUD	673
6.111.1.2	DEFAULT_TTY_DEVICE	673
6.111.2	Function Documentation	673
6.111.2.1	parse_baud	673
6.111.2.2	tty_init	673
6.111.2.3	tty_open	673
6.111.3	Variable Documentation	673
6.111.3.1	baud_table	673
6.111.3.2	name	673
6.111.3.3	tty_channel_ops	673
6.111.3.4	value	673
6.112	peripheral/channels/tty.h File Reference	674
6.112.1	Variable Documentation	674
6.112.1.1	tty_channel_ops	674
6.113	peripheral/channels/xterm.c File Reference	675
6.113.1	Define Documentation	676
6.113.1.1	MAX_XTERM_ARGS	676
6.113.2	Function Documentation	676
6.113.2.1	basename	676
6.113.2.2	xterm_close	676
6.113.2.3	xterm_init	676
6.113.2.4	xterm_open	676
6.113.3	Variable Documentation	676
6.113.3.1	xterm_channel_ops	676
6.114	peripheral/channels/xterm.h File Reference	677

6.114.1 Variable Documentation	677
6.114.1.1 xterm_channel_ops	677
6.115 peripheral/crc32.c File Reference	678
6.115.1 Function Documentation	678
6.115.1.1 crc32	678
6.115.1.2 crc32_close	678
6.115.1.3 crc32_feed_bytes	678
6.115.1.4 crc32_init	678
6.115.2 Variable Documentation	678
6.115.2.1 crc32_table	678
6.116 peripheral/crc32.h File Reference	679
6.116.1 Function Documentation	679
6.116.1.1 crc32	679
6.116.1.2 crc32_close	679
6.116.1.3 crc32_feed_bytes	679
6.116.1.4 crc32_init	679
6.117 peripheral/dma-defs.h File Reference	680
6.117.1 Define Documentation	682
6.117.1.1 DMA_ADDR_SPACE	682
6.117.1.2 DMA_CH_A0	682
6.117.1.3 DMA_CH_A0_ADDR_OFFSET	682
6.117.1.4 DMA_CH_A0_ADDR_WIDTH	682
6.117.1.5 DMA_CH_A1	682
6.117.1.6 DMA_CH_A1_ADDR_OFFSET	682
6.117.1.7 DMA_CH_A1_ADDR_WIDTH	682
6.117.1.8 DMA_CH_AM0	682
6.117.1.9 DMA_CH_AM0_MASK_OFFSET	682
6.117.1.10 DMA_CH_AM0_MASK_WIDTH	682
6.117.1.11 DMA_CH_AM1	682
6.117.1.12 DMA_CH_AM1_MASK_OFFSET	682
6.117.1.13 DMA_CH_AM1_MASK_WIDTH	682
6.117.1.14 DMA_CH_BASE	682
6.117.1.15 DMA_CH_CSR	684
6.117.1.16 DMA_CH_CSR_ARS_OFFSET	684
6.117.1.17 DMA_CH_CSR_BUSY_OFFSET	684
6.117.1.18 DMA_CH_CSR_CH_EN_OFFSET	684

6.117.1.19	DMA_CH_CSR_DONE_OFFSET	684
6.117.1.20	DMA_CH_CSR_DST_SEL_OFFSET	684
6.117.1.21	DMA_CH_CSR_ERR_OFFSET	684
6.117.1.22	DMA_CH_CSR_INC_DST_OFFSET	684
6.117.1.23	DMA_CH_CSR_INC_SRC_OFFSET	684
6.117.1.24	DMA_CH_CSR_INE_CHK_DONE_OFFSET	684
6.117.1.25	DMA_CH_CSR_INE_DONE_OFFSET	684
6.117.1.26	DMA_CH_CSR_INE_ERR_OFFSET	684
6.117.1.27	DMA_CH_CSR_INT_CHUNK_DONE_OFFSET	684
6.117.1.28	DMA_CH_CSR_INT_DONE_OFFSET	684
6.117.1.29	DMA_CH_CSR_INT_ERR_OFFSET	684
6.117.1.30	DMA_CH_CSR_MODE_OFFSET	684
6.117.1.31	DMA_CH_CSR_PRIORITY_OFFSET	684
6.117.1.32	DMA_CH_CSR_PRIORITY_WIDTH	684
6.117.1.33	DMA_CH_CSR_RESERVED_OFFSET	684
6.117.1.34	DMA_CH_CSR_RESERVED_WIDTH	684
6.117.1.35	DMA_CH_CSR_REST_EN_OFFSET	684
6.117.1.36	DMA_CH_CSR_SRC_SEL_OFFSET	684
6.117.1.37	DMA_CH_CSR_STOP_OFFSET	684
6.117.1.38	DMA_CH_CSR_SZ_WB_OFFSET	684
6.117.1.39	DMA_CH_CSR_USE_ED_OFFSET	684
6.117.1.40	DMA_CH_CSR_WRITE_MASK	684
6.117.1.41	DMA_CH_DESC	684
6.117.1.42	DMA_CH_DESC_ADDR_OFFSET	684
6.117.1.43	DMA_CH_DESC_ADDR_WIDTH	684
6.117.1.44	DMA_CH_SIZE	684
6.117.1.45	DMA_CH_SWPTR	686
6.117.1.46	DMA_CH_SWPTR_EN_OFFSET	686
6.117.1.47	DMA_CH_SWPTR_PTR_OFFSET	686
6.117.1.48	DMA_CH_SWPTR_PTR_WIDTH	686
6.117.1.49	DMA_CH_SZ	686
6.117.1.50	DMA_CH_SZ_CHK_SZ_OFFSET	686
6.117.1.51	DMA_CH_SZ_CHK_SZ_WIDTH	686
6.117.1.52	DMA_CH_SZ_TOT_SZ_OFFSET	686
6.117.1.53	DMA_CH_SZ_TOT_SZ_WIDTH	686
6.117.1.54	DMA_CSR	686

6.117.1.55	DMA_CSR_PAUSE_OFFSET	686
6.117.1.56	DMA_DESC_ADR0	686
6.117.1.57	DMA_DESC_ADR1	686
6.117.1.58	DMA_DESC_CSR	686
6.117.1.59	DMA_DESC_CSR_DST_SEL_OFFSET	686
6.117.1.60	DMA_DESC_CSR_EOL_OFFSET	686
6.117.1.61	DMA_DESC_CSR_INC_DST_OFFSET	686
6.117.1.62	DMA_DESC_CSR_INC_SRC_OFFSET	686
6.117.1.63	DMA_DESC_CSR_SRC_SEL_OFFSET	686
6.117.1.64	DMA_DESC_CSR_TOT_SZ_OFFSET	686
6.117.1.65	DMA_DESC_CSR_TOT_SZ_WIDTH	686
6.117.1.66	DMA_DESC_NEXT	686
6.117.1.67	DMA_INT_MSK_A	686
6.117.1.68	DMA_INT_MSK_B	686
6.117.1.69	DMA_INT_SRC_A	686
6.117.1.70	DMA_INT_SRC_B	686
6.117.1.71	DMA_NUM_CHANNELS	686
6.118	peripheral/dma.c File Reference	687
6.118.1	Define Documentation	689
6.118.1.1	CHANNEL_ND_I	689
6.118.2	Function Documentation	689
6.118.2.1	check_dma_ack_o	689
6.118.2.2	clear_dma_nd_i	689
6.118.2.3	clear_dma_req_i	689
6.118.2.4	dma_baseaddr	689
6.118.2.5	dma_channel_clock	689
6.118.2.6	dma_channel_terminate_transfer	690
6.118.2.7	dma_enabled	690
6.118.2.8	dma_init_transfer	690
6.118.2.9	dma_irq	690
6.118.2.10	dma_load_descriptor	690
6.118.2.11	dma_read32	691
6.118.2.12	dma_read_ch_csr	691
6.118.2.13	dma_reset	691
6.118.2.14	dma_sec_end	691
6.118.2.15	dma_sec_start	691

6.118.2.16	dma_status	692
6.118.2.17	dma_vapi_id	692
6.118.2.18	dma_write32	692
6.118.2.19	dma_write_ch_csr	692
6.118.2.20	find_dma_controller_ch	693
6.118.2.21	masked_increase	693
6.118.2.22	reg_dma_sec	693
6.118.2.23	set_dma_nd_i	693
6.118.2.24	set_dma_req_i	693
6.118.3	Variable Documentation	693
6.118.3.1	dmass	693
6.119	peripheral/dma.h File Reference	694
6.119.1	Function Documentation	695
6.119.1.1	check_dma_ack_o	695
6.119.1.2	clear_dma_nd_i	695
6.119.1.3	clear_dma_req_i	695
6.119.1.4	find_dma_controller_ch	695
6.119.1.5	reg_dma_sec	695
6.119.1.6	set_dma_nd_i	695
6.119.1.7	set_dma_req_i	695
6.120	peripheral/eth.c File Reference	696
6.120.1	Define Documentation	702
6.120.1.1	ETH_ADDR_SPACE	702
6.120.1.2	ETH_ALEN	702
6.120.1.3	ETH_BD_BASE	702
6.120.1.4	ETH_BD_COUNT	702
6.120.1.5	ETH_BD_SPACE	702
6.120.1.6	ETH_CMODER_PASSALL_OFFSET	702
6.120.1.7	ETH_CMODER_RXFLOW_OFFSET	702
6.120.1.8	ETH_CMODER_TXFLOW_OFFSET	702
6.120.1.9	ETH_COLLCONF	702
6.120.1.10	ETH_COLLCONF_COLLVALID_OFFSET	702
6.120.1.11	ETH_COLLCONF_COLLVALID_WIDTH	702
6.120.1.12	ETH_COLLCONF_MAXRET_OFFSET	702
6.120.1.13	ETH_COLLCONF_MAXRET_WIDTH	702
6.120.1.14	ETH_CTRLMODER	702

6.120.1.15	ETH_DMA_RX_TX	702
6.120.1.16	ETH_HASH0	702
6.120.1.17	ETH_HASH1	702
6.120.1.18	ETH_INT_MASK	702
6.120.1.19	ETH_INT_MASK_BUSY_M_OFFSET	702
6.120.1.20	ETH_INT_MASK_RXB_M_OFFSET	702
6.120.1.21	ETH_INT_MASK_RXC_M_OFFSET	702
6.120.1.22	ETH_INT_MASK_RXE_M_OFFSET	702
6.120.1.23	ETH_INT_MASK_TXB_M_OFFSET	702
6.120.1.24	ETH_INT_MASK_TXC_M_OFFSET	702
6.120.1.25	ETH_INT_MASK_TXE_M_OFFSET	702
6.120.1.26	ETH_INT_SOURCE	702
6.120.1.27	ETH_INT_SOURCE_BUSY_OFFSET	702
6.120.1.28	ETH_INT_SOURCE_RXB_OFFSET	702
6.120.1.29	ETH_INT_SOURCE_RXC_OFFSET	702
6.120.1.30	ETH_INT_SOURCE_RXE_OFFSET	702
6.120.1.31	ETH_INT_SOURCE_TXB_OFFSET	702
6.120.1.32	ETH_INT_SOURCE_TXC_OFFSET	702
6.120.1.33	ETH_INT_SOURCE_TXE_OFFSET	702
6.120.1.34	ETH_IPGR1	702
6.120.1.35	ETH_IPGR2	702
6.120.1.36	ETH_IPGT	702
6.120.1.37	ETH_MAC_ADDR0	702
6.120.1.38	ETH_MAC_ADDR1	702
6.120.1.39	ETH_MAXPL	702
6.120.1.40	ETH_MIIADDR_FIAD_OFFSET	702
6.120.1.41	ETH_MIIADDR_FIAD_WIDTH	702
6.120.1.42	ETH_MIIADDR_RGAD_OFFSET	702
6.120.1.43	ETH_MIIADDR_RGAD_WIDTH	702
6.120.1.44	ETH_MIIADDRESS	702
6.120.1.45	ETH_MIICOMM_RSTAT_OFFSET	702
6.120.1.46	ETH_MIICOMM_SCANS_OFFSET	702
6.120.1.47	ETH_MIICOMM_WCDATA_OFFSET	702
6.120.1.48	ETH_MIICOMMAND	702
6.120.1.49	ETH_MIIMODER	702
6.120.1.50	ETH_MIIMODER_CLKDIV_OFFSET	702

6.120.1.51	ETH_MIIMODER_CLKDIV_WIDTH	702
6.120.1.52	ETH_MIIMODER_MRST_OFFSET	702
6.120.1.53	ETH_MIIMODER_NOPRE_OFFSET	702
6.120.1.54	ETH_MIIRX_DATA	702
6.120.1.55	ETH_MIISTAT_BUSY_OFFSET	702
6.120.1.56	ETH_MIISTAT_FAIL_OFFSET	702
6.120.1.57	ETH_MIISTAT_NVALID_OFFSET	702
6.120.1.58	ETH_MIISTATUS	702
6.120.1.59	ETH_MIITX_DATA	702
6.120.1.60	ETH_MODER	702
6.120.1.61	ETH_MODER_BRO_OFFSET	702
6.120.1.62	ETH_MODER_CRCEN_OFFSET	702
6.120.1.63	ETH_MODER_DLYCRCEN_OFFSET	702
6.120.1.64	ETH_MODER_DMAEN_OFFSET	702
6.120.1.65	ETH_MODER_EXDFREN_OFFSET	702
6.120.1.66	ETH_MODER_FULLLD_OFFSET	702
6.120.1.67	ETH_MODER_HUGEN_OFFSET	702
6.120.1.68	ETH_MODER_IAM_OFFSET	702
6.120.1.69	ETH_MODER_IFG_OFFSET	702
6.120.1.70	ETH_MODER_LOOPBCK_OFFSET	702
6.120.1.71	ETH_MODER_NOBCKOF_OFFSET	702
6.120.1.72	ETH_MODER_NOPRE_OFFSET	702
6.120.1.73	ETH_MODER_PAD_OFFSET	702
6.120.1.74	ETH_MODER_PRO_OFFSET	702
6.120.1.75	ETH_MODER_RECSMALL_OFFSET	702
6.120.1.76	ETH_MODER_RST_OFFSET	702
6.120.1.77	ETH_MODER_RXEN_OFFSET	702
6.120.1.78	ETH_MODER_TXEN_OFFSET	702
6.120.1.79	ETH_PACKETLEN	702
6.120.1.80	ETH_PACKETLEN_MAXFL_OFFSET	702
6.120.1.81	ETH_PACKETLEN_MAXFL_WIDTH	702
6.120.1.82	ETH_PACKETLEN_MINFL_OFFSET	702
6.120.1.83	ETH_PACKETLEN_MINFL_WIDTH	702
6.120.1.84	ETH_RTX_FILE	702
6.120.1.85	ETH_RTX_SOCK	702
6.120.1.86	ETH_RTX_VAPI	702

6.120.1.87	ETH_RX_BD_COLLISION_OFFSET	702
6.120.1.88	ETH_RX_BD_CRC_OFFSET	702
6.120.1.89	ETH_RX_BD_DRIBBLE_OFFSET	702
6.120.1.90	ETH_RX_BD_INVALID_OFFSET	702
6.120.1.91	ETH_RX_BD_IRQ_OFFSET	702
6.120.1.92	ETH_RX_BD_LENGTH_OFFSET	702
6.120.1.93	ETH_RX_BD_LENGTH_WIDTH	702
6.120.1.94	ETH_RX_BD_MISS_OFFSET	702
6.120.1.95	ETH_RX_BD_READY_OFFSET	702
6.120.1.96	ETH_RX_BD_TOOBIG_OFFSET	702
6.120.1.97	ETH_RX_BD_TOOSHORT_OFFSET	702
6.120.1.98	ETH_RX_BD_UVERRUN_OFFSET	702
6.120.1.99	ETH_RX_BD_WRAP_OFFSET	702
6.120.1.100	ETH_RXSTATE_IDLE	702
6.120.1.101	ETH_RXSTATE_RECV	702
6.120.1.102	ETH_RXSTATE_WAIT4BD	702
6.120.1.103	ETH_RXSTATE_WRITEFIFO	702
6.120.1.104	ETH_TX_BD_COLLISION_OFFSET	702
6.120.1.105	ETH_TX_BD_CRC_OFFSET	702
6.120.1.106	ETH_TX_BD_DEFER_OFFSET	702
6.120.1.107	ETH_TX_BD_IRQ_OFFSET	702
6.120.1.108	ETH_TX_BD_LAST_OFFSET	702
6.120.1.109	ETH_TX_BD_LENGTH_OFFSET	702
6.120.1.110	ETH_TX_BD_LENGTH_WIDTH	702
6.120.1.111	ETH_TX_BD_NO_CARRIER_OFFSET	702
6.120.1.112	ETH_TX_BD_NUM	702
6.120.1.113	ETH_TX_BD_PAD_OFFSET	702
6.120.1.114	ETH_TX_BD_PAUSE_OFFSET	702
6.120.1.115	ETH_TX_BD_READY_OFFSET	702
6.120.1.116	ETH_TX_BD_RETRANSMIT_OFFSET	702
6.120.1.117	ETH_TX_BD_RETRY_OFFSET	702
6.120.1.118	ETH_TX_BD_RETRY_WIDTH	702
6.120.1.119	ETH_TX_BD_UNDERRUN_OFFSET	702
6.120.1.120	ETH_TX_BD_WRAP_OFFSET	702
6.120.1.121	ETH_TXSTATE_IDLE	702
6.120.1.122	ETH_TXSTATE_READFIFO	702

6.120.1.12	BTH_TXSTATE_TRANSMIT	702
6.120.1.12	BTH_TXSTATE_WAIT4BD	702
6.120.1.12	BETHER_ADDR_LEN	702
6.120.1.12	BETHER_CRC_LEN	702
6.120.1.12	BETHER_HDR_LEN	702
6.120.1.12	BETHER_IS_VALID_LEN	702
6.120.1.12	BETHER_MAX_LEN	702
6.120.1.13	BETHER_MIN_LEN	702
6.120.1.13	BETHER_TYPE_LEN	702
6.120.1.13	BETHERMIN	702
6.120.1.13	BETHERMTU	702
6.120.1.13	BETHERTYPE_ARP	702
6.120.1.13	BETHERTYPE_IP	702
6.120.1.13	BETHERTYPE_NTRAILER	702
6.120.1.13	BETHERTYPE_PUP	702
6.120.1.13	BETHERTYPE_REVARP	702
6.120.1.13	BETHERTYPE_TRAIL	702
6.120.2	Enumeration Type Documentation	702
6.120.2.1	"@45	702
6.120.3	Function Documentation	703
6.120.3.1	eth_baseaddr	703
6.120.3.2	eth_controller_rx_clock	703
6.120.3.3	eth_controller_tx_clock	703
6.120.3.4	eth_dma	704
6.120.3.5	eth_enabled	704
6.120.3.6	eth_irq	704
6.120.3.7	eth_read32	704
6.120.3.8	eth_read_rx_file	704
6.120.3.9	eth_reset	704
6.120.3.10	eth_rtx_type	705
6.120.3.11	eth_rx_channel	705
6.120.3.12	eth_rx_next_packet	705
6.120.3.13	eth_rxfile	705
6.120.3.14	eth_sec_end	707
6.120.3.15	eth_sec_start	708
6.120.3.16	eth_skip_rx_file	708

6.120.3.17	eth_sockif	708
6.120.3.18	eth_status	708
6.120.3.19	eth_tx_channel	708
6.120.3.20	eth_txfile	708
6.120.3.21	eth_vapi_id	709
6.120.3.22	eth_vapi_read	709
6.120.3.23	eth_write32	709
6.120.3.24	eth_write_tx_bd_num	709
6.120.3.25	reg_ethernet_sec	709
6.121	peripheral/eth.h File Reference	711
6.121.1	Function Documentation	711
6.121.1.1	reg_ethernet_sec	711
6.122	peripheral/fb.c File Reference	713
6.122.1	Define Documentation	714
6.122.1.1	CAM_SIZEX	714
6.122.1.2	CAM_SIZEY	714
6.122.1.3	CNV16	714
6.122.1.4	CNV32	714
6.122.1.5	FB_BUFADDR	714
6.122.1.6	FB_CAMBUFADDR	714
6.122.1.7	FB_CAMPOSADDR	714
6.122.1.8	FB_CTRL	714
6.122.1.9	FB_PAL	714
6.122.1.10	FB_SIZEX	714
6.122.1.11	FB_SIZEY	714
6.122.1.12	FB_WRAP	714
6.122.1.13	REFRESH_DIVIDER	714
6.122.2	Function Documentation	715
6.122.2.1	change_buf_addr	715
6.122.2.2	fb_baseaddr	715
6.122.2.3	fb_dump_image24	715
6.122.2.4	fb_dump_image8	715
6.122.2.5	fb_enabled	715
6.122.2.6	fb_filename	715
6.122.2.7	fb_job	716
6.122.2.8	fb_read32	716

6.122.2.9 fb_refresh_rate	716
6.122.2.10fb_reset	716
6.122.2.11fb_sec_end	717
6.122.2.12fb_sec_start	717
6.122.2.13fb_write32	717
6.122.2.14reg_fb_sec	718
6.123peripheral/fb.h File Reference	719
6.123.1 Function Documentation	719
6.123.1.1 reg_fb_sec	719
6.124peripheral/fields.h File Reference	720
6.124.1 Define Documentation	720
6.124.1.1 ASSIGN_FLAG	720
6.124.1.2 CLEAR_FLAG	721
6.124.1.3 FIELD_MASK	721
6.124.1.4 FIELD_SHIFT	721
6.124.1.5 FLAG_MASK	721
6.124.1.6 FLAG_SHIFT	721
6.124.1.7 GET_FIELD	721
6.124.1.8 SET_FIELD	721
6.124.1.9 SET_FLAG	721
6.124.1.10TEST_FLAG	721
6.125peripheral/generic.c File Reference	722
6.125.1 Function Documentation	723
6.125.1.1 ext_read	723
6.125.1.2 ext_write	723
6.125.1.3 generic_baseaddr	723
6.125.1.4 generic_byte_enabled	723
6.125.1.5 generic_enabled	723
6.125.1.6 generic_hw_enabled	723
6.125.1.7 generic_name	723
6.125.1.8 generic_read_byte	723
6.125.1.9 generic_read_hw	723
6.125.1.10generic_read_word	724
6.125.1.11generic_reset	724
6.125.1.12generic_sec_end	724
6.125.1.13generic_sec_start	725

6.125.1.14	generic_size	725
6.125.1.15	generic_status	725
6.125.1.16	generic_word_enabled	725
6.125.1.17	generic_write_byte	725
6.125.1.18	generic_write_hw	725
6.125.1.19	generic_write_word	725
6.125.1.20	reg_generic_sec	726
6.126	peripheral/channels/generic.c File Reference	727
6.126.1	Function Documentation	727
6.126.1.1	generic_close	727
6.126.1.2	generic_free	727
6.126.1.3	generic_open	727
6.127	peripheral/generic.h File Reference	728
6.127.1	Function Documentation	728
6.127.1.1	reg_generic_sec	728
6.128	peripheral/channels/generic.h File Reference	729
6.128.1	Function Documentation	729
6.128.1.1	generic_close	729
6.128.1.2	generic_free	729
6.128.1.3	generic_open	729
6.129	peripheral/gpio.c File Reference	730
6.129.1	Define Documentation	732
6.129.1.1	GPIO_ADDR_SPACE	732
6.129.1.2	RGPIO_AUX	732
6.129.1.3	RGPIO_CTRL	732
6.129.1.4	RGPIO_CTRL_ECLK	732
6.129.1.5	RGPIO_CTRL_INTE	732
6.129.1.6	RGPIO_CTRL_INTS	732
6.129.1.7	RGPIO_CTRL_NEC	732
6.129.1.8	RGPIO_IN	732
6.129.1.9	RGPIO_INTE	732
6.129.1.10	RGPIO_INTS	732
6.129.1.11	RGPIO_OE	732
6.129.1.12	RGPIO_OUT	732
6.129.1.13	RGPIO_PTRIG	732
6.129.2	Enumeration Type Documentation	732

6.129.2.1 "@51	732
6.129.3 Function Documentation	733
6.129.3.1 gpio_base_vapi_id	733
6.129.3.2 gpio_baseaddr	733
6.129.3.3 gpio_clock	733
6.129.3.4 gpio_device_clock	733
6.129.3.5 gpio_do_int	734
6.129.3.6 gpio_enabled	734
6.129.3.7 gpio_external_clock	734
6.129.3.8 gpio_irq	735
6.129.3.9 gpio_read32	735
6.129.3.10 gpio_reset	735
6.129.3.11 gpio_sec_end	735
6.129.3.12 gpio_sec_start	736
6.129.3.13 gpio_status	736
6.129.3.14 gpio_vapi_read	736
6.129.3.15 gpio_write32	736
6.129.3.16 reg_gpio_sec	736
6.130 peripheral/gpio.h File Reference	737
6.130.1 Function Documentation	737
6.130.1.1 reg_gpio_sec	737
6.131 peripheral/mc.c File Reference	738
6.131.1 Define Documentation	742
6.131.1.1 MC_ADDR_SPACE	742
6.131.1.2 MC_BA_MASK	742
6.131.1.3 MC_BA_MASK_VALID	742
6.131.1.4 MC_CSC	742
6.131.1.5 MC_CSC_BAS_OFFSET	742
6.131.1.6 MC_CSC_BW_OFFSET	742
6.131.1.7 MC_CSC_BW_WIDTH	742
6.131.1.8 MC_CSC_EN_OFFSET	742
6.131.1.9 MC_CSC_KRO_OFFSET	742
6.131.1.10 MC_CSC_MEMTYPE_ASYNC	742
6.131.1.11 MC_CSC_MEMTYPE_OFFSET	742
6.131.1.12 MC_CSC_MEMTYPE_SDRAM	742
6.131.1.13 MC_CSC_MEMTYPE_SSRAM	742

6.131.1.14	MC_CSC_MEMTYPE_SYNC	742
6.131.1.15	MC_CSC_MEMTYPE_WIDTH	742
6.131.1.16	MC_CSC_MS_OFFSET	742
6.131.1.17	MC_CSC_MS_WIDTH	742
6.131.1.18	MC_CSC_PEN_OFFSET	742
6.131.1.19	MC_CSC_SEL_OFFSET	742
6.131.1.20	MC_CSC_SEL_WIDTH	742
6.131.1.21	MC_CSC_VALID	742
6.131.1.22	MC_CSC_WP_OFFSET	742
6.131.1.23	MC_CSR	742
6.131.1.24	MC_CSR_VALID	742
6.131.1.25	MC_POC	742
6.131.1.26	MC_POC_EN_BW_OFFSET	742
6.131.1.27	MC_POC_EN_BW_WIDTH	742
6.131.1.28	MC_POC_EN_MEMTYPE_OFFSET	742
6.131.1.29	MC_POC_EN_MEMTYPE_WIDTH	742
6.131.1.30	MC_POC_VALID	742
6.131.1.31	MC_TMS	742
6.131.1.32	MC_TMS_ASYNC_TRDV_OFFSET	742
6.131.1.33	MC_TMS_ASYNC_TRDV_WIDTH	742
6.131.1.34	MC_TMS_ASYNC_TRDZ_OFFSET	742
6.131.1.35	MC_TMS_ASYNC_TRDZ_WIDTH	742
6.131.1.36	MC_TMS_ASYNC_TWD_OFFSET	742
6.131.1.37	MC_TMS_ASYNC_TWD_WIDTH	742
6.131.1.38	MC_TMS_ASYNC_TWPW_OFFSET	742
6.131.1.39	MC_TMS_ASYNC_TWPW_WIDTH	742
6.131.1.40	MC_TMS_ASYNC_TWWD_OFFSET	742
6.131.1.41	MC_TMS_ASYNC_TWWD_WIDTH	742
6.131.1.42	MC_TMS_ASYNC_VALID	742
6.131.1.43	MC_TMS_SDRAM_BL_OFFSET	742
6.131.1.44	MC_TMS_SDRAM_BL_WIDTH	742
6.131.1.45	MC_TMS_SDRAM_BT_OFFSET	742
6.131.1.46	MC_TMS_SDRAM_CL_OFFSET	742
6.131.1.47	MC_TMS_SDRAM_CL_WIDTH	742
6.131.1.48	MC_TMS_SDRAM_OM_OFFSET	742
6.131.1.49	MC_TMS_SDRAM_OM_WIDTH	742

6.131.1.50	MC_TMS_SDRAM_TRCD_OFFSET	742
6.131.1.51	MC_TMS_SDRAM_TRCD_WIDTH	742
6.131.1.52	MC_TMS_SDRAM_TRFC_OFFSET	742
6.131.1.53	MC_TMS_SDRAM_TRFC_WIDTH	742
6.131.1.54	MC_TMS_SDRAM_TRP_OFFSET	742
6.131.1.55	MC_TMS_SDRAM_TRP_WIDTH	742
6.131.1.56	MC_TMS_SDRAM_TWR_OFFSET	742
6.131.1.57	MC_TMS_SDRAM_TWR_WIDTH	742
6.131.1.58	MC_TMS_SDRAM_VALID	742
6.131.1.59	MC_TMS_SDRAM_WBL_OFFSET	742
6.131.1.60	MC_TMS_SSRAM_VALID	742
6.131.1.61	MC_TMS_SYNC_TRDV_OFFSET	742
6.131.1.62	MC_TMS_SYNC_TRDV_WIDTH	742
6.131.1.63	MC_TMS_SYNC_TRDZ_OFFSET	742
6.131.1.64	MC_TMS_SYNC_TRDZ_WIDTH	742
6.131.1.65	MC_TMS_SYNC_TTO_OFFSET	742
6.131.1.66	MC_TMS_SYNC_TTO_WIDTH	742
6.131.1.67	MC_TMS_SYNC_TWR_OFFSET	742
6.131.1.68	MC_TMS_SYNC_TWR_WIDTH	742
6.131.1.69	MC_TMS_SYNC_VALID	742
6.131.1.70	MC_TMS_VALID	742
6.131.1.71	IN_CE	742
6.131.2	Function Documentation	742
6.131.2.1	mc_baseaddr	742
6.131.2.2	mc_done	742
6.131.2.3	mc_enabled	743
6.131.2.4	mc_index	743
6.131.2.5	mc_poc	743
6.131.2.6	mc_read_word	743
6.131.2.7	mc_reg_mem_area	743
6.131.2.8	mc_reset	743
6.131.2.9	mc_sec_end	744
6.131.2.10	mc_sec_start	744
6.131.2.11	lmc_status	744
6.131.2.12	mc_write_word	744
6.131.2.13	reg_mc_sec	745

6.131.2.14	set_csc_tms	745
6.131.3	Variable Documentation	745
6.131.3.1	mc_areas	745
6.131.3.2	mcs	745
6.132	peripheral/mc.h File Reference	746
6.132.1	Function Documentation	746
6.132.1.1	mc_done	746
6.132.1.2	mc_reg_mem_area	747
6.132.1.3	reg_mc_sec	747
6.133	peripheral/memory.h File Reference	748
6.133.1	Function Documentation	749
6.133.1.1	reg_memory_sec	749
6.134	peripheral/ps2kbd.c File Reference	750
6.134.1	Define Documentation	753
6.134.1.1	KBD_BAUD_RATE	753
6.134.1.2	KBD_CCMD_DKI	753
6.134.1.3	KBD_CCMD_EKI	753
6.134.1.4	KBD_CCMD_RCB	753
6.134.1.5	KBD_CCMD_ST1	753
6.134.1.6	KBD_CCMD_ST2	753
6.134.1.7	KBD_CCMD_WCB	753
6.134.1.8	KBD_CCMDBYTE_EN	753
6.134.1.9	KBD_CCMDBYTE_EN2	753
6.134.1.10	KBD_CCMDBYTE_INT	753
6.134.1.11	KBD_CCMDBYTE_INT2	753
6.134.1.12	KBD_CCMDBYTE_SYS	753
6.134.1.13	KBD_CCMDBYTE_XLAT	753
6.134.1.14	KBD_CTRL	753
6.134.1.15	KBD_DATA	753
6.134.1.16	KBD_KCMD_DK	753
6.134.1.17	KBD_KCMD_ECHO	753
6.134.1.18	KBD_KCMD_EK	753
6.134.1.19	KBD_KCMD_RST	753
6.134.1.20	KBD_KCMD_SRL	753
6.134.1.21	KBD_KRESP_ACK	753
6.134.1.22	KBD_KRESP_ECHO	753

6.134.1.23	KBD_KRESP_RSTOK	753
6.134.1.24	KBD_MAX_BUF	753
6.134.1.25	KBD_SPACE	753
6.134.1.26	KBD_STATUS_A2	753
6.134.1.27	KBD_STATUS_IBF	753
6.134.1.28	KBD_STATUS_INH	753
6.134.1.29	KBD_STATUS_MOBF	753
6.134.1.30	KBD_STATUS_OBF	753
6.134.1.31	KBD_STATUS_PERR	753
6.134.1.32	KBD_STATUS_SYS	753
6.134.1.33	KBD_STATUS_TO	753
6.134.2	Function Documentation	753
6.134.2.1	kbd_baseaddr	753
6.134.2.2	kbd_enabled	753
6.134.2.3	kbd_info	753
6.134.2.4	kbd_irq	753
6.134.2.5	kbd_job	753
6.134.2.6	kbd_put	754
6.134.2.7	kbd_read8	754
6.134.2.8	kbd_reset	754
6.134.2.9	kbd_rxfile	754
6.134.2.10	kbd_sec_end	755
6.134.2.11	kbd_sec_start	755
6.134.2.12	kbd_write8	755
6.134.2.13	reg_kbd_sec	756
6.134.2.14	scan_decode	756
6.134.3	Variable Documentation	756
6.134.3.1	code	756
6.134.3.2	scan_table	756
6.134.3.3	shift	756
6.135	peripheral/ps2kbd.h File Reference	757
6.135.1	Function Documentation	757
6.135.1.1	reg_kbd_sec	757
6.136	peripheral/vga.c File Reference	758
6.136.1	Define Documentation	760
6.136.1.1	VGA_ADDR_SPACE	760

6.136.1.2	VGA_CLUTA	760
6.136.1.3	VGA_CLUTB	760
6.136.1.4	VGA_CTRL	760
6.136.1.5	VGA_CTRL_CD	760
6.136.1.6	VGA_CTRL_PC	760
6.136.1.7	VGA_CTRL_VEN	760
6.136.1.8	VGA_HTIM	760
6.136.1.9	VGA_HVLEN	760
6.136.1.10	VGA_MASK	760
6.136.1.11	VGA_STAT	760
6.136.1.12	VGA_VBARA	760
6.136.1.13	VGA_VBARB	760
6.136.1.14	VGA_VTIM	760
6.136.2	Function Documentation	760
6.136.2.1	reg_vga_sec	760
6.136.2.2	vga_baseaddr	761
6.136.2.3	vga_dump_image	761
6.136.2.4	vga_enabled	761
6.136.2.5	vga_filename	761
6.136.2.6	vga_irq	762
6.136.2.7	vga_job	762
6.136.2.8	vga_read32	762
6.136.2.9	vga_refresh_rate	762
6.136.2.10	vga_reset	762
6.136.2.11	vga_sec_end	763
6.136.2.12	vga_sec_start	763
6.136.2.13	vga_write32	763
6.137	peripheral/vga.h File Reference	764
6.137.1	Function Documentation	764
6.137.1.1	reg_vga_sec	764
6.138	pic/pic.c File Reference	765
6.138.1	Function Documentation	766
6.138.1.1	clear_interrupt	766
6.138.1.2	pic_edge_trigger	766
6.138.1.3	pic_enabled	766
6.138.1.4	pic_ints_en	766

6.138.1.5	pic_rep_int	767
6.138.1.6	pic_reset	767
6.138.1.7	reg_pic_sec	767
6.138.1.8	report_interrupt	768
6.138.2	Variable Documentation	768
6.138.2.1	pic_state	768
6.138.2.2	pic_state_int	768
6.139	pic/pic.h File Reference	769
6.139.1	Function Documentation	769
6.139.1.1	clear_interrupt	769
6.139.1.2	pic_ints_en	769
6.139.1.3	pic_reset	769
6.139.1.4	reg_pic_sec	769
6.139.1.5	report_interrupt	770
6.140	pm/pm.c File Reference	771
6.140.1	Function Documentation	771
6.140.1.1	pm_enabled	771
6.140.1.2	pm_reset	771
6.140.1.3	reg_pm_sec	772
6.141	pm/pm.h File Reference	773
6.141.1	Function Documentation	773
6.141.1.1	pm_reset	773
6.141.1.2	reg_pm_sec	773
6.142	port/isblank.c File Reference	774
6.142.1	Function Documentation	774
6.142.1.1	isblank	774
6.143	port/port.h File Reference	775
6.143.1	Define Documentation	775
6.143.1.1	PRIx16	775
6.143.1.2	PRIx8	775
6.143.2	Function Documentation	775
6.143.2.1	isblank	775
6.143.2.2	strndup	775
6.144	port/strndup.c File Reference	776
6.144.1	Function Documentation	776
6.144.1.1	strndup	776

6.145 profiler.c File Reference	777
6.145.1 Define Documentation	778
6.145.1.1 MAX_STACK	778
6.145.2 Function Documentation	778
6.145.2.1 main_profiler	778
6.145.2.2 prof_acquire	778
6.145.2.3 prof_print	779
6.145.2.4 prof_set	779
6.145.3 Variable Documentation	779
6.145.3.1 cumulative	779
6.145.3.2 fprof	779
6.145.3.3 maxstack	779
6.145.3.4 nfunccalls	779
6.145.3.5 nstack	779
6.145.3.6 notcalls	779
6.145.3.7 prof_cycles	779
6.145.3.8 prof_func	779
6.145.3.9 prof_nfuncs	779
6.145.3.10 quiet	779
6.145.3.11 stack	780
6.146 profiler.h File Reference	781
6.146.1 Define Documentation	781
6.146.1.1 MAX_FUNCS	781
6.146.2 Function Documentation	781
6.146.2.1 main_profiler	781
6.146.2.2 prof_acquire	782
6.146.2.3 prof_set	782
6.146.3 Variable Documentation	782
6.146.3.1 prof_cycles	782
6.146.3.2 prof_func	782
6.146.3.3 prof_nfuncs	782
6.147 sim-cmd.c File Reference	783
6.147.1 Function Documentation	785
6.147.1.1 check_insn_exec	785
6.147.1.2 handle_sim_command	786
6.147.1.3 print_insn_exec	787

6.147.1.4	reenter_int	787
6.147.1.5	reg_sim_stat	788
6.147.1.6	sim_cmd_break	788
6.147.1.7	sim_cmd_breaks	788
6.147.1.8	sim_cmd_cm	788
6.147.1.9	sim_cmd_cuc	790
6.147.1.10	sim_cmd_de	791
6.147.1.11	sim_cmd_debug	791
6.147.1.12	sim_cmd_dh	791
6.147.1.13	sim_cmd_dm	792
6.147.1.14	sim_cmd_dv	792
6.147.1.15	sim_cmd_help	792
6.147.1.16	sim_cmd_hist	793
6.147.1.17	sim_cmd_info	793
6.147.1.18	sim_cmd_mprofile	793
6.147.1.19	sim_cmd_pc	794
6.147.1.20	sim_cmd_pm	794
6.147.1.21	sim_cmd_pr	794
6.147.1.22	sim_cmd_profile	794
6.147.1.23	sim_cmd_quit	795
6.147.1.24	sim_cmd_r	795
6.147.1.25	sim_cmd_reset	795
6.147.1.26	sim_cmd_run	796
6.147.1.27	sim_cmd_set	796
6.147.1.28	sim_cmd_setdbch	796
6.147.1.29	sim_cmd_stall	796
6.147.1.30	sim_cmd_stats	797
6.147.1.31	sim_cmd_trace	797
6.147.1.32	sim_cmd_unstall	797
6.147.1.33	strip_space	797
6.147.2	Variable Documentation	798
6.147.2.1	sim_commands	798
6.147.2.2	sim_stats	798
6.147.2.3	to_insn_num	798
6.148	sim-cmd.h File Reference	799
6.148.1	Function Documentation	799

6.148.1.1	handle_sim_command	799
6.148.1.2	reg_sim_stat	799
6.149	sim-config.c File Reference	800
6.149.1	Function Documentation	802
6.149.1.1	base_include	802
6.149.1.2	init_defconfig	802
6.149.1.3	parse_args	802
6.149.1.4	print_config	803
6.149.1.5	read_script_file	803
6.149.1.6	reg_config_param	805
6.149.1.7	reg_config_sec	805
6.149.1.8	reg_config_secs	805
6.149.1.9	reg_sim_sec	806
6.149.1.10	set_config	807
6.149.1.11	set_config_command	807
6.149.1.12	sim_clkcycle	807
6.149.1.13	sim_debug	807
6.149.1.14	sim_exe_log	808
6.149.1.15	sim_exe_log_end	808
6.149.1.16	sim_exe_log_fn	808
6.149.1.17	sim_exe_log_marker	808
6.149.1.18	sim_exe_log_start	808
6.149.1.19	sim_exe_log_type	808
6.149.1.20	sim_history	808
6.149.1.21	sim_mprof_fn	808
6.149.1.22	sim_mprofile	808
6.149.1.23	sim_prof_fn	808
6.149.1.24	sim_profile	808
6.149.1.25	sim_verbose	808
6.149.1.26	switch_param	808
6.149.2	Variable Documentation	808
6.149.2.1	config	808
6.149.2.2	cur_section	808
6.149.2.3	runtime	808
6.149.2.4	sections	808
6.150	sim-config.h File Reference	809

6.150.1 Define Documentation	810
6.150.1.1 CHECK_INT_TIME	810
6.150.1.2 EXE_LOG_HARDWARE	810
6.150.1.3 EXE_LOG_SIMPLE	810
6.150.1.4 EXE_LOG_SOFTWARE	810
6.150.1.5 MAX_SBUF_LEN	810
6.150.1.6 PRINTF	810
6.150.1.7 STR_SIZE	810
6.150.2 Enumeration Type Documentation	810
6.150.2.1 param_t	810
6.150.3 Function Documentation	811
6.150.3.1 init_defconfig	811
6.150.3.2 parse_args	811
6.150.3.3 print_config	811
6.150.3.4 reg_config_param	813
6.150.3.5 reg_config_sec	813
6.150.3.6 reg_config_secs	813
6.150.3.7 set_config_command	814
6.150.4 Variable Documentation	814
6.150.4.1 config	814
6.150.4.2 cur_section	814
6.150.4.3 do_stats	814
6.150.4.4 runtime	814
6.151 support/dbchs.h File Reference	815
6.152 support/debug.c File Reference	816
6.152.1 Define Documentation	817
6.152.1.1 __ORSIM_NO_DEC_DBCH	817
6.152.1.2 DECLARE_DEBUG_CHANNEL	817
6.152.1.3 DECLARE_DEBUG_CHANNEL	817
6.152.2 Function Documentation	817
6.152.2.1 debug	817
6.152.2.2 orsim_dbcl_set	817
6.152.2.3 orsim_dbcl_set_name	817
6.152.2.4 orsim_dbg_log	817
6.152.2.5 parse_dbchs	817
6.152.3 Variable Documentation	817

6.152.3.1	<code>__orsim_dbchs</code>	817
6.152.3.2	<code>debug_classes</code>	817
6.153	<code>support/debug.h</code> File Reference	818
6.153.1	Define Documentation	819
6.153.1.1	<code>__ORSIM_DBG_USE_FUNC</code>	819
6.153.1.2	<code>__ORSIM_DEBUG_LOG</code>	819
6.153.1.3	<code>__ORSIM_DPRINTF</code>	819
6.153.1.4	<code>__ORSIM_GET_DEBUGGING</code>	819
6.153.1.5	<code>__ORSIM_GET_DEBUGGING_ERR</code>	819
6.153.1.6	<code>__ORSIM_GET_DEBUGGING_FIXME</code>	819
6.153.1.7	<code>__ORSIM_GET_DEBUGGING_TRACE</code>	819
6.153.1.8	<code>__ORSIM_GET_DEBUGGING_WARN</code>	819
6.153.1.9	<code>DECLARE_DEBUG_CHANNEL</code>	819
6.153.1.10	<code>DEFAULT_DEBUG_CHANNEL</code>	819
6.153.1.11	<code>IERR</code>	820
6.153.1.12	<code>ERR_</code>	820
6.153.1.13	<code>ERR_ON</code>	820
6.153.1.14	<code>FIXME</code>	820
6.153.1.15	<code>FIXME_</code>	820
6.153.1.16	<code>FIXME_ON</code>	820
6.153.1.17	<code>TRACE</code>	820
6.153.1.18	<code>TRACE_</code>	820
6.153.1.19	<code>TRACE_ON</code>	820
6.153.1.20	<code>WARN</code>	820
6.153.1.21	<code>IWARN_</code>	820
6.153.1.22	<code>WARN_ON</code>	820
6.153.2	Enumeration Type Documentation	820
6.153.2.1	<code>__ORSIM_DEBUG_CLASS</code>	820
6.153.3	Function Documentation	820
6.153.3.1	<code>debug</code>	820
6.153.3.2	<code>orsim_dbcl_set_name</code>	821
6.153.3.3	<code>orsim_dbg_log</code>	821
6.153.3.4	<code>parse_dbchs</code>	821
6.154	<code>support/dumpverilog.c</code> File Reference	822
6.154.1	Define Documentation	823
6.154.1.1	<code>DISWIDTH</code>	823

6.154.1.2 DW	823
6.154.1.3 DWQ	823
6.154.1.4 LABELEND_CHAR	823
6.154.1.5 OR1K_MEM_VERILOG_FOOTER	823
6.154.1.6 OR1K_MEM_VERILOG_HEADER	823
6.154.2 Function Documentation	824
6.154.2.1 dumphex	824
6.154.2.2 dumpverilog	824
6.155support/dumpverilog.h File Reference	825
6.155.1 Function Documentation	826
6.155.1.1 dumphex	826
6.155.1.2 dumpverilog	826
6.156support/misc.c File Reference	827
6.156.1 Function Documentation	827
6.156.1.1 is_power2	827
6.156.1.2 log2_int	827
6.157support/misc.h File Reference	828
6.157.1 Function Documentation	828
6.157.1.1 is_power2	828
6.157.1.2 log2_int	828
6.158support/profile.c File Reference	829
6.158.1 Function Documentation	829
6.158.1.1 mprofile	829
6.159support/profile.h File Reference	830
6.159.1 Define Documentation	831
6.159.1.1 MPROF_16	831
6.159.1.2 MPROF_32	831
6.159.1.3 MPROF_8	831
6.159.1.4 MPROF_FETCH	831
6.159.1.5 MPROF_READ	831
6.159.1.6 MPROF_WRITE	831
6.159.2 Function Documentation	831
6.159.2.1 mprofile	831
6.160support/sched.c File Reference	832
6.160.1 Define Documentation	833
6.160.1.1 SCHED_HEAP_SIZE	833

6.160.1.2 SCHED_TIME_MAX	833
6.160.2 Function Documentation	833
6.160.2.1 DECLARE_DEBUG_CHANNEL	833
6.160.2.2 do_scheduler	833
6.160.2.3 sched_add	833
6.160.2.4 sched_find_remove	833
6.160.2.5 sched_guard	833
6.160.2.6 sched_init	833
6.160.2.7 sched_next_insn	833
6.160.2.8 sched_reset	833
6.160.3 Variable Documentation	833
6.160.3.1 scheduler	833
6.161 support/sched.h File Reference	834
6.161.1 Define Documentation	834
6.161.1.1 SCHED_ADD	834
6.161.1.2 SCHED_FIND_REMOVE	834
6.161.2 Function Documentation	835
6.161.2.1 do_scheduler	835
6.161.2.2 sched_add	835
6.161.2.3 sched_find_remove	835
6.161.2.4 sched_init	835
6.161.2.5 sched_next_insn	835
6.161.2.6 sched_reset	835
6.161.3 Variable Documentation	835
6.161.3.1 scheduler	835
6.162 support/simprintf.c File Reference	836
6.162.1 Define Documentation	837
6.162.1.1 FMTLEN	837
6.162.1.2 STACK_ARGS	837
6.162.2 Function Documentation	837
6.162.2.1 simgetstr	837
6.162.2.2 simprintf	837
6.162.3 Variable Documentation	837
6.162.3.1 fmtstr	837
6.163 support/simprintf.h File Reference	838
6.163.1 Function Documentation	838

6.163.1.1 simprintf	838
6.164tick/tick.c File Reference	839
6.164.1 Function Documentation	840
6.164.1.1 sched_timer_job	840
6.164.1.2 spr_read_ttcrr	840
6.164.1.3 spr_write_ttcrr	840
6.164.1.4 spr_write_ttmr	841
6.164.1.5 tick_one_shot	841
6.164.1.6 tick_raise_except	841
6.164.1.7 tick_reset	841
6.164.1.8 tick_restart	842
6.164.2 Variable Documentation	842
6.164.2.1 cycles_start	842
6.164.2.2 tick_count	842
6.165tick/tick.h File Reference	843
6.165.1 Function Documentation	843
6.165.1.1 spr_read_ttcrr	843
6.165.1.2 spr_write_ttcrr	843
6.165.1.3 spr_write_ttmr	844
6.165.1.4 tick_reset	844
6.166toplevel-mprofile.c File Reference	845
6.166.1 Function Documentation	847
6.166.1.1 main	847
6.167toplevel-profile.c File Reference	848
6.167.1 Function Documentation	850
6.167.1.1 main	850
6.168toplevel-support.c File Reference	851
6.168.1 Function Documentation	852
6.168.1.1 check_int	852
6.168.1.2 ctrl_c	853
6.168.1.3 reg_sim_reset	854
6.168.1.4 sim_done	854
6.168.1.5 sim_init	854
6.168.1.6 sim_reset	855
6.168.2 Variable Documentation	856
6.168.2.1 sim_reset_hooks	856

6.169	toplevel-support.h File Reference	857
6.169.1	Function Documentation	857
6.169.1.1	check_int	857
6.169.1.2	ctrl_c	858
6.169.1.3	reg_sim_reset	858
6.169.1.4	sim_done	859
6.169.1.5	sim_init	859
6.169.1.6	sim_reset	860
6.170	toplevel.c File Reference	862
6.170.1	Function Documentation	862
6.170.1.1	main	862
6.171	vapi/vapi.c File Reference	865
6.171.1	Function Documentation	866
6.171.1.1	add_handler	866
6.171.1.2	find_handler	866
6.171.1.3	get_server_socket	867
6.171.1.4	handler_fits_id	867
6.171.1.5	read_packet	867
6.171.1.6	rebuild_fds	867
6.171.1.7	reg_vapi_sec	867
6.171.1.8	server_request	867
6.171.1.9	vapi_check	868
6.171.1.10	vapi_done	868
6.171.1.11	vapi_enabled	868
6.171.1.12	vapi_hide_device_id	868
6.171.1.13	vapi_init	868
6.171.1.14	vapi_install_handler	868
6.171.1.15	vapi_install_multi_handler	869
6.171.1.16	vapi_log_enabled	869
6.171.1.17	vapi_log_fn	869
6.171.1.18	vapi_num_unconnected	869
6.171.1.19	vapi_read_stream	869
6.171.1.20	vapi_request	869
6.171.1.21	vapi_send	870
6.171.1.22	vapi_server_port	870
6.171.1.23	vapi_write_log_file	870

6.171.1.24	vapi_write_stream	870
6.171.1.25	write_packet	870
6.171.2	Variable Documentation	870
6.171.2.1	fds	870
6.171.2.2	nfds	870
6.171.2.3	nhandlers	870
6.171.2.4	server_fd	870
6.171.2.5	serverIP	870
6.171.2.6	tcp_level	870
6.172	vapi/vapi.h File Reference	871
6.172.1	Define Documentation	871
6.172.1.1	VAPI_MAX_DEVID	871
6.172.2	Enumeration Type Documentation	871
6.172.2.1	VAPI_COMMAND	871
6.172.3	Function Documentation	872
6.172.3.1	reg_vapi_sec	872
6.172.3.2	vapi_check	872
6.172.3.3	vapi_done	872
6.172.3.4	vapi_init	873
6.172.3.5	vapi_install_handler	873
6.172.3.6	vapi_install_multi_handler	873
6.172.3.7	vapi_num_unconnected	873
6.172.3.8	vapi_send	873
6.172.3.9	vapi_write_log_file	874

Chapter 1

Or1ksim: the OpenRISC 1000 Architectural Simulator

1.1 About

This is the entire code base for Or1ksim. It is described in more detail in its user guide, which can be found in the `doc` directory.

Or1ksim uses the ArgTable 2 library to parse its arguments. This documentation excludes that code base.

1.2 Installation

See the `INSTALL` file in the main directory for information on building and installing these programs

1.3 Documentation

The user guide can be found as a texinfo file in the `doc` subdirectory.

This document may be converted to HTML, PDF, info files or PostScript by the commands "make html", "make pdf", "make info" and "make ps" respectively.

1.4 Copying

This file is part of OpenRISC Architectural Simulator.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Chapter 2

Todo List

Global [rsp_insert_matchpoint](#) This doesn't work with icache/immu yet

Global [rsp_remove_matchpoint](#) This doesn't work with icache/immu yet

Global [rsp_write_all_regs](#) There is no error checking at present. Non-hex chars will generate a warning message, but there is no other check that the right amount of data is present. The result is always "OK".

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

<code>_csm_list</code>	13
<code>_cuc_func</code>	15
<code>_dep_list_t</code>	17
<code>archf</code>	18
<code>ata_device</code>	19
<code>ata_devices</code>	24
<code>ata_host</code>	25
<code>bff</code>	28
<code>BMP_HEADER</code>	29
<code>bpb_entry</code>	30
<code>bpbstat</code>	31
<code>branchstat</code>	32
<code>breakpoint_entry</code>	33
<code>btic_entry</code>	34
<code>bticstat</code>	35
<code>cachestats_entry</code>	36
<code>channel</code>	37
<code>channel_factory</code>	38
<code>channel_ops</code>	39
<code>COFF_AOUTHDR</code>	40
<code>COFF_auxent</code>	41
<code>COFF_filehdr</code>	45
<code>COFF_lineno</code>	46
<code>COFF_reloc</code>	47
<code>COFF_scnhdr</code>	48
<code>COFF_slib</code>	49
<code>COFF_syment</code>	50
<code>config</code>	51
<code>config::pic</code>	57
<code>config_param</code>	58
<code>config_section</code>	59
<code>cpu_state</code>	60
<code>cuc_bb</code>	62

cuc_conv	64
cuc_insn	65
cuc_known_insn	66
cuc_shared_item	67
cuc_timing_table	68
cuc_timings	69
dc_set	70
dev_16450	71
dev_generic	76
dev_memarea	78
dma_channel	80
dma_controller	83
dmmu	85
dmmustats_entry	87
dstats_entry	88
dyn_page	89
dynamic	90
elf32_hdr	91
elf32_note	93
elf32_phdr	94
elf32_rel	95
elf32_rela	96
elf32_shdr	97
elf32_sym	98
Elf64_Dyn	99
elf64_hdr	100
elf64_note	102
elf64_phdr	103
elf64_rel	104
elf64_rela	105
elf64_shdr	106
elf64_sym	107
elf_obj	108
eth_device	110
ether_addr	115
ether_header	116
fb_state	117
fd_channel	119
file_channel	120
fstats_entry	121
func_struct	122
gpio_device	123
hist_exec	125
ic	126
immu	128
immustats_entry	130
INFOHEADER	131
iqueue_entry	132
jtr_chain_message	133
jtr_chain_response	134
jtr_failure_response	135
jtr_read_block_message	136
jtr_read_block_response	137
jtr_read_message	138

jtr_read_response	139
jtr_write_block_message	140
jtr_write_block_response	141
jtr_write_message	142
jtr_write_response	143
kbd_state	144
label_entry	145
mc	146
mc_area	148
mem_config	149
mem_ops	151
memory_hash	153
mp_entry	154
mprofentry_struct	155
mstats_entry	156
op_queue	157
param_val	159
raw_stats	160
reloc	161
rsp_buf	162
runtime	163
sched_entry	167
scheduler_struct	168
sim_command	169
sim_reset_hook	170
sim_stat	171
spr_bit_def	172
spr_def	173
sstats_entry	174
stack_struct	175
tcp_channel	176
tty_channel	177
vapi_handler	178
vga_state	179
xterm_channel	181

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

cpu-config.c	206
cpu-config.h	210
libtoplevel.c	563
mainpage	569
mprofiler.c	587
mprofiler.h	591
or1ksim.h	592
profiler.c	777
profiler.h	781
sim-cmd.c	783
sim-cmd.h	799
sim-config.c	800
sim-config.h	809
toplevel-mprofile.c	845
toplevel-profile.c	848
toplevel-support.c	851
toplevel-support.h	857
toplevel.c	862
bpb/branch-predict.c	183
bpb/branch-predict.h	188
cache/dcache-model.c	190
cache/dcache-model.h	195
cache/icache-model.c	198
cache/icache-model.h	203
cpu/common/abstract.c	211
cpu/common/abstract.h	229
cpu/common/coff.h	244
cpu/common/elf.h	251
cpu/common/execute.h	261
cpu/common/labels.c	268
cpu/common/labels.h	270
cpu/common/parse.c	272
cpu/common/parse.h	278

cpu/common/stats.c	280
cpu/common/stats.h	284
cpu/common/trace.c	286
cpu/common/trace.h	288
cpu/or1k/arch.h	290
cpu/or1k/except.c	292
cpu/or1k/except.h	294
cpu/or1k/spr-defs.h	298
cpu/or1k/spr-dump.c	308
cpu/or1k/spr-dump.h	320
cpu/or1k/sprs.c	321
cpu/or1k/sprs.h	323
cpu/or32/common-i386.h	325
cpu/or32/def-op-t.h	326
cpu/or32/dyn-rec.c	328
cpu/or32/dyn-rec.h	352
cpu/or32/dyn32-defs.h	357
cpu/or32/dyngen-elf.c	358
cpu/or32/dyngen-i386.c	360
cpu/or32/dyngen.c	362
cpu/or32/dyngen.h	365
cpu/or32/execute.c	366
cpu/or32/generate.c	377
cpu/or32/i386-regs.h	380
cpu/or32/insnset.c	381
cpu/or32/op-1t-op.h	393
cpu/or32/op-1t.h	394
cpu/or32/op-2t-op.h	395
cpu/or32/op-2t.h	396
cpu/or32/op-3t-op.h	397
cpu/or32/op-3t.h	398
cpu/or32/op-arith-op.h	399
cpu/or32/op-comp-op.h	400
cpu/or32/op-extend-op.h	401
cpu/or32/op-ff1-op.h	402
cpu/or32/op-i386.h	403
cpu/or32/op-lwhb-op.h	404
cpu/or32/op-mac-op.h	405
cpu/or32/op-mftspr-op.h	406
cpu/or32/op-support.c	407
cpu/or32/op-support.h	409
cpu/or32/op-swhb-op.h	411
cpu/or32/op-t-reg-mov-op.h	412
cpu/or32/op.c	416
cpu/or32/or32.c	430
cpu/or32/rec-i386.h	437
cpu/or32/sched-i386.h	438
cpu/or32/simpl32-defs.h	439
cuc/adv.c	440
cuc/bb.c	442
cuc/cuc.c	448
cuc/cuc.h	464
cuc/insn.c	482
cuc/insn.h	488

cuc/load.c	494
cuc/memory.c	499
cuc/timings.c	507
cuc/verilog.c	510
cuc/verilog.h	513
debug/debug-unit.c	514
debug/debug-unit.h	524
debug/gdb.h	529
debug/gdbcomm.c	531
debug/gdbcomm.h	535
debug/rsp-server.c	537
debug/rsp-server.h	560
mmu/dmmu.c	570
mmu/dmmu.h	576
mmu/immu.c	579
mmu/immu.h	584
peripheral/16450.c	595
peripheral/16450.h	617
peripheral/atacmd.h	621
peripheral/atadevice-cmdi.c	625
peripheral/atadevice-cmdi.h	630
peripheral/atadevice.c	635
peripheral/atadevice.h	639
peripheral/atahost.c	644
peripheral/atahost.h	654
peripheral/crc32.c	678
peripheral/crc32.h	679
peripheral/dma-defs.h	680
peripheral/dma.c	687
peripheral/dma.h	694
peripheral/eth.c	696
peripheral/eth.h	711
peripheral/fb.c	713
peripheral/fb.h	719
peripheral/fields.h	720
peripheral/generic.c	722
peripheral/generic.h	728
peripheral/gpio.c	730
peripheral/gpio.h	737
peripheral/mc.c	738
peripheral/mc.h	746
peripheral/memory.c	502
peripheral/memory.h	748
peripheral/ps2kbd.c	750
peripheral/ps2kbd.h	757
peripheral/vga.c	758
peripheral/vga.h	764
peripheral/channels/channel.c	660
peripheral/channels/channel.h	662
peripheral/channels/fd.c	663
peripheral/channels/fd.h	665
peripheral/channels/file.c	666
peripheral/channels/file.h	668
peripheral/channels/generic.c	727

peripheral/channels/ generic.h	729
peripheral/channels/ tcp.c	669
peripheral/channels/ tcp.h	671
peripheral/channels/ tty.c	672
peripheral/channels/ tty.h	674
peripheral/channels/ xterm.c	675
peripheral/channels/ xterm.h	677
pic/ pic.c	765
pic/ pic.h	769
pm/ pm.c	771
pm/ pm.h	773
port/ isblank.c	774
port/ port.h	775
port/ strndup.c	776
support/ dbchs.h	815
support/ debug.c	816
support/ debug.h	818
support/ dumpverilog.c	822
support/ dumpverilog.h	825
support/ misc.c	827
support/ misc.h	828
support/ profile.c	829
support/ profile.h	830
support/ sched.c	832
support/ sched.h	834
support/ simprintf.c	836
support/ simprintf.h	838
tick/ tick.c	839
tick/ tick.h	843
vapi/ vapi.c	865
vapi/ vapi.h	871

Chapter 5

Data Structure Documentation

5.1 `_csm_list` Struct Reference

```
#include <cuc.h>
```

Collaboration diagram for `_csm_list`:



Data Fields

- `int ref`
- `int cnt`
- `int cmovs`
- `double size`
- `double osize`
- `int cmatch`
- `int dead`
- `int ninsn`
- `struct _csm_list * from`
- `struct _csm_list * next`

5.1.1 Field Documentation

5.1.1.1 `int _csm_list::ref`

5.1.1.2 `int _csm_list::cnt`

5.1.1.3 `int _csm_list::cmovs`

5.1.1.4 `double _csm_list::size`

5.1.1.5 `double _csm_list::osize`

5.1.1.6 `int _csm_list::cmatch`

5.1.1.7 `int _csm_list::dead`

5.1.1.8 `int _csm_list::ninsn`

5.1.1.9 `struct _csm_list* _csm_list::from` [read]

5.1.1.10 `struct _csm_list* _csm_list::next` [read]

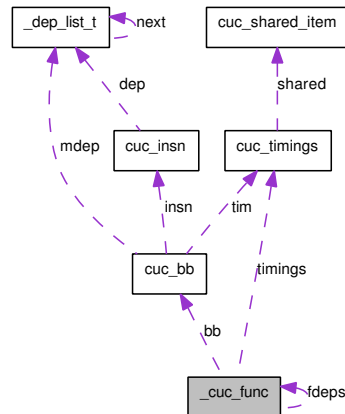
The documentation for this struct was generated from the following file:

- [cuc/cuc.h](#)

5.2 `_cuc_func` Struct Reference

```
#include <cuc.h>
```

Collaboration diagram for `_cuc_func`:



Data Fields

- int `num_bb`
- `cuc_bb` `bb` [MAX_BB]
- int `saved_regs` [MAX_REGS]
- int `lur` [MAX_REGS]
- int `used_regs` [MAX_REGS]
- int `nmsched`
- int `msched` [MAX_INSNS]
- int `mtype` [MAX_INSNS]
- int `num_init_bb`
- int * `init_bb_reloc`
- int `orig_time`
- int `num_runs`
- `cuc_timings` `timings`
- unsigned long `start_addr`
- unsigned long `end_addr`
- int `memory_order`
- int `nfdeps`
- struct `_cuc_func` ** `fdeps`
- int `tmp`

5.2.1 Field Documentation

- 5.2.1.1 `int _cuc_func::num_bb`
- 5.2.1.2 `cuc_bb _cuc_func::bb[MAX_BB]`
- 5.2.1.3 `int _cuc_func::saved_regs[MAX_REGS]`
- 5.2.1.4 `int _cuc_func::lur[MAX_REGS]`
- 5.2.1.5 `int _cuc_func::used_regs[MAX_REGS]`
- 5.2.1.6 `int _cuc_func::nmsched`
- 5.2.1.7 `int _cuc_func::msched[MAX_INSNS]`
- 5.2.1.8 `int _cuc_func::mtype[MAX_INSNS]`
- 5.2.1.9 `int _cuc_func::num_init_bb`
- 5.2.1.10 `int* _cuc_func::init_bb_reloc`
- 5.2.1.11 `int _cuc_func::orig_time`
- 5.2.1.12 `int _cuc_func::num_runs`
- 5.2.1.13 `cuc_timings _cuc_func::timings`
- 5.2.1.14 `unsigned long _cuc_func::start_addr`
- 5.2.1.15 `unsigned long _cuc_func::end_addr`
- 5.2.1.16 `int _cuc_func::memory_order`
- 5.2.1.17 `int _cuc_func::nfdeps`
- 5.2.1.18 `struct _cuc_func** _cuc_func::fdeps` [read]
- 5.2.1.19 `int _cuc_func::tmp`

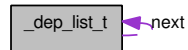
The documentation for this struct was generated from the following file:

- [cuc/cuc.h](#)

5.3 `_dep_list_t` Struct Reference

```
#include <cuc.h>
```

Collaboration diagram for `_dep_list_t`:



Data Fields

- unsigned long `ref`
- struct `_dep_list_t` * `next`

5.3.1 Field Documentation

5.3.1.1 unsigned long `_dep_list_t::ref`

5.3.1.2 struct `_dep_list_t`* `_dep_list_t::next` [read]

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

- [cuc/cuc.h](#)

5.4 archf Struct Reference

```
#include <dyngen.h>
```

Data Fields

- unsigned int(* [get_real_func_len](#))(void *func, unsigned int len, char *name)
- void(* [gen_reloc](#))(FILE *f, struct [reloc](#) *reloc, unsigned int param)
- void(* [gen_func_reloc](#))(FILE *f, struct [reloc](#) *reloc)

5.4.1 Field Documentation

5.4.1.1 unsigned int(* archf::get_real_func_len)(void *func, unsigned int len, char *name)

5.4.1.2 void(* archf::gen_reloc)(FILE *f, struct [reloc](#) *reloc, unsigned int param)

5.4.1.3 void(* archf::gen_func_reloc)(FILE *f, struct [reloc](#) *reloc)

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

- [cpu/or32/dyngen.h](#)

5.5 ata_device Struct Reference

```
#include <atadevice.h>
```

Data Fields

- struct {
 - void * [host](#)
 - int [dev](#)
 - int [pio_mode](#)
 - int [dma_mode](#)
 - uint16_t [dbuf](#) [4096]
 - uint16_t * [dbuf_ptr](#)
 - uint16_t [dbuf_cnt](#)
 - int [state](#)
 - unsigned int [heads_per_cylinder](#)
 - unsigned int [sectors_per_track](#)
 - uint32_t [lba](#)
 - int [nr_sect](#)
 - void(* [end_t_func](#))(struct [ata_device](#) *)
- [internals](#)
- struct {
 - uint8_t [command](#)
 - uint8_t [cylinder_low](#)
 - uint8_t [cylinder_high](#)
 - uint8_t [device_control](#)
 - uint8_t [device_head](#)
 - uint8_t [error](#)
 - uint8_t [features](#)
 - uint8_t [sector_count](#)
 - uint8_t [sector_number](#)
 - uint8_t [status](#)
 - uint16_t [dataport_i](#)
- [regs](#)
- struct {
 - int [iordy](#)
 - int [intrq](#)
 - int [dmarq](#)
 - int [pdiagi](#)
 - int [pdiago](#)
 - int [daspi](#)
 - int [daspo](#)
- [sigs](#)
- struct {
 - char * [file](#)
 - FILE * [stream](#)
 - int [type](#)
 - uint32_t [size](#)
 - uint32_t [size_sect](#)

```
int packet
unsigned int heads
unsigned int sectors
char * firmware
unsigned int mwdma
unsigned int pio
} conf
```


5.5.1 Field Documentation

5.5.1.1 void* ata_device::host

5.5.1.2 int ata_device::dev

5.5.1.3 int ata_device::pio_mode

5.5.1.4 int ata_device::dma_mode

5.5.1.5 uint16_t ata_device::dbuf[4096]

5.5.1.6 uint16_t* ata_device::dbuf_ptr

5.5.1.7 uint16_t ata_device::dbuf_cnt

5.5.1.8 int ata_device::state

5.5.1.9 unsigned int ata_device::heads_per_cylinder

5.5.1.10 unsigned int ata_device::sectors_per_track

5.5.1.11 uint32_t ata_device::lba

5.5.1.12 int ata_device::nr_sect

5.5.1.13 void(* ata_device::end_t_func)(struct ata_device *)

5.5.1.14 struct { ... } ata_device::internals

5.5.1.15 uint8_t ata_device::command

5.5.1.16 uint8_t ata_device::cylinder_low

5.5.1.17 uint8_t ata_device::cylinder_high

5.5.1.18 uint8_t ata_device::device_control

5.5.1.19 uint8_t ata_device::device_head

5.5.1.20 uint8_t ata_device::error

5.5.1.21 uint8_t ata_device::features

5.5.1.22 uint8_t ata_device::sector_count

5.5.1.23 uint8_t ata_device::sector_number

5.5.1.24 uint8_t ata_device::status

5.5.1.25 uint16_t ata_device::dataport_i

5.5.1.26 struct { ... } ata_device::regs

5.5.1.27 int ata_device::iordy

5.5.1.28 int ata_device::intrq

5.5.1.29 int ata_device::dmarq

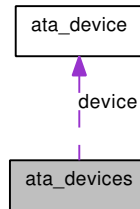
5.5.1.30 int ata_device::ndiagi

- [peripheral/atadevice.h](#)

5.6 ata_devices Struct Reference

```
#include <atadevice.h>
```

Collaboration diagram for ata_devices:



Data Fields

- struct [ata_device](#) `device` [2]

5.6.1 Field Documentation

5.6.1.1 struct `ata_device` `ata_devices::device[2]` [read]

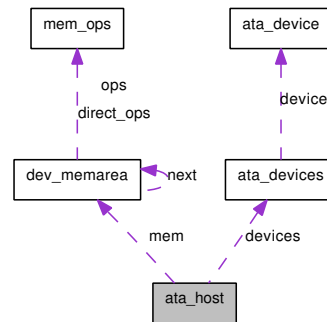
The documentation for this struct was generated from the following file:

- [peripheral/atadevice.h](#)

5.7 ata_host Struct Reference

```
#include <atahost.h>
```

Collaboration diagram for ata_host:



Data Fields

- int [enabled](#)
- [oraddr_t](#) baseaddr
- struct [dev_memarea](#) * mem
- int irq
- int dev_id
- int rev
- int dev_sel
- [uint8_t](#) pio_mode0_t1
- [uint8_t](#) pio_mode0_t2
- [uint8_t](#) pio_mode0_t4
- [uint8_t](#) pio_mode0_teoc
- [uint8_t](#) dma_mode0_tm
- [uint8_t](#) dma_mode0_td
- [uint8_t](#) dma_mode0_teoc
- struct {
 - [uint32_t](#) ctrl
 - [uint32_t](#) stat
 - [uint32_t](#) pctr
 - [uint32_t](#) pftr0
 - [uint32_t](#) pftr1
 - [uint32_t](#) dtr0
 - [uint32_t](#) dtr1
 - [uint32_t](#) txb
 - [uint32_t](#) rxb
 } [regs](#)
- struct [ata_devices](#) devices

5.7.1 Field Documentation

- 5.7.1.1 `int ata_host::enabled`
- 5.7.1.2 `oraddr_t ata_host::baseaddr`
- 5.7.1.3 `struct dev_memarea* ata_host::mem` [read]
- 5.7.1.4 `int ata_host::irq`
- 5.7.1.5 `int ata_host::dev_id`
- 5.7.1.6 `int ata_host::rev`
- 5.7.1.7 `int ata_host::dev_sel`
- 5.7.1.8 `uint8_t ata_host::pio_mode0_t1`
- 5.7.1.9 `uint8_t ata_host::pio_mode0_t2`
- 5.7.1.10 `uint8_t ata_host::pio_mode0_t4`
- 5.7.1.11 `uint8_t ata_host::pio_mode0_tec`
- 5.7.1.12 `uint8_t ata_host::dma_mode0_tm`
- 5.7.1.13 `uint8_t ata_host::dma_mode0_td`
- 5.7.1.14 `uint8_t ata_host::dma_mode0_tec`
- 5.7.1.15 `uint32_t ata_host::ctrl`
- 5.7.1.16 `uint32_t ata_host::stat`
- 5.7.1.17 `uint32_t ata_host::pctr`
- 5.7.1.18 `uint32_t ata_host::pftr0`
- 5.7.1.19 `uint32_t ata_host::pftr1`
- 5.7.1.20 `uint32_t ata_host::dtr0`
- 5.7.1.21 `uint32_t ata_host::dtr1`
- 5.7.1.22 `uint32_t ata_host::txb`
- 5.7.1.23 `uint32_t ata_host::rxb`
- 5.7.1.24 `struct { ... } ata_host::regs`
- 5.7.1.25 `struct ata_devices ata_host::devices` [read]

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

- [peripheral/atahost.h](#)

5.8 bff Struct Reference

```
#include <dyngen.h>
```

Data Fields

- void *****(***** [open_obj](#))(const char *****object)
- void(***** [close_obj](#))(void *****)
- char *****(***** [get_func_name](#))(void *****, unsigned int [func](#))
- void *****(***** [get_func_start](#))(void *****, unsigned int [func](#))
- unsigned int(***** [get_func_len](#))(void *****, unsigned int [func](#))
- int(***** [get_func_reloc](#))(void *****, unsigned int [func](#), unsigned int [relocn](#), struct [reloc](#) *****[reloc](#))

5.8.1 Field Documentation

5.8.1.1 void *****(***** [bff::open_obj](#))(const char *****object)

5.8.1.2 void(***** [bff::close_obj](#))(void *****)

5.8.1.3 char *****(***** [bff::get_func_name](#))(void *****, unsigned int [func](#))

5.8.1.4 void *****(***** [bff::get_func_start](#))(void *****, unsigned int [func](#))

5.8.1.5 unsigned int(***** [bff::get_func_len](#))(void *****, unsigned int [func](#))

5.8.1.6 int(***** [bff::get_func_reloc](#))(void *****, unsigned int [func](#), unsigned int [relocn](#), struct [reloc](#) *****[reloc](#))

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

- [cpu/or32/dyngen.h](#)

5.9 BMP_HEADER Struct Reference

Data Fields

- unsigned short int [type](#)
- unsigned int [size](#)
- unsigned short int [reserved1](#)
- unsigned short int [reserved2](#)
- unsigned int [offset](#)

5.9.1 Field Documentation

5.9.1.1 unsigned short int BMP_HEADER::type

5.9.1.2 unsigned int BMP_HEADER::size

5.9.1.3 unsigned short int BMP_HEADER::reserved1

5.9.1.4 unsigned short int BMP_HEADER::reserved2

5.9.1.5 unsigned int BMP_HEADER::offset

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

- [peripheral/vga.c](#)

5.10 bpb_entry Struct Reference

Data Fields

- struct {
 - [oraddr_t](#) addr
 - int taken
 - int lru
- [way](#) [BPB_WAYS]

5.10.1 Field Documentation

5.10.1.1 [oraddr_t](#) bpb_entry::addr

5.10.1.2 int bpb_entry::taken

5.10.1.3 int bpb_entry::lru

5.10.1.4 struct { ... } bpb_entry::way[BPB_WAYS]

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

- [bpb/branch-predict.c](#)

5.11 bpbstat Struct Reference

```
#include <stats.h>
```

Data Fields

- int [hit](#)
- int [miss](#)
- int [correct](#)
- int [incorrect](#)

5.11.1 Field Documentation

5.11.1.1 int `bpbstat::hit`

5.11.1.2 int `bpbstat::miss`

5.11.1.3 int `bpbstat::correct`

5.11.1.4 int `bpbstat::incorrect`

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

- `cpu/common/stats.h`

5.12 branchstat Struct Reference

Data Fields

- int [taken](#)
- int [nottaken](#)
- int [forward](#)
- int [backward](#)

5.12.1 Field Documentation

5.12.1.1 int branchstat::taken

5.12.1.2 int branchstat::nottaken

5.12.1.3 int branchstat::forward

5.12.1.4 int branchstat::backward

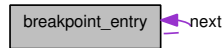
The documentation for this struct was generated from the following file:

- [cpu/common/stats.c](#)

5.13 breakpoint_entry Struct Reference

```
#include <labels.h>
```

Collaboration diagram for breakpoint_entry:



Data Fields

- [oraddr_t addr](#)
- struct [breakpoint_entry](#) * [next](#)

5.13.1 Detailed Description

Structure representing a breakpoint

5.13.2 Field Documentation

5.13.2.1 oraddr_t breakpoint_entry::addr

5.13.2.2 struct breakpoint_entry* breakpoint_entry::next [read]

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

- [cpu/common/labels.h](#)

5.14 btic_entry Struct Reference

Data Fields

- struct {
 - [oraddr_t](#) `addr`
 - int `lru`
 - char * `insn`
 - } `way` [`BTIC_WAYS`]

5.14.1 Field Documentation

5.14.1.1 `oraddr_t btic_entry::addr`

5.14.1.2 `int btic_entry::lru`

5.14.1.3 `char* btic_entry::insn`

5.14.1.4 `struct { ... } btic_entry::way[BTIC_WAYS]`

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

- [bpb/branch-predict.c](#)

5.15 bticstat Struct Reference

```
#include <stats.h>
```

Data Fields

- int [hit](#)
- int [miss](#)

5.15.1 Field Documentation

5.15.1.1 int bticstat::hit

5.15.1.2 int bticstat::miss

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

- [cpu/common/stats.h](#)

5.16 cachestats_entry Struct Reference

```
#include <stats.h>
```

Data Fields

- int [readhit](#)
- int [readmiss](#)
- int [writehit](#)
- int [writemiss](#)

5.16.1 Field Documentation

5.16.1.1 int cachestats_entry::readhit

5.16.1.2 int cachestats_entry::readmiss

5.16.1.3 int cachestats_entry::writehit

5.16.1.4 int cachestats_entry::writemiss

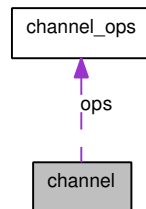
The documentation for this struct was generated from the following file:

- [cpu/common/stats.h](#)

5.17 channel Struct Reference

```
#include <channel.h>
```

Collaboration diagram for channel:



Data Fields

- struct [channel_ops](#) * ops
- void * data

5.17.1 Detailed Description

A data structure representing a [channel](#). Its operations and data

5.17.2 Field Documentation

5.17.2.1 struct [channel_ops](#)* [channel::ops](#) [read]

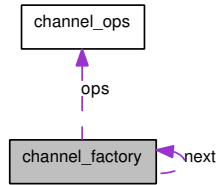
5.17.2.2 void* [channel::data](#)

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

- peripheral/channels/[channel.h](#)

5.18 channel_factory Struct Reference

Collaboration diagram for channel_factory:



Data Fields

- const char * [name](#)
- struct [channel_ops](#) * [ops](#)
- struct [channel_factory](#) * [next](#)

5.18.1 Field Documentation

5.18.1.1 const char* `channel_factory::name`

5.18.1.2 struct [channel_ops](#)* `channel_factory::ops` [read]

5.18.1.3 struct [channel_factory](#)* `channel_factory::next` [read]

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

- [peripheral/channels/channel.c](#)

5.19 channel_ops Struct Reference

```
#include <channel.h>
```

Data Fields

- void *****(*** init**)(const char *)
- int(*** open**)(void *)
- void(*** close**)(void *)
- int(*** read**)(void *, char *, int)
- int(*** write**)(void *, const char *, int)
- void(*** free**)(void *)
- int(*** isok**)(void *)
- char *****(*** status**)(void *)

5.19.1 Detailed Description

A data structure representing all the functions required on a [channel](#)

5.19.2 Field Documentation

5.19.2.1 void(*** channel_ops::init**)(const char *)

5.19.2.2 int(*** channel_ops::open**)(void *)

5.19.2.3 void(*** channel_ops::close**)(void *)

5.19.2.4 int(*** channel_ops::read**)(void *, char *, int)

5.19.2.5 int(*** channel_ops::write**)(void *, const char *, int)

5.19.2.6 void(*** channel_ops::free**)(void *)

5.19.2.7 int(*** channel_ops::isok**)(void *)

5.19.2.8 char(*** channel_ops::status**)(void *)

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

- peripheral/channels/[channel.h](#)

5.20 COFF_AOUTHDR Struct Reference

```
#include <coff.h>
```

Data Fields

- char [magic](#) [2]
- char [vstamp](#) [2]
- char [tsize](#) [4]
- char [dsize](#) [4]
- char [bsize](#) [4]
- char [entry](#) [4]
- char [text_start](#) [4]
- char [data_start](#) [4]

5.20.1 Field Documentation

5.20.1.1 char COFF_AOUTHDR::magic[2]

5.20.1.2 char COFF_AOUTHDR::vstamp[2]

5.20.1.3 char COFF_AOUTHDR::tsize[4]

5.20.1.4 char COFF_AOUTHDR::dsize[4]

5.20.1.5 char COFF_AOUTHDR::bsize[4]

5.20.1.6 char COFF_AOUTHDR::entry[4]

5.20.1.7 char COFF_AOUTHDR::text_start[4]

5.20.1.8 char COFF_AOUTHDR::data_start[4]

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

- [cpu/common/coff.h](#)

5.21 COFF_auxent Union Reference

```
#include <coff.h>
```

Data Fields

- struct {
 - char [x_tagndx](#) [4]
 - union {
 - struct {
 - char [x_inno](#) [2]
 - char [x_size](#) [2]
 - } [x_insz](#)
 - char [x_fsize](#) [4]
 - } [x_misc](#)
 - union {
 - struct {
 - char [x_innoptr](#) [4]
 - char [x_endndx](#) [4]
 - } [x_fcn](#)
 - struct {
 - char [x_dimen](#) [E_DIMNUM][2]
 - } [x_ary](#)
 - } [x_fcary](#)
 - char [x_tvndx](#) [2]
 - } [x_sym](#)
- union {
 - char [x_fname](#) [E_FILNMLEN]
 - struct {
 - char [x_zeroes](#) [4]
 - char [x_offset](#) [4]
 - } [x_n](#)
- } [x_file](#)
- struct {
 - char [x_scnlen](#) [4]
 - char [x_nreloc](#) [2]
 - char [x_nlinno](#) [2]
- } [x_scn](#)
- struct {
 - char [x_tvfill](#) [4]
 - char [x_tvlen](#) [2]
 - char [x_tvran](#) [2][2]
- } [x_tv](#)

5.21.1 Field Documentation

- 5.21.1.1 char COFF_auxent::x_tagndx[4]
- 5.21.1.2 char COFF_auxent::x_inno[2]
- 5.21.1.3 char COFF_auxent::x_size[2]
- 5.21.1.4 struct { ... } ::@25 COFF_auxent::x_insz
- 5.21.1.5 char COFF_auxent::x_fsize[4]
- 5.21.1.6 union { ... } COFF_auxent::x_misc
- 5.21.1.7 char COFF_auxent::x_innoptr[4]
- 5.21.1.8 char COFF_auxent::x_endndx[4]
- 5.21.1.9 struct { ... } ::@26 COFF_auxent::x_fcn
- 5.21.1.10 char COFF_auxent::x_dimen[E_DIMNUM][2]
- 5.21.1.11 struct { ... } ::@27 COFF_auxent::x_ary
- 5.21.1.12 union { ... } COFF_auxent::x_fcary
- 5.21.1.13 char COFF_auxent::x_tvndx[2]
- 5.21.1.14 struct { ... } COFF_auxent::x_sym
- 5.21.1.15 char COFF_auxent::x_fname[E_FILNMLEN]
- 5.21.1.16 char COFF_auxent::x_zeroes[4]
- 5.21.1.17 char COFF_auxent::x_offset[4]
- 5.21.1.18 struct { ... } COFF_auxent::x_n
- 5.21.1.19 union { ... } COFF_auxent::x_file
- 5.21.1.20 char COFF_auxent::x_scnlen[4]
- 5.21.1.21 char COFF_auxent::x_nreloc[2]
- 5.21.1.22 char COFF_auxent::x_nlinno[2]
- 5.21.1.23 struct { ... } COFF_auxent::x_scn
- 5.21.1.24 char COFF_auxent::x_tvfill[4]
- 5.21.1.25 char COFF_auxent::x_tvlen[2]
- 5.21.1.26 char COFF_auxent::x_tvran[2][2]

5.21.1.27 struct { ... } COFF_auxent::x_tv

Generated on Tue Nov 11 11:50:04 2008 for COFF.auxent by Doxygen

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

- [cpu/common/coff.h](#)

5.22 COFF_filehdr Struct Reference

```
#include <coff.h>
```

Data Fields

- char [f_magic](#) [2]
- char [f_nscns](#) [2]
- char [f_timdat](#) [4]
- char [f_symptr](#) [4]
- char [f_nsyms](#) [4]
- char [f_opthdr](#) [2]
- char [f_flags](#) [2]

5.22.1 Field Documentation

5.22.1.1 char COFF_filehdr::f_magic[2]

5.22.1.2 char COFF_filehdr::f_nscns[2]

5.22.1.3 char COFF_filehdr::f_timdat[4]

5.22.1.4 char COFF_filehdr::f_symptr[4]

5.22.1.5 char COFF_filehdr::f_nsyms[4]

5.22.1.6 char COFF_filehdr::f_opthdr[2]

5.22.1.7 char COFF_filehdr::f_flags[2]

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

- [cpu/common/coff.h](#)

5.23 COFF_lineno Struct Reference

```
#include <coff.h>
```

Data Fields

- union {
 - char [l_symndx](#) [4]
 - char [l_paddr](#) [4]
- } [l_addr](#)
- char [l_inno](#) [2]

5.23.1 Field Documentation

5.23.1.1 char COFF_lineno::l_symndx[4]

5.23.1.2 char COFF_lineno::l_paddr[4]

5.23.1.3 union { ... } COFF_lineno::l_addr

5.23.1.4 char COFF_lineno::l_inno[2]

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

- [cpu/common/coff.h](#)

5.24 COFF_reloc Struct Reference

```
#include <coff.h>
```

Data Fields

- char [r_vaddr](#) [4]
- char [r_symndx](#) [4]
- char [r_type](#) [2]

5.24.1 Field Documentation

5.24.1.1 char COFF_reloc::r_vaddr[4]

5.24.1.2 char COFF_reloc::r_symndx[4]

5.24.1.3 char COFF_reloc::r_type[2]

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

- [cpu/common/coff.h](#)

5.25 COFF_scnhdr Struct Reference

```
#include <coff.h>
```

Data Fields

- char [s_name](#) [8]
- char [s_paddr](#) [4]
- char [s_vaddr](#) [4]
- char [s_size](#) [4]
- char [s_scnptr](#) [4]
- char [s_relptr](#) [4]
- char [s_innoptr](#) [4]
- char [s_nreloc](#) [2]
- char [s_nlno](#) [2]
- char [s_flags](#) [4]

5.25.1 Field Documentation

5.25.1.1 char COFF_scnhdr::s_name[8]

5.25.1.2 char COFF_scnhdr::s_paddr[4]

5.25.1.3 char COFF_scnhdr::s_vaddr[4]

5.25.1.4 char COFF_scnhdr::s_size[4]

5.25.1.5 char COFF_scnhdr::s_scnptr[4]

5.25.1.6 char COFF_scnhdr::s_relptr[4]

5.25.1.7 char COFF_scnhdr::s_innoptr[4]

5.25.1.8 char COFF_scnhdr::s_nreloc[2]

5.25.1.9 char COFF_scnhdr::s_nlno[2]

5.25.1.10 char COFF_scnhdr::s_flags[4]

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

- [cpu/common/coff.h](#)

5.26 COFF_slib Struct Reference

```
#include <coff.h>
```

Data Fields

- char [sl_entz](#) [4]
- char [sl_pathndx](#) [4]

5.26.1 Field Documentation

5.26.1.1 char COFF_slib::sl_entz[4]

5.26.1.2 char COFF_slib::sl_pathndx[4]

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

- [cpu/common/coff.h](#)

5.27 COFF_syment Struct Reference

```
#include <coff.h>
```

Data Fields

- union {
 - char [e_name](#) [E_SYMNMLEN]
 - struct {
 - char [e_zeroes](#) [4]
 - char [e_offset](#) [4]
 - } [e](#)
- char [e_value](#) [4]
- char [e_snum](#) [2]
- char [e_type](#) [2]
- char [e_sclass](#) [1]
- char [e_numaux](#) [1]

5.27.1 Field Documentation

5.27.1.1 char COFF_syment::e_name[E_SYMNMLEN]

5.27.1.2 char COFF_syment::e_zeroes[4]

5.27.1.3 char COFF_syment::e_offset[4]

5.27.1.4 struct { ... } COFF_syment::e

5.27.1.5 union { ... } COFF_syment::e

5.27.1.6 char COFF_syment::e_value[4]

5.27.1.7 char COFF_syment::e_snum[2]

5.27.1.8 char COFF_syment::e_type[2]

5.27.1.9 char COFF_syment::e_sclass[1]

5.27.1.10 char COFF_syment::e_numaux[1]

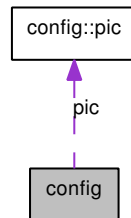
The documentation for this struct was generated from the following file:

- [cpu/common/coff.h](#)

5.28 config Struct Reference

```
#include <sim-config.h>
```

Collaboration diagram for config:



Data Fields

- struct {
 - void * [class_ptr](#)
 - unsigned long int(* [read_up](#))(void *[class_ptr](#), unsigned long int addr, unsigned long int mask)
 - void(* [write_up](#))(void *[class_ptr](#), unsigned long int addr, unsigned long int mask, unsigned long int wdata)
 } [ext](#)
- struct {
 - int [debug](#)
 - int [verbose](#)
 - int [profile](#)
 - char * [prof_fn](#)
 - int [mprofile](#)
 - char * [mprof_fn](#)
 - int [history](#)
 - int [exe_log](#)
 - int [exe_log_type](#)
 - long long int [exe_log_start](#)
 - long long int [exe_log_end](#)
 - int [exe_log_marker](#)
 - char * [exe_log_fn](#)
 - long [clkcycle_ps](#)
 } [sim](#)
- struct {
 - int [enabled](#)
 - int [server_port](#)
 - int [log_enabled](#)
 - int [hide_device_id](#)
 - char * [vapi_fn](#)
 } [vapi](#)
- struct {
 - char * [timings_fn](#)
 - int [memory_order](#)
 - int [calling_convention](#)
 - int [enable_bursts](#)

- ```
 int no_multicycle
} cuc
```
- struct {  
 int superscalar  
 int hazards  
 int dependstats  
 int sbuf\_len  
} cpu
  - struct {  
 int enabled  
 int nways  
 int nsets  
 int blocksize  
 int ustates  
 int store\_missdelay  
 int store\_hitdelay  
 int load\_missdelay  
 int load\_hitdelay  
} dc
  - struct config::pic pic
  - struct {  
 int enabled  
} pm
  - struct {  
 int enabled  
 int sbp\_bnf\_fwd  
 int sbp\_bf\_fwd  
 int btic  
 int missdelay  
 int hitdelay  
} bpb
  - struct {  
 int enabled  
 int gdb\_enabled  
 int rsp\_enabled  
 int server\_port  
 int rsp\_port  
 unsigned long vapi\_id  
} debug

## Data Structures

- struct pic

### 5.28.1 Detailed Description

Data structure for configuration data



## 5.28.2 Field Documentation

5.28.2.1 void\* config::class\_ptr

5.28.2.2 unsigned long int(\* config::read\_up)(void \*class\_ptr, unsigned long int addr, unsigned long int mask)

5.28.2.3 void(\* config::write\_up)(void \*class\_ptr, unsigned long int addr, unsigned long int mask, unsigned long int wdata)

5.28.2.4 struct { ... } config::ext

5.28.2.5 int config::debug

5.28.2.6 int config::verbose

5.28.2.7 int config::profile

5.28.2.8 char\* config::prof\_fn

5.28.2.9 int config::mprofile

5.28.2.10 char\* config::mprof\_fn

5.28.2.11 int config::history

5.28.2.12 int config::exe\_log

5.28.2.13 int config::exe\_log\_type

5.28.2.14 long long int config::exe\_log\_start

5.28.2.15 long long int config::exe\_log\_end

5.28.2.16 int config::exe\_log\_marker

5.28.2.17 char\* config::exe\_log\_fn

5.28.2.18 long config::clkcycle\_ps

5.28.2.19 struct { ... } config::sim

5.28.2.20 int config::enabled

5.28.2.21 int config::server\_port

5.28.2.22 int config::log\_enabled

5.28.2.23 int config::hide\_device\_id

5.28.2.24 char\* config::vapi\_fn

5.28.2.25 struct { ... } config::vapi

5.28.2.26 char\* config::timings\_fn

---

Generated on Tue Nov 11 11:50:34 2008 for OpenRISC: The OpenRISC 1000 Architectural Simulator by Doxygen

5.28.2.27 int config::memory\_order

5.28.2.28 int config::calling\_convention

5.28.2.29 int config::enable\_bursts

- [sim-config.h](#)



## 5.29 config::pic Struct Reference

```
#include <sim-config.h>
```

### Data Fields

- int [enabled](#)
- int [edge\\_trigger](#)

### 5.29.1 Field Documentation

#### 5.29.1.1 int config::pic::enabled

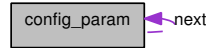
#### 5.29.1.2 int config::pic::edge\_trigger

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

- [sim-config.h](#)

## 5.30 config\_param Struct Reference

Collaboration diagram for config\_param:



### Data Fields

- char \* [name](#)
- enum [param\\_t](#) [type](#)
- void(\* [func](#) )(union [param\\_val](#), void \*dat)
- struct [config\\_param](#) \* [next](#)

### 5.30.1 Field Documentation

**5.30.1.1** char\* [config\\_param::name](#)

**5.30.1.2** enum [param\\_t](#) [config\\_param::type](#)

**5.30.1.3** void(\* [config\\_param::func](#))(union [param\\_val](#), void \*dat)

**5.30.1.4** struct [config\\_param](#)\* [config\\_param::next](#) [[read](#)]

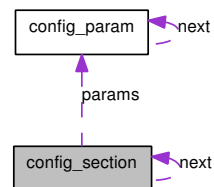
The documentation for this struct was generated from the following file:

- [sim-config.c](#)

## 5.31 config\_section Struct Reference

```
#include <sim-config.h>
```

Collaboration diagram for config\_section:



### Data Fields

- char \* [name](#)
- void \*(\* [sec\\_start](#) )(void)
- void(\* [sec\\_end](#) )(void \*)
- void \* [dat](#)
- struct [config\\_param](#) \* [params](#)
- struct [config\\_section](#) \* [next](#)

### 5.31.1 Field Documentation

**5.31.1.1** char\* [config\\_section::name](#)

**5.31.1.2** void>(\* [config\\_section::sec\\_start](#))(void)

**5.31.1.3** void(\* [config\\_section::sec\\_end](#))(void \*)

**5.31.1.4** void\* [config\\_section::dat](#)

**5.31.1.5** struct [config\\_param](#)\* [config\\_section::params](#) [read]

**5.31.1.6** struct [config\\_section](#)\* [config\\_section::next](#) [read]

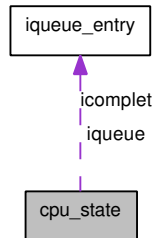
The documentation for this struct was generated from the following file:

- [sim-config.h](#)

## 5.32 `cpu_state` Struct Reference

```
#include <execute.h>
```

Collaboration diagram for `cpu_state`:



### Data Fields

- `uorreg_t reg` [MAX\_GPRS]
- `uorreg_t sprs` [MAX\_SPRS]
- `oraddr_t insn_ea`
- `int delay_insn`
- `oraddr_t pc`
- `oraddr_t pc_delay`
- `uint32_t pic_lines`
- `struct iqueue_entry iqueue`
- `struct iqueue_entry icomplet`

### 5.32.1 Detailed Description

The main structure holding the current execution state of the CPU

Not to be confused with `runtime`, which holds the state of the simulation.

`insn_ea` field is only used to get `dump_exe_log()` correct.

`iqueue` and `icomplet` fields are only used in `analysis()`.

The micro-operation queue, `opqs`, is only used to speed up `recompile_page()`.

### 5.32.2 Field Documentation

#### 5.32.2.1 `uorreg_t cpu_state::reg`[MAX\_GPRS]

General purpose registers

#### 5.32.2.2 `uorreg_t cpu_state::sprs`[MAX\_SPRS]

Special purpose registers

**5.32.2.3** `oraddr_t cpu_state::insn_ea`

EA of instrs that have an EA

**5.32.2.4** `int cpu_state::delay_insn`

Is current instr in delay slot

**5.32.2.5** `oraddr_t cpu_state::pc`

PC (and translated PC)

**5.32.2.6** `oraddr_t cpu_state::pc_delay`

Delay instr EA register

**5.32.2.7** `uint32_t cpu_state::pic_lines`

State of PIC lines

**5.32.2.8** `struct iqueue_entry cpu_state::iqueue` [read]

Decode of just executed instr

**5.32.2.9** `struct iqueue_entry cpu_state::icomplet` [read]

Decode of instr before this

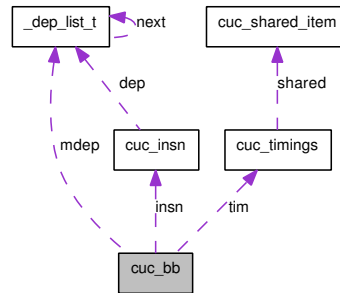
The documentation for this struct was generated from the following file:

- [cpu/common/execute.h](#)

### 5.33 cuc\_bb Struct Reference

```
#include <cuc.h>
```

Collaboration diagram for cuc\_bb:



#### Data Fields

- unsigned long `type`
- int `first`
- int `last`
- int `prev` [2]
- int `next` [2]
- int `tmp`
- `cuc_insn` \* `insn`
- int `ninsn`
- int `last_used_reg` [MAX\_REGS]
- `dep_list` \* `mdep`
- int `nmemory`
- int `cnt`
- int `unrolled`
- int `ntim`
- `cuc_timings` \* `tim`
- int `selected_tim`

### 5.33.1 Field Documentation

- 5.33.1.1 unsigned long cuc\_bb::type
- 5.33.1.2 int cuc\_bb::first
- 5.33.1.3 int cuc\_bb::last
- 5.33.1.4 int cuc\_bb::prev[2]
- 5.33.1.5 int cuc\_bb::next[2]
- 5.33.1.6 int cuc\_bb::tmp
- 5.33.1.7 cuc\_insn\* cuc\_bb::insn
- 5.33.1.8 int cuc\_bb::ninsn
- 5.33.1.9 int cuc\_bb::last\_used\_reg[MAX\_REGS]
- 5.33.1.10 dep\_list\* cuc\_bb::mdep
- 5.33.1.11 int cuc\_bb::nmemory
- 5.33.1.12 int cuc\_bb::cnt
- 5.33.1.13 int cuc\_bb::unrolled
- 5.33.1.14 int cuc\_bb::ntim
- 5.33.1.15 cuc\_timings\* cuc\_bb::tim
- 5.33.1.16 int cuc\_bb::selected\_tim

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

- [cuc/cuc.h](#)

## 5.34 `cuc_conv` Struct Reference

```
#include <insn.h>
```

### Data Fields

- const char \* [from](#)
- const int [to](#)

### 5.34.1 Field Documentation

#### 5.34.1.1 `const char* cuc_conv::from`

#### 5.34.1.2 `const int cuc_conv::to`

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

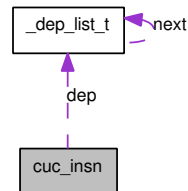
- [cuc/insn.h](#)



## 5.35 cuc\_insn Struct Reference

```
#include <cuc.h>
```

Collaboration diagram for cuc\_insn:



### Data Fields

- int `type`
- int `index`
- int `opt` [MAX\_OPERANDS]
- unsigned long `op` [MAX\_OPERANDS]
- `dep_list` \* `dep`
- unsigned long `insn`
- char `disasm` [40]
- unsigned long `max`
- int `tmp`

### 5.35.1 Field Documentation

5.35.1.1 int `cuc_insn::type`

5.35.1.2 int `cuc_insn::index`

5.35.1.3 int `cuc_insn::opt`[MAX\_OPERANDS]

5.35.1.4 unsigned long `cuc_insn::op`[MAX\_OPERANDS]

5.35.1.5 `dep_list`\* `cuc_insn::dep`

5.35.1.6 unsigned long `cuc_insn::insn`

5.35.1.7 char `cuc_insn::disasm`[40]

5.35.1.8 unsigned long `cuc_insn::max`

5.35.1.9 int `cuc_insn::tmp`

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

- [cuc/cuc.h](#)

## 5.36 `cuc_known_insn` Struct Reference

```
#include <insn.h>
```

### Data Fields

- char \* [name](#)
- int [comutative](#)
- char \* [rtl](#)

### 5.36.1 Field Documentation

**5.36.1.1** char\* `cuc_known_insn::name`

**5.36.1.2** int `cuc_known_insn::comutative`

**5.36.1.3** char\* `cuc_known_insn::rtl`

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

- [cuc/insn.h](#)

## 5.37 `cuc_shared_item` Struct Reference

```
#include <cuc.h>
```

### Data Fields

- `int ref`
- `int cmatch`

### 5.37.1 Field Documentation

#### 5.37.1.1 `int cuc_shared_item::ref`

#### 5.37.1.2 `int cuc_shared_item::cmatch`

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

- `cuc/cuc.h`

## 5.38 `cuc_timing_table` Struct Reference

```
#include <insn.h>
```

### Data Fields

- double [delay](#)
- double [size](#)
- double [delayi](#)
- double [sizei](#)

### 5.38.1 Field Documentation

**5.38.1.1** double `cuc_timing_table::delay`

**5.38.1.2** double `cuc_timing_table::size`

**5.38.1.3** double `cuc_timing_table::delayi`

**5.38.1.4** double `cuc_timing_table::sizei`

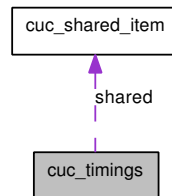
The documentation for this struct was generated from the following file:

- [cuc/insn.h](#)

## 5.39 cuc\_timings Struct Reference

```
#include <cuc.h>
```

Collaboration diagram for cuc\_timings:



### Data Fields

- int `b`
- int `preroll`
- int `unroll`
- int `nshared`
- `cuc_shared_item * shared`
- int `new_time`
- double `size`

### 5.39.1 Field Documentation

**5.39.1.1** int `cuc_timings::b`

**5.39.1.2** int `cuc_timings::preroll`

**5.39.1.3** int `cuc_timings::unroll`

**5.39.1.4** int `cuc_timings::nshared`

**5.39.1.5** `cuc_shared_item*` `cuc_timings::shared`

**5.39.1.6** int `cuc_timings::new_time`

**5.39.1.7** double `cuc_timings::size`

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

- [cuc/cuc.h](#)

## 5.40 dc\_set Struct Reference

### Data Fields

- struct {
  - uint32\_t [line](#) [MAX\_DC\_BLOCK\_SIZE/4]
  - [oraddr\\_t tagaddr](#)
  - int [lru](#)
- } [way](#) [MAX\_DC\_WAYS]

### 5.40.1 Field Documentation

5.40.1.1 [uint32\\_t dc\\_set::line](#)[MAX\_DC\_BLOCK\_SIZE/4]

5.40.1.2 [oraddr\\_t dc\\_set::tagaddr](#)

5.40.1.3 [int dc\\_set::lru](#)

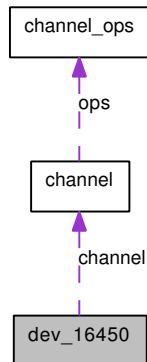
5.40.1.4 [struct { ... } dc\\_set::way](#)[MAX\_DC\_WAYS]

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

- [cache/dcache-model.c](#)

## 5.41 dev\_16450 Struct Reference

Collaboration diagram for dev\_16450:



### Data Fields

- struct {
  - uint8\_t `txbuf` [UART\_MAX\_FIFO\_LEN]
  - uint16\_t `rxbuf` [UART\_MAX\_FIFO\_LEN]
  - uint8\_t `dll`
  - uint8\_t `dlh`
  - uint8\_t `ier`
  - uint8\_t `iir`
  - uint8\_t `fcr`
  - uint8\_t `lcr`
  - uint8\_t `mcr`
  - uint8\_t `lsr`
  - uint8\_t `msr`
  - uint8\_t `scr`
 } `regs`
- struct {
  - uint8\_t `txser`
  - uint16\_t `rxser`
  - uint8\_t `loopback`
 } `iregs`
- struct {
  - int `txbuf_head`
  - int `txbuf_tail`
  - int `rxbuf_head`
  - int `rxbuf_tail`
  - unsigned int `txbuf_full`
  - unsigned int `rxbuf_full`
  - int `receiveing`
  - int `recv_break`
  - int `ints`
 } `istat`

- unsigned long `char_clks`
- struct {
  - unsigned long `char_clks`
  - uint8\_t `dll`
  - uint8\_t `dlh`
  - uint8\_t `lcr`
  - int `skew` } `vapi`
- unsigned long `vapi_buf` [UART\_VAPI\_BUF\_LEN]
- int `vapi_buf_head_ptr`
- int `vapi_buf_tail_ptr`
- int `fifo_len`
- struct `channel` \* `channel`
- int `enabled`
- int `jitter`
- `oraddr_t` `baseaddr`
- int `irq`
- unsigned long `vapi_id`
- int `uart16550`
- char \* `channel_str`





### 5.41.1 Field Documentation

5.41.1.1 `uint8_t dev_16450::txbuf[UART_MAX_FIFO_LEN]`

5.41.1.2 `uint16_t dev_16450::rxbuf[UART_MAX_FIFO_LEN]`

5.41.1.3 `uint8_t dev_16450::dll`

5.41.1.4 `uint8_t dev_16450::dlh`

5.41.1.5 `uint8_t dev_16450::ier`

5.41.1.6 `uint8_t dev_16450::iir`

5.41.1.7 `uint8_t dev_16450::fcr`

5.41.1.8 `uint8_t dev_16450::lcr`

5.41.1.9 `uint8_t dev_16450::mcr`

5.41.1.10 `uint8_t dev_16450::lsr`

5.41.1.11 `uint8_t dev_16450::msr`

5.41.1.12 `uint8_t dev_16450::scr`

5.41.1.13 `struct { ... } dev_16450::regs`

5.41.1.14 `uint8_t dev_16450::txser`

5.41.1.15 `uint16_t dev_16450::rxser`

5.41.1.16 `uint8_t dev_16450::loopback`

5.41.1.17 `struct { ... } dev_16450::iregs`

5.41.1.18 `int dev_16450::txbuf_head`

5.41.1.19 `int dev_16450::txbuf_tail`

5.41.1.20 `int dev_16450::rxbuf_head`

5.41.1.21 `int dev_16450::rxbuf_tail`

5.41.1.22 `unsigned int dev_16450::txbuf_full`

5.41.1.23 `unsigned int dev_16450::rxbuf_full`

5.41.1.24 `int dev_16450::receiveing`

5.41.1.25 `int dev_16450::recv_break`

5.41.1.26 `int dev_16450::ints`

5.41.1.27 `struct { ... } dev_16450::istat`

5.41.1.28 `unsigned long dev_16450::char_clks`

5.41.1.29 `int dev_16450::skew`

5.41.1.30 `struct { ... } dev_16450::vapi`

- [peripheral/16450.c](#)

## 5.42 dev\_generic Struct Reference

### Public Types

- enum { [GENERIC\\_READ](#), [GENERIC\\_WRITE](#) }
- enum { [GENERIC\\_BYTE](#), [GENERIC\\_HW](#), [GENERIC\\_WORD](#) }

### Data Fields

- enum dev\_generic:: { ... } [trans\\_direction](#)
- enum dev\_generic:: { ... } [trans\\_size](#)
- uint32\_t [value](#)
- int [enabled](#)
- int [byte\\_enabled](#)
- int [hw\\_enabled](#)
- int [word\\_enabled](#)
- char \* [name](#)
- [oraddr\\_t](#) [baseaddr](#)
- uint32\_t [size](#)

### 5.42.1 Detailed Description

State associated with the generic device.

### 5.42.2 Member Enumeration Documentation

#### 5.42.2.1 anonymous enum

Enumerator:

*GENERIC\_READ*

*GENERIC\_WRITE*

#### 5.42.2.2 anonymous enum

Enumerator:

*GENERIC\_BYTE*

*GENERIC\_HW*

*GENERIC\_WORD*

### 5.42.3 Field Documentation

5.42.3.1 `enum { ... } dev_generic::trans_direction`

5.42.3.2 `enum { ... } dev_generic::trans_size`

5.42.3.3 `uint32_t dev_generic::value`

5.42.3.4 `int dev_generic::enabled`

5.42.3.5 `int dev_generic::byte_enabled`

5.42.3.6 `int dev_generic::hw_enabled`

5.42.3.7 `int dev_generic::word_enabled`

5.42.3.8 `char* dev_generic::name`

5.42.3.9 `oraddr_t dev_generic::baseaddr`

5.42.3.10 `uint32_t dev_generic::size`

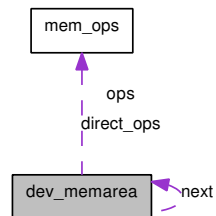
The documentation for this struct was generated from the following file:

- [peripheral/generic.c](#)

## 5.43 dev\_memarea Struct Reference

```
#include <abstract.h>
```

Collaboration diagram for dev\_memarea:



### Data Fields

- struct `dev_memarea` \* `next`
- `oraddr_t` `addr_mask`
- `oraddr_t` `addr_compare`
- `uint32_t` `size`
- `oraddr_t` `size_mask`
- int `valid`
- FILE \* `log`
- struct `mem_ops` `ops`
- struct `mem_ops` `direct_ops`

### 5.43.1 Detailed Description

Memory regions assigned to devices

## 5.43.2 Field Documentation

5.43.2.1 `struct dev_memarea* dev_memarea::next` [read]

5.43.2.2 `oraddr_t dev_memarea::addr_mask`

5.43.2.3 `oraddr_t dev_memarea::addr_compare`

5.43.2.4 `uint32_t dev_memarea::size`

5.43.2.5 `oraddr_t dev_memarea::size_mask`

5.43.2.6 `int dev_memarea::valid`

5.43.2.7 `FILE* dev_memarea::log`

5.43.2.8 `struct mem_ops dev_memarea::ops` [read]

5.43.2.9 `struct mem_ops dev_memarea::direct_ops` [read]

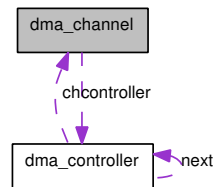
The documentation for this struct was generated from the following file:

- [cpu/common/abstract.h](#)

## 5.44 dma\_channel Struct Reference

```
#include <dma.h>
```

Collaboration diagram for dma\_channel:



### Data Fields

- struct [dma\\_controller](#) \* [controller](#)
- unsigned [channel\\_number](#)
- unsigned long [channel\\_mask](#)
- unsigned [referenced](#)
- unsigned [load\\_next\\_descriptor\\_when\\_done](#)
- unsigned long [current\\_descriptor](#)
- [oraddr\\_t](#) [source](#)
- [oraddr\\_t](#) [destination](#)
- [oraddr\\_t](#) [source\\_mask](#)
- [oraddr\\_t](#) [destination\\_mask](#)
- unsigned long [chunk\\_size](#)
- unsigned long [total\\_size](#)
- unsigned long [words\\_transferred](#)
- struct {
  - unsigned long [csr](#)
  - unsigned long [sz](#)
  - unsigned long [a0](#)
  - unsigned long [am0](#)
  - unsigned long [a1](#)
  - unsigned long [am1](#)
  - unsigned long [desc](#)
  - unsigned long [swptr](#)
 } [regs](#)
- unsigned [dma\\_req\\_i](#)
- unsigned [dma\\_ack\\_o](#)
- unsigned [dma\\_nd\\_i](#)



### 5.44.1 Field Documentation

- 5.44.1.1 `struct dma_controller* dma_channel::controller` [read]
- 5.44.1.2 `unsigned dma_channel::channel_number`
- 5.44.1.3 `unsigned long dma_channel::channel_mask`
- 5.44.1.4 `unsigned dma_channel::referenced`
- 5.44.1.5 `unsigned dma_channel::load_next_descriptor_when_done`
- 5.44.1.6 `unsigned long dma_channel::current_descriptor`
- 5.44.1.7 `oraddr_t dma_channel::source`
- 5.44.1.8 `oraddr_t dma_channel::destination`
- 5.44.1.9 `oraddr_t dma_channel::source_mask`
- 5.44.1.10 `oraddr_t dma_channel::destination_mask`
- 5.44.1.11 `unsigned long dma_channel::chunk_size`
- 5.44.1.12 `unsigned long dma_channel::total_size`
- 5.44.1.13 `unsigned long dma_channel::words_transferred`
- 5.44.1.14 `unsigned long dma_channel::csr`
- 5.44.1.15 `unsigned long dma_channel::sz`
- 5.44.1.16 `unsigned long dma_channel::a0`
- 5.44.1.17 `unsigned long dma_channel::am0`
- 5.44.1.18 `unsigned long dma_channel::a1`
- 5.44.1.19 `unsigned long dma_channel::am1`
- 5.44.1.20 `unsigned long dma_channel::desc`
- 5.44.1.21 `unsigned long dma_channel::swptr`
- 5.44.1.22 `struct { ... } dma_channel::regs`
- 5.44.1.23 `unsigned dma_channel::dma_req_i`
- 5.44.1.24 `unsigned dma_channel::dma_ack_o`
- 5.44.1.25 `unsigned dma_channel::dma_nd_i`

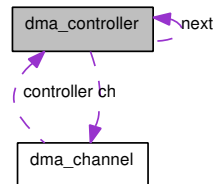
The documentation for this struct was generated from the following file:

- [peripheral/dma.h](#)

## 5.45 dma\_controller Struct Reference

```
#include <dma.h>
```

Collaboration diagram for dma\_controller:



### Data Fields

- int `enabled`
- `oraddr_t` `baseaddr`
- unsigned `irq`
- int `vapi_id`
- struct {
  - unsigned long `csr`
  - unsigned long `int_msk_a`
  - unsigned long `int_msk_b`
  - unsigned long `int_src_a`
  - unsigned long `int_src_b`} `regs`
- struct `dma_channel` `ch` [DMA\_NUM\_CHANNELS]
- struct `dma_controller` \* `next`

### 5.45.1 Field Documentation

5.45.1.1 `int dma_controller::enabled`

5.45.1.2 `oraddr_t dma_controller::baseaddr`

5.45.1.3 `unsigned dma_controller::irq`

5.45.1.4 `int dma_controller::vapi_id`

5.45.1.5 `unsigned long dma_controller::csr`

5.45.1.6 `unsigned long dma_controller::int_msk_a`

5.45.1.7 `unsigned long dma_controller::int_msk_b`

5.45.1.8 `unsigned long dma_controller::int_src_a`

5.45.1.9 `unsigned long dma_controller::int_src_b`

5.45.1.10 `struct { ... } dma_controller::regs`

5.45.1.11 `struct dma_channel dma_controller::ch[DMA_NUM_CHANNELS]` [read]

5.45.1.12 `struct dma_controller* dma_controller::next` [read]

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

- [peripheral/dma.h](#)

## 5.46 dmmu Struct Reference

```
#include <dmmu.h>
```

### Data Fields

- int `enabled`
- int `nways`
- int `nsets`
- int `pagesize`
- int `pagesize_log2`
- `oraddr_t` `page_offset_mask`
- `oraddr_t` `page_mask`
- `oraddr_t` `vpn_mask`
- int `lru_reload`
- `oraddr_t` `set_mask`
- int `entrysize`
- int `ustates`
- int `missdelay`
- int `hitdelay`

## 5.46.1 Field Documentation

- 5.46.1.1 `int dmmu::enabled`
- 5.46.1.2 `int dmmu::nways`
- 5.46.1.3 `int dmmu::nsets`
- 5.46.1.4 `int dmmu::pagesize`
- 5.46.1.5 `int dmmu::pagesize_log2`
- 5.46.1.6 `oraddr_t dmmu::page_offset_mask`
- 5.46.1.7 `oraddr_t dmmu::page_mask`
- 5.46.1.8 `oraddr_t dmmu::vpn_mask`
- 5.46.1.9 `int dmmu::lru_reload`
- 5.46.1.10 `oraddr_t dmmu::set_mask`
- 5.46.1.11 `int dmmu::entrysize`
- 5.46.1.12 `int dmmu::ustates`
- 5.46.1.13 `int dmmu::missdelay`
- 5.46.1.14 `int dmmu::hitdelay`

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

- [mmu/dmmu.h](#)

## 5.47 dmmustats\_entry Struct Reference

```
#include <stats.h>
```

### Data Fields

- int [loads\\_tlbhit](#)
- int [loads\\_tlbmiss](#)
- int [loads\\_pagefaults](#)
- int [stores\\_tlbhit](#)
- int [stores\\_tlbmiss](#)
- int [stores\\_pagefaults](#)

### 5.47.1 Field Documentation

**5.47.1.1** int dmmustats\_entry::loads\_tlbhit

**5.47.1.2** int dmmustats\_entry::loads\_tlbmiss

**5.47.1.3** int dmmustats\_entry::loads\_pagefaults

**5.47.1.4** int dmmustats\_entry::stores\_tlbhit

**5.47.1.5** int dmmustats\_entry::stores\_tlbmiss

**5.47.1.6** int dmmustats\_entry::stores\_pagefaults

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

- [cpu/common/stats.h](#)

## 5.48 dstats\_entry Struct Reference

### Data Fields

- int [insn1](#)
- int [insn2](#)
- int [cnt\\_dynamic](#)
- int [depend](#)

### 5.48.1 Field Documentation

5.48.1.1 int `dstats_entry::insn1`

5.48.1.2 int `dstats_entry::insn2`

5.48.1.3 int `dstats_entry::cnt_dynamic`

5.48.1.4 int `dstats_entry::depend`

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

- `cpu/common/stats.c`



## 5.49 dyn\_page Struct Reference

```
#include <dyn-rec.h>
```

### Data Fields

- [oraddr\\_t or\\_page](#)
- void \* [host\\_page](#)
- unsigned int [host\\_len](#)
- int [dirty](#)
- int [delayr](#)
- uint16\_t [ts\\_bound](#) [2049]
- void \*\* [locs](#)
- uint32\_t \* [insns](#)
- unsigned int \* [insn\\_indexes](#)

### 5.49.1 Field Documentation

**5.49.1.1** [oraddr\\_t dyn\\_page::or\\_page](#)

**5.49.1.2** [void\\* dyn\\_page::host\\_page](#)

**5.49.1.3** [unsigned int dyn\\_page::host\\_len](#)

**5.49.1.4** [int dyn\\_page::dirty](#)

**5.49.1.5** [int dyn\\_page::delayr](#)

**5.49.1.6** [uint16\\_t dyn\\_page::ts\\_bound\[2049\]](#)

**5.49.1.7** [void\\*\\* dyn\\_page::locs](#)

**5.49.1.8** [uint32\\_t\\* dyn\\_page::insns](#)

**5.49.1.9** [unsigned int\\* dyn\\_page::insn\\_indexes](#)

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

- [cpu/or32/dyn-rec.h](#)

## 5.50 dynamic Struct Reference

```
#include <elf.h>
```

### Data Fields

- [Elf32\\_Sword d\\_tag](#)
- union {
  - [Elf32\\_Sword d\\_val](#)
  - [Elf32\\_Addr d\\_ptr](#)
- [d\\_un](#)

### 5.50.1 Field Documentation

**5.50.1.1** [Elf32\\_Sword dynamic::d\\_tag](#)

**5.50.1.2** [Elf32\\_Sword dynamic::d\\_val](#)

**5.50.1.3** [Elf32\\_Addr dynamic::d\\_ptr](#)

**5.50.1.4** [union { ... } dynamic::d\\_un](#)

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

- [cpu/common/elf.h](#)

## 5.51 elf32\_hdr Struct Reference

```
#include <elf.h>
```

### Data Fields

- unsigned char [e\\_ident](#) [EI\_NIDENT]
- [Elf32\\_Half e\\_type](#)
- [Elf32\\_Half e\\_machine](#)
- [Elf32\\_Word e\\_version](#)
- [Elf32\\_Addr e\\_entry](#)
- [Elf32\\_Off e\\_phoff](#)
- [Elf32\\_Off e\\_shoff](#)
- [Elf32\\_Word e\\_flags](#)
- [Elf32\\_Half e\\_ehsize](#)
- [Elf32\\_Half e\\_phentsize](#)
- [Elf32\\_Half e\\_phnum](#)
- [Elf32\\_Half e\\_shentsize](#)
- [Elf32\\_Half e\\_shnum](#)
- [Elf32\\_Half e\\_shstrndx](#)

### 5.51.1 Field Documentation

5.51.1.1 unsigned char elf32\_hdr::e\_ident[EI\_NIDENT]

5.51.1.2 Elf32\_Half elf32\_hdr::e\_type

5.51.1.3 Elf32\_Half elf32\_hdr::e\_machine

5.51.1.4 Elf32\_Word elf32\_hdr::e\_version

5.51.1.5 Elf32\_Addr elf32\_hdr::e\_entry

5.51.1.6 Elf32\_Off elf32\_hdr::e\_phoff

5.51.1.7 Elf32\_Off elf32\_hdr::e\_shoff

5.51.1.8 Elf32\_Word elf32\_hdr::e\_flags

5.51.1.9 Elf32\_Half elf32\_hdr::e\_ehsize

5.51.1.10 Elf32\_Half elf32\_hdr::e\_phentsize

5.51.1.11 Elf32\_Half elf32\_hdr::e\_phnum

5.51.1.12 Elf32\_Half elf32\_hdr::e\_shentsize

5.51.1.13 Elf32\_Half elf32\_hdr::e\_shnum

5.51.1.14 Elf32\_Half elf32\_hdr::e\_shstrndx

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

- [cpu/common/elf.h](#)

## 5.52 elf32\_note Struct Reference

```
#include <elf.h>
```

### Data Fields

- [Elf32\\_Word n\\_namesz](#)
- [Elf32\\_Word n\\_descsz](#)
- [Elf32\\_Word n\\_type](#)

### 5.52.1 Field Documentation

**5.52.1.1** [Elf32\\_Word elf32\\_note::n\\_namesz](#)

**5.52.1.2** [Elf32\\_Word elf32\\_note::n\\_descsz](#)

**5.52.1.3** [Elf32\\_Word elf32\\_note::n\\_type](#)

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

- [cpu/common/elf.h](#)

## 5.53 elf32\_phdr Struct Reference

```
#include <elf.h>
```

### Data Fields

- [Elf32\\_Word p\\_type](#)
- [Elf32\\_Off p\\_offset](#)
- [Elf32\\_Addr p\\_vaddr](#)
- [Elf32\\_Addr p\\_paddr](#)
- [Elf32\\_Word p\\_filesz](#)
- [Elf32\\_Word p\\_memsz](#)
- [Elf32\\_Word p\\_flags](#)
- [Elf32\\_Word p\\_align](#)

### 5.53.1 Field Documentation

**5.53.1.1 Elf32\_Word elf32\_phdr::p\_type**

**5.53.1.2 Elf32\_Off elf32\_phdr::p\_offset**

**5.53.1.3 Elf32\_Addr elf32\_phdr::p\_vaddr**

**5.53.1.4 Elf32\_Addr elf32\_phdr::p\_paddr**

**5.53.1.5 Elf32\_Word elf32\_phdr::p\_filesz**

**5.53.1.6 Elf32\_Word elf32\_phdr::p\_memsz**

**5.53.1.7 Elf32\_Word elf32\_phdr::p\_flags**

**5.53.1.8 Elf32\_Word elf32\_phdr::p\_align**

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

- [cpu/common/elf.h](#)

## 5.54 elf32\_rel Struct Reference

```
#include <elf.h>
```

### Data Fields

- [Elf32\\_Addr r\\_offset](#)
- [Elf32\\_Word r\\_info](#)

### 5.54.1 Field Documentation

#### 5.54.1.1 Elf32\_Addr elf32\_rel::r\_offset

#### 5.54.1.2 Elf32\_Word elf32\_rel::r\_info

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

- [cpu/common/elf.h](#)

## 5.55 elf32\_rela Struct Reference

```
#include <elf.h>
```

### Data Fields

- [Elf32\\_Addr r\\_offset](#)
- [Elf32\\_Word r\\_info](#)
- [Elf32\\_Sword r\\_addend](#)

### 5.55.1 Field Documentation

**5.55.1.1 Elf32\_Addr elf32\_rela::r\_offset**

**5.55.1.2 Elf32\_Word elf32\_rela::r\_info**

**5.55.1.3 Elf32\_Sword elf32\_rela::r\_addend**

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

- [cpu/common/elf.h](#)



## 5.56 elf32\_shdr Struct Reference

```
#include <elf.h>
```

### Data Fields

- [Elf32\\_Word sh\\_name](#)
- [Elf32\\_Word sh\\_type](#)
- [Elf32\\_Word sh\\_flags](#)
- [Elf32\\_Addr sh\\_addr](#)
- [Elf32\\_Off sh\\_offset](#)
- [Elf32\\_Word sh\\_size](#)
- [Elf32\\_Word sh\\_link](#)
- [Elf32\\_Word sh\\_info](#)
- [Elf32\\_Word sh\\_addralign](#)
- [Elf32\\_Word sh\\_entsize](#)

### 5.56.1 Field Documentation

**5.56.1.1** [Elf32\\_Word elf32\\_shdr::sh\\_name](#)

**5.56.1.2** [Elf32\\_Word elf32\\_shdr::sh\\_type](#)

**5.56.1.3** [Elf32\\_Word elf32\\_shdr::sh\\_flags](#)

**5.56.1.4** [Elf32\\_Addr elf32\\_shdr::sh\\_addr](#)

**5.56.1.5** [Elf32\\_Off elf32\\_shdr::sh\\_offset](#)

**5.56.1.6** [Elf32\\_Word elf32\\_shdr::sh\\_size](#)

**5.56.1.7** [Elf32\\_Word elf32\\_shdr::sh\\_link](#)

**5.56.1.8** [Elf32\\_Word elf32\\_shdr::sh\\_info](#)

**5.56.1.9** [Elf32\\_Word elf32\\_shdr::sh\\_addralign](#)

**5.56.1.10** [Elf32\\_Word elf32\\_shdr::sh\\_entsize](#)

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

- [cpu/common/elf.h](#)

## 5.57 elf32\_sym Struct Reference

```
#include <elf.h>
```

### Data Fields

- [Elf32\\_Word st\\_name](#)
- [Elf32\\_Addr st\\_value](#)
- [Elf32\\_Word st\\_size](#)
- unsigned char [st\\_info](#)
- unsigned char [st\\_other](#)
- [Elf32\\_Half st\\_shndx](#)

### 5.57.1 Field Documentation

**5.57.1.1** [Elf32\\_Word elf32\\_sym::st\\_name](#)

**5.57.1.2** [Elf32\\_Addr elf32\\_sym::st\\_value](#)

**5.57.1.3** [Elf32\\_Word elf32\\_sym::st\\_size](#)

**5.57.1.4** [unsigned char elf32\\_sym::st\\_info](#)

**5.57.1.5** [unsigned char elf32\\_sym::st\\_other](#)

**5.57.1.6** [Elf32\\_Half elf32\\_sym::st\\_shndx](#)

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

- [cpu/common/elf.h](#)

## 5.58 Elf64\_Dyn Struct Reference

```
#include <elf.h>
```

### Data Fields

- unsigned long long [d\\_tag](#)
- union {
  - unsigned long long [d\\_val](#)
  - unsigned long long [d\\_ptr](#)
- } [d\\_un](#)

### 5.58.1 Field Documentation

**5.58.1.1** unsigned long long Elf64\_Dyn::d\_tag

**5.58.1.2** unsigned long long Elf64\_Dyn::d\_val

**5.58.1.3** unsigned long long Elf64\_Dyn::d\_ptr

**5.58.1.4** union { ... } Elf64\_Dyn::d\_un

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

- [cpu/common/elf.h](#)

## 5.59 elf64\_hdr Struct Reference

```
#include <elf.h>
```

### Data Fields

- unsigned char [e\\_ident](#) [16]
- short int [e\\_type](#)
- short unsigned int [e\\_machine](#)
- int [e\\_version](#)
- unsigned long long [e\\_entry](#)
- unsigned long long [e\\_phoff](#)
- unsigned long long [e\\_shoff](#)
- int [e\\_flags](#)
- short int [e\\_ehsize](#)
- short int [e\\_phentsize](#)
- short int [e\\_phnum](#)
- short int [e\\_shentsize](#)
- short int [e\\_shnum](#)
- short int [e\\_shstrndx](#)

## 5.59.1 Field Documentation

- 5.59.1.1 unsigned char elf64\_hdr::e\_ident[16]
- 5.59.1.2 short int elf64\_hdr::e\_type
- 5.59.1.3 short unsigned int elf64\_hdr::e\_machine
- 5.59.1.4 int elf64\_hdr::e\_version
- 5.59.1.5 unsigned long long elf64\_hdr::e\_entry
- 5.59.1.6 unsigned long long elf64\_hdr::e\_phoff
- 5.59.1.7 unsigned long long elf64\_hdr::e\_shoff
- 5.59.1.8 int elf64\_hdr::e\_flags
- 5.59.1.9 short int elf64\_hdr::e\_ehsize
- 5.59.1.10 short int elf64\_hdr::e\_phentsize
- 5.59.1.11 short int elf64\_hdr::e\_phnum
- 5.59.1.12 short int elf64\_hdr::e\_shentsize
- 5.59.1.13 short int elf64\_hdr::e\_shnum
- 5.59.1.14 short int elf64\_hdr::e\_shstrndx

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

- [cpu/common/elf.h](#)

## 5.60 elf64\_note Struct Reference

```
#include <elf.h>
```

### Data Fields

- unsigned int [n\\_namesz](#)
- unsigned int [n\\_descsz](#)
- unsigned int [n\\_type](#)

### 5.60.1 Field Documentation

**5.60.1.1** unsigned int `elf64_note::n_namesz`

**5.60.1.2** unsigned int `elf64_note::n_descsz`

**5.60.1.3** unsigned int `elf64_note::n_type`

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

- [cpu/common/elf.h](#)

## 5.61 elf64\_phdr Struct Reference

```
#include <elf.h>
```

### Data Fields

- int [p\\_type](#)
- int [p\\_flags](#)
- unsigned long long [p\\_offset](#)
- unsigned long long [p\\_vaddr](#)
- unsigned long long [p\\_paddr](#)
- unsigned long long [p\\_filesz](#)
- unsigned long long [p\\_memsz](#)
- unsigned long long [p\\_align](#)

### 5.61.1 Field Documentation

**5.61.1.1 int elf64\_phdr::p\_type**

**5.61.1.2 int elf64\_phdr::p\_flags**

**5.61.1.3 unsigned long long elf64\_phdr::p\_offset**

**5.61.1.4 unsigned long long elf64\_phdr::p\_vaddr**

**5.61.1.5 unsigned long long elf64\_phdr::p\_paddr**

**5.61.1.6 unsigned long long elf64\_phdr::p\_filesz**

**5.61.1.7 unsigned long long elf64\_phdr::p\_memsz**

**5.61.1.8 unsigned long long elf64\_phdr::p\_align**

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

- [cpu/common/elf.h](#)

## 5.62 elf64\_rel Struct Reference

```
#include <elf.h>
```

### Data Fields

- unsigned long long [r\\_offset](#)
- unsigned long long [r\\_info](#)

### 5.62.1 Field Documentation

#### 5.62.1.1 unsigned long long elf64\_rel::r\_offset

#### 5.62.1.2 unsigned long long elf64\_rel::r\_info

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

- [cpu/common/elf.h](#)



## 5.63 elf64\_rela Struct Reference

```
#include <elf.h>
```

### Data Fields

- unsigned long long [r\\_offset](#)
- unsigned long long [r\\_info](#)
- unsigned long long [r\\_addend](#)

### 5.63.1 Field Documentation

**5.63.1.1** unsigned long long `elf64_rela::r_offset`

**5.63.1.2** unsigned long long `elf64_rela::r_info`

**5.63.1.3** unsigned long long `elf64_rela::r_addend`

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

- [cpu/common/elf.h](#)

## 5.64 elf64\_shdr Struct Reference

```
#include <elf.h>
```

### Data Fields

- unsigned int [sh\\_name](#)
- unsigned int [sh\\_type](#)
- unsigned long long [sh\\_flags](#)
- unsigned long long [sh\\_addr](#)
- unsigned long long [sh\\_offset](#)
- unsigned long long [sh\\_size](#)
- unsigned int [sh\\_link](#)
- unsigned int [sh\\_info](#)
- unsigned long long [sh\\_addralign](#)
- unsigned long long [sh\\_entsize](#)

### 5.64.1 Field Documentation

**5.64.1.1** unsigned int `elf64_shdr::sh_name`

**5.64.1.2** unsigned int `elf64_shdr::sh_type`

**5.64.1.3** unsigned long long `elf64_shdr::sh_flags`

**5.64.1.4** unsigned long long `elf64_shdr::sh_addr`

**5.64.1.5** unsigned long long `elf64_shdr::sh_offset`

**5.64.1.6** unsigned long long `elf64_shdr::sh_size`

**5.64.1.7** unsigned int `elf64_shdr::sh_link`

**5.64.1.8** unsigned int `elf64_shdr::sh_info`

**5.64.1.9** unsigned long long `elf64_shdr::sh_addralign`

**5.64.1.10** unsigned long long `elf64_shdr::sh_entsize`

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

- `cpu/common/elf.h`

## 5.65 elf64\_sym Struct Reference

```
#include <elf.h>
```

### Data Fields

- unsigned int [st\\_name](#)
- unsigned char [st\\_info](#)
- unsigned char [st\\_other](#)
- unsigned short [st\\_shndx](#)
- unsigned long long [st\\_value](#)
- unsigned long long [st\\_size](#)

### 5.65.1 Field Documentation

**5.65.1.1** unsigned int `elf64_sym::st_name`

**5.65.1.2** unsigned char `elf64_sym::st_info`

**5.65.1.3** unsigned char `elf64_sym::st_other`

**5.65.1.4** unsigned short `elf64_sym::st_shndx`

**5.65.1.5** unsigned long long `elf64_sym::st_value`

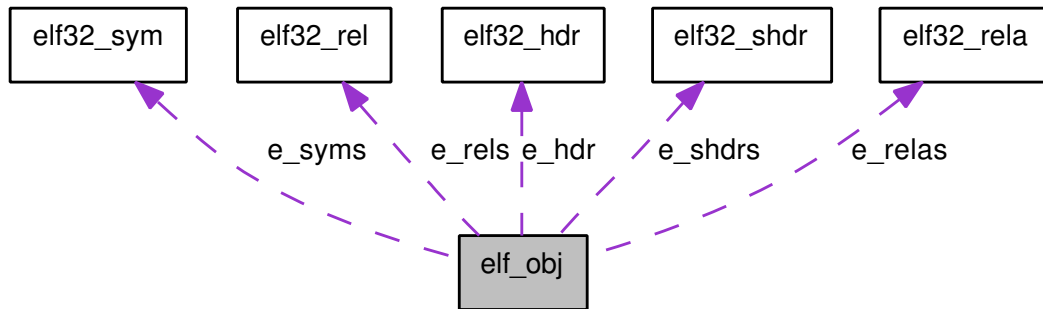
**5.65.1.6** unsigned long long `elf64_sym::st_size`

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

- [cpu/common/elf.h](#)

## 5.66 elf\_obj Struct Reference

Collaboration diagram for elf\_obj:



### Data Fields

- [Elf32\\_Ehdr](#) e\_hdr
- [Elf32\\_Shdr](#) \* e\_shdrs
- void \*\* e\_sections
- [Elf32\\_Sym](#) \* e\_syms
- unsigned int e\_sym\_num
- unsigned int e\_sym\_str\_tab
- [Elf32\\_Rel](#) \* e\_rels
- unsigned int e\_rel\_num
- unsigned int e\_rel\_sym
- unsigned int e\_rel\_sec
- [Elf32\\_Rela](#) \* e\_relas
- unsigned int e\_rela\_num
- unsigned int e\_rela\_sym
- unsigned int e\_rela\_sec

## 5.66.1 Field Documentation

5.66.1.1 Elf32\_Ehdr elf\_obj::e\_hdr

5.66.1.2 Elf32\_Shdr\* elf\_obj::e\_shdrs

5.66.1.3 void\*\* elf\_obj::e\_sections

5.66.1.4 Elf32\_Sym\* elf\_obj::e\_syms

5.66.1.5 unsigned int elf\_obj::e\_sym\_num

5.66.1.6 unsigned int elf\_obj::e\_sym\_str\_tab

5.66.1.7 Elf32\_Rel\* elf\_obj::e\_rels

5.66.1.8 unsigned int elf\_obj::e\_rel\_num

5.66.1.9 unsigned int elf\_obj::e\_rel\_sym

5.66.1.10 unsigned int elf\_obj::e\_rel\_sec

5.66.1.11 Elf32\_Rela\* elf\_obj::e\_relas

5.66.1.12 unsigned int elf\_obj::e\_rela\_num

5.66.1.13 unsigned int elf\_obj::e\_rela\_sym

5.66.1.14 unsigned int elf\_obj::e\_rela\_sec

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

- [cpu/or32/dyngen-elf.c](#)

## 5.67 eth\_device Struct Reference

### Data Fields

- int `enabled`
- `oraddr_t` `baseaddr`
- unsigned `dma`
- unsigned `tx_channel`
- unsigned `rx_channel`
- unsigned char `mac_address` [ETHER\_ADDR\_LEN]
- unsigned long `mac_int`
- unsigned long `base_vapi_id`
- char \* `rxfile`
- char \* `txfile`
- int `txfd`
- int `rxfd`
- `off_t` `loopback_offset`
- char \* `sockif`
- int `rtx_sock`
- int `rtx_type`
- struct ifreq `ifr`
- `fd_set` `rfd`s
- `fd_set` `wfd`s
- struct {
  - unsigned long `state`
  - unsigned long `bd_index`
  - unsigned long `bd`
  - unsigned long `bd_addr`
  - unsigned `working`
  - unsigned `waiting_for_dma`
  - unsigned `error`
  - long `packet_length`
  - unsigned `minimum_length`
  - unsigned `maximum_length`
  - unsigned `add_crc`
  - unsigned `crc_dly`
  - unsigned long `crc_value`
  - long `bytes_left`
  - long `bytes_sent`
- } `tx`
- struct {
  - unsigned long `state`
  - unsigned long `bd_index`
  - unsigned long `bd`
  - unsigned long `bd_addr`
  - int `fd`
  - `off_t` \* `offset`
  - unsigned `working`
  - unsigned `error`
  - unsigned `waiting_for_dma`
  - long `packet_length`

```
 long bytes_read
 long bytes_left
} rx
```

- struct {
    - unsigned long moder
    - unsigned long int\_source
    - unsigned long int\_mask
    - unsigned long ipgt
    - unsigned long ipgr1
    - unsigned long ipgr2
    - unsigned long packetlen
    - unsigned long collconf
    - unsigned long tx\_bd\_num
    - unsigned long controlmoder
    - unsigned long miimoder
    - unsigned long miicommand
    - unsigned long miiaddress
    - unsigned long miitx\_data
    - unsigned long miirx\_data
    - unsigned long miistatus
    - unsigned long hash0
    - unsigned long hash1
    - unsigned long bd\_ram [ETH\_BD\_SPACE/4]
- ```
} regs
```

- unsigned char rx_buff [ETH_MAXPL]

- unsigned char tx_buff [ETH_MAXPL]

- unsigned char lo_buff [ETH_MAXPL]

5.67.1 Field Documentation

5.67.1.1 int eth_device::enabled

5.67.1.2 oraddr_t eth_device::baseaddr

5.67.1.3 unsigned eth_device::dma

5.67.1.4 unsigned eth_device::tx_channel

5.67.1.5 unsigned eth_device::rx_channel

5.67.1.6 unsigned char eth_device::mac_address[ETHER_ADDR_LEN]

5.67.1.7 unsigned long eth_device::mac_int

5.67.1.8 unsigned long eth_device::base_vapi_id

5.67.1.9 char* eth_device::rxfile

5.67.1.10 char * eth_device::txfile

5.67.1.11 int eth_device::txfd

5.67.1.12 int eth_device::rxfd

5.67.1.13 off_t eth_device::loopback_offset

5.67.1.14 char* eth_device::sockif

5.67.1.15 int eth_device::rtx_sock

5.67.1.16 int eth_device::rtx_type

5.67.1.17 struct ifreq eth_device::ifr [read]

5.67.1.18 fd_set eth_device::rfd

5.67.1.19 fd_set eth_device::wfd

5.67.1.20 unsigned long eth_device::state

5.67.1.21 unsigned long eth_device::bd_index

5.67.1.22 unsigned long eth_device::bd

5.67.1.23 unsigned long eth_device::bd_addr

5.67.1.24 unsigned eth_device::working

5.67.1.25 unsigned eth_device::waiting_for_dma

5.67.1.26 unsigned eth_device::error

5.67.1.27 long eth_device::packet_length

5.67.1.28 unsigned eth_device::minimum_length

5.67.1.29 unsigned eth_device::maximum_length

5.67.1.30 unsigned eth_device::add_crc

- [peripheral/eth.c](#)

5.68 ether_addr Struct Reference

Data Fields

- `u_int8_t ether_addr_octet` [ETH_ALEN]

5.68.1 Field Documentation

5.68.1.1 `u_int8_t ether_addr::ether_addr_octet`[ETH_ALEN]

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

- `peripheral/eth.c`

5.69 ether_header Struct Reference

Data Fields

- `u_int8_t ether_dhost` [ETH_ALEN]
- `u_int8_t ether_shost` [ETH_ALEN]
- `u_int16_t ether_type`

5.69.1 Field Documentation

5.69.1.1 `u_int8_t ether_header::ether_dhost`[ETH_ALEN]

5.69.1.2 `u_int8_t ether_header::ether_shost`[ETH_ALEN]

5.69.1.3 `u_int16_t ether_header::ether_type`

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

- `peripheral/eth.c`

5.70 fb_state Struct Reference

Data Fields

- int `enabled`
- unsigned long `pal` [256]
- int `ctrl`
- int `pic`
- int `in_refresh`
- int `refresh_count`
- `oraddr_t` `addr`
- `oraddr_t` `cam_addr`
- int `camerax`
- int `cameray`
- int `camera_pos`
- `oraddr_t` `baseaddr`
- int `refresh`
- int `refresh_rate`
- char * `filename`

5.70.1 Field Documentation

5.70.1.1 int fb_state::enabled

5.70.1.2 unsigned long fb_state::pal[256]

5.70.1.3 int fb_state::ctrl

5.70.1.4 int fb_state::pic

5.70.1.5 int fb_state::in_refresh

5.70.1.6 int fb_state::refresh_count

5.70.1.7 oraddr_t fb_state::addr

5.70.1.8 oraddr_t fb_state::cam_addr

5.70.1.9 int fb_state::camerax

5.70.1.10 int fb_state::cameray

5.70.1.11 int fb_state::camera_pos

5.70.1.12 oraddr_t fb_state::baseaddr

5.70.1.13 int fb_state::refresh

5.70.1.14 int fb_state::refresh_rate

5.70.1.15 char* fb_state::filename

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

- [peripheral/fb.c](#)

5.71 fd_channel Struct Reference

```
#include <fd.h>
```

Data Fields

- int [fdin](#)
- int [fdout](#)

5.71.1 Detailed Description

Data structure to represent a [channel](#) through file descriptors

5.71.2 Field Documentation

5.71.2.1 int fd_channel::fdin

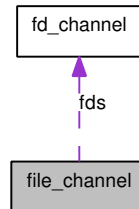
5.71.2.2 int fd_channel::fdout

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

- [peripheral/channels/fd.h](#)

5.72 file_channel Struct Reference

Collaboration diagram for file_channel:



Data Fields

- struct `fd_channel` `fds`
- char * `namein`
- char * `nameout`

5.72.1 Detailed Description

Data structure representing a [channel](#) to/from a file

5.72.2 Field Documentation

5.72.2.1 struct `fd_channel` `file_channel::fds` [`read`]

5.72.2.2 char* `file_channel::namein`

5.72.2.3 char* `file_channel::nameout`

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

- [peripheral/channels/file.c](#)

5.73 fstats_entry Struct Reference

Data Fields

- enum insn_type [insn1](#)
- enum insn_type [insn2](#)
- int [cnt_dynamic](#)
- int [depend](#)

5.73.1 Field Documentation

5.73.1.1 enum insn_type fstats_entry::insn1

5.73.1.2 enum insn_type fstats_entry::insn2

5.73.1.3 int fstats_entry::cnt_dynamic

5.73.1.4 int fstats_entry::depend

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

- [cpu/common/stats.c](#)

5.74 func_struct Struct Reference

```
#include <profiler.h>
```

Data Fields

- unsigned int [addr](#)
- char [name](#) [33]
- long [cum_cycles](#)
- long [calls](#)

5.74.1 Detailed Description

Data structure for information about functions

5.74.2 Field Documentation

5.74.2.1 unsigned int func_struct::addr

Start address of function

5.74.2.2 char func_struct::name[33]

Name of the function

5.74.2.3 long func_struct::cum_cycles

Total cycles spent in function

5.74.2.4 long func_struct::calls

Calls to this function

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

- [profiler.h](#)

5.75 gpio_device Struct Reference

Data Fields

- int `enabled`
- `oraddr_t` `baseaddr`
- int `irq`
- unsigned `gpio_number`
- unsigned long `base_vapi_id`
- unsigned long `auxiliary_inputs`
- struct {
 - unsigned long `in`
 - unsigned long `out`
 - unsigned long `oe`
 - unsigned long `inte`
 - unsigned long `ptrig`
 - unsigned long `aux`
 - unsigned long `ctrl`
 - unsigned long `ints`
 - int `external_clock`
- struct {
 - unsigned long `in`
 - unsigned long `out`
 - unsigned long `oe`
 - unsigned long `inte`
 - unsigned long `ptrig`
 - unsigned long `aux`
 - unsigned long `ctrl`
 - unsigned long `ints`
 - int `external_clock`

5.75.1 Field Documentation

- 5.75.1.1 `int gpio_device::enabled`
- 5.75.1.2 `oraddr_t gpio_device::baseaddr`
- 5.75.1.3 `int gpio_device::irq`
- 5.75.1.4 `unsigned gpio_device::gpio_number`
- 5.75.1.5 `unsigned long gpio_device::base_vapi_id`
- 5.75.1.6 `unsigned long gpio_device::auxiliary_inputs`
- 5.75.1.7 `unsigned long gpio_device::in`
- 5.75.1.8 `unsigned long gpio_device::out`
- 5.75.1.9 `unsigned long gpio_device::oe`
- 5.75.1.10 `unsigned long gpio_device::inte`
- 5.75.1.11 `unsigned long gpio_device::ptrig`
- 5.75.1.12 `unsigned long gpio_device::aux`
- 5.75.1.13 `unsigned long gpio_device::ctrl`
- 5.75.1.14 `unsigned long gpio_device::ints`
- 5.75.1.15 `int gpio_device::external_clock`
- 5.75.1.16 `struct { ... } gpio_device::curr`
- 5.75.1.17 `struct { ... } gpio_device::next`

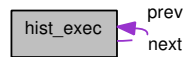
The documentation for this struct was generated from the following file:

- [peripheral/gpio.c](#)

5.76 hist_exec Struct Reference

```
#include <execute.h>
```

Collaboration diagram for hist_exec:



Data Fields

- [oraddr_t addr](#)
- struct [hist_exec](#) * [prev](#)
- struct [hist_exec](#) * [next](#)

5.76.1 Detailed Description

History of execution

5.76.2 Field Documentation

5.76.2.1 oraddr_t hist_exec::addr

5.76.2.2 struct [hist_exec](#)* [hist_exec::prev](#) [read]

5.76.2.3 struct [hist_exec](#)* [hist_exec::next](#) [read]

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

- [cpu/common/execute.h](#)

5.77 ic Struct Reference

```
#include <icache-model.h>
```

Data Fields

- uint8_t * [mem](#)
- unsigned int * [lrus](#)
- [oraddr_t](#) * [tags](#)
- int [enabled](#)
- unsigned int [nways](#)
- unsigned int [nsets](#)
- unsigned int [blocksize](#)
- unsigned int [ustates](#)
- int [missdelay](#)
- int [hitdelay](#)
- unsigned int [blocksize_log2](#)
- [oraddr_t](#) [set_mask](#)
- [oraddr_t](#) [tagaddr_mask](#)
- [oraddr_t](#) [last_way](#)
- [oraddr_t](#) [block_offset_mask](#)
- [oraddr_t](#) [block_mask](#)
- unsigned int [ustates_reload](#)

5.77.1 Field Documentation

5.77.1.1 `uint8_t* ic::mem`

5.77.1.2 `unsigned int* ic::lrus`

5.77.1.3 `oraddr_t* ic::tags`

5.77.1.4 `int ic::enabled`

5.77.1.5 `unsigned int ic::nways`

5.77.1.6 `unsigned int ic::nsets`

5.77.1.7 `unsigned int ic::blocksize`

5.77.1.8 `unsigned int ic::ustates`

5.77.1.9 `int ic::missdelay`

5.77.1.10 `int ic::hitdelay`

5.77.1.11 `unsigned int ic::blocksize_log2`

5.77.1.12 `oraddr_t ic::set_mask`

5.77.1.13 `oraddr_t ic::tagaddr_mask`

5.77.1.14 `oraddr_t ic::last_way`

5.77.1.15 `oraddr_t ic::block_offset_mask`

5.77.1.16 `oraddr_t ic::block_mask`

5.77.1.17 `unsigned int ic::ustates_reload`

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

- [cache/icache-model.h](#)

5.78 immu Struct Reference

```
#include <immu.h>
```

Data Fields

- int `enabled`
- int `nways`
- int `nsets`
- `oraddr_t` `pagesize`
- int `pagesize_log2`
- `oraddr_t` `page_offset_mask`
- `oraddr_t` `page_mask`
- `oraddr_t` `vpn_mask`
- int `lru_reload`
- `oraddr_t` `set_mask`
- int `entrysize`
- int `ustates`
- int `missdelay`
- int `hitdelay`

5.78.1 Field Documentation

5.78.1.1 `int immu::enabled`

5.78.1.2 `int immu::nways`

5.78.1.3 `int immu::nsets`

5.78.1.4 `oraddr_t immu::pagesize`

5.78.1.5 `int immu::pagesize_log2`

5.78.1.6 `oraddr_t immu::page_offset_mask`

5.78.1.7 `oraddr_t immu::page_mask`

5.78.1.8 `oraddr_t immu::vpn_mask`

5.78.1.9 `int immu::lru_reload`

5.78.1.10 `oraddr_t immu::set_mask`

5.78.1.11 `int immu::entrysize`

5.78.1.12 `int immu::ustates`

5.78.1.13 `int immu::missdelay`

5.78.1.14 `int immu::hitdelay`

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

- [mmu/immu.h](#)

5.79 immustats_entry Struct Reference

```
#include <stats.h>
```

Data Fields

- int [fetch_tlbhit](#)
- int [fetch_tlbmiss](#)
- int [fetch_pagefaults](#)

5.79.1 Field Documentation

5.79.1.1 int immustats_entry::fetch_tlbhit

5.79.1.2 int immustats_entry::fetch_tlbmiss

5.79.1.3 int immustats_entry::fetch_pagefaults

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

- [cpu/common/stats.h](#)

5.80 INFOHEADER Struct Reference

Data Fields

- unsigned int [size](#)
- int [width](#)
- int [height](#)
- unsigned short int [planes](#)
- unsigned short int [bits](#)
- unsigned int [compression](#)
- unsigned int [imagesize](#)
- int [xresolution](#)
- int [yresolution](#)
- unsigned int [ncolours](#)
- unsigned int [importantcolours](#)

5.80.1 Field Documentation

5.80.1.1 unsigned int INFOHEADER::size

5.80.1.2 int INFOHEADER::width

5.80.1.3 int INFOHEADER::height

5.80.1.4 unsigned short int INFOHEADER::planes

5.80.1.5 unsigned short int INFOHEADER::bits

5.80.1.6 unsigned int INFOHEADER::compression

5.80.1.7 unsigned int INFOHEADER::imagesize

5.80.1.8 int INFOHEADER::xresolution

5.80.1.9 int INFOHEADER::yresolution

5.80.1.10 unsigned int INFOHEADER::ncolours

5.80.1.11 unsigned int INFOHEADER::importantcolours

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

- [peripheral/vga.c](#)

5.81 iqueue_entry Struct Reference

```
#include <abstract.h>
```

Data Fields

- int [insn_index](#)
- uint32_t [insn](#)
- oraddr_t [insn_addr](#)

5.81.1 Detailed Description

Instruction queue

5.81.2 Field Documentation

5.81.2.1 int iqueue_entry::insn_index

5.81.2.2 uint32_t iqueue_entry::insn

5.81.2.3 oraddr_t iqueue_entry::insn_addr

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

- [cpu/common/abstract.h](#)

5.82 jtr_chain_message Struct Reference

```
#include <gdb.h>
```

Data Fields

- [uint32_t command](#)
- [uint32_t length](#)
- [uint32_t chain](#)

5.82.1 Field Documentation

5.82.1.1 [uint32_t jtr_chain_message::command](#)

5.82.1.2 [uint32_t jtr_chain_message::length](#)

5.82.1.3 [uint32_t jtr_chain_message::chain](#)

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

- [debug/gdb.h](#)

5.83 jtr_chain_response Struct Reference

```
#include <gdb.h>
```

Data Fields

- [int32_t status](#)

5.83.1 Field Documentation

5.83.1.1 int32_t jtr_chain_response::status

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

- [debug/gdb.h](#)

5.84 `jtr_failure_response` Struct Reference

```
#include <gdb.h>
```

Data Fields

- `int32_t status`

5.84.1 Field Documentation

5.84.1.1 `int32_t jtr_failure_response::status`

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

- `debug/gdb.h`

5.85 jtr_read_block_message Struct Reference

```
#include <gdb.h>
```

Data Fields

- uint32_t [command](#)
- uint32_t [length](#)
- uint32_t [address](#)
- int32_t [num_regs](#)

5.85.1 Field Documentation

5.85.1.1 uint32_t jtr_read_block_message::command

5.85.1.2 uint32_t jtr_read_block_message::length

5.85.1.3 uint32_t jtr_read_block_message::address

5.85.1.4 int32_t jtr_read_block_message::num_regs

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

- [debug/gdb.h](#)

5.86 jtr_read_block_response Struct Reference

```
#include <gdb.h>
```

Data Fields

- [int32_t status](#)
- [int32_t num_regs](#)
- [uint32_t data](#) [1]

5.86.1 Field Documentation

5.86.1.1 [int32_t jtr_read_block_response::status](#)

5.86.1.2 [int32_t jtr_read_block_response::num_regs](#)

5.86.1.3 [uint32_t jtr_read_block_response::data\[1\]](#)

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

- [debug/gdb.h](#)

5.87 jtr_read_message Struct Reference

```
#include <gdb.h>
```

Data Fields

- [uint32_t command](#)
- [uint32_t length](#)
- [uint32_t address](#)

5.87.1 Field Documentation

5.87.1.1 uint32_t jtr_read_message::command

5.87.1.2 uint32_t jtr_read_message::length

5.87.1.3 uint32_t jtr_read_message::address

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

- [debug/gdb.h](#)

5.88 jtr_read_response Struct Reference

```
#include <gdb.h>
```

Data Fields

- [int32_t status](#)
- [uint32_t data_h](#)
- [uint32_t data_l](#)

5.88.1 Field Documentation

5.88.1.1 [int32_t jtr_read_response::status](#)

5.88.1.2 [uint32_t jtr_read_response::data_h](#)

5.88.1.3 [uint32_t jtr_read_response::data_l](#)

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

- [debug/gdb.h](#)

5.89 jtr_write_block_message Struct Reference

```
#include <gdb.h>
```

Data Fields

- [uint32_t command](#)
- [uint32_t length](#)
- [uint32_t address](#)
- [int32_t num_regs](#)
- [uint32_t data \[1\]](#)

5.89.1 Field Documentation

5.89.1.1 [uint32_t jtr_write_block_message::command](#)

5.89.1.2 [uint32_t jtr_write_block_message::length](#)

5.89.1.3 [uint32_t jtr_write_block_message::address](#)

5.89.1.4 [int32_t jtr_write_block_message::num_regs](#)

5.89.1.5 [uint32_t jtr_write_block_message::data\[1\]](#)

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

- [debug/gdb.h](#)

5.90 jtr_write_block_response Struct Reference

```
#include <gdb.h>
```

Data Fields

- [int32_t status](#)

5.90.1 Field Documentation

5.90.1.1 int32_t jtr_write_block_response::status

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

- [debug/gdb.h](#)

5.91 jtr_write_message Struct Reference

```
#include <gdb.h>
```

Data Fields

- [uint32_t command](#)
- [uint32_t length](#)
- [uint32_t address](#)
- [uint32_t data_h](#)
- [uint32_t data_l](#)

5.91.1 Field Documentation

5.91.1.1 `uint32_t jtr_write_message::command`

5.91.1.2 `uint32_t jtr_write_message::length`

5.91.1.3 `uint32_t jtr_write_message::address`

5.91.1.4 `uint32_t jtr_write_message::data_h`

5.91.1.5 `uint32_t jtr_write_message::data_l`

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

- [debug/gdb.h](#)

5.92 jtr_write_response Struct Reference

```
#include <gdb.h>
```

Data Fields

- [int32_t status](#)

5.92.1 Field Documentation

5.92.1.1 int32_t jtr_write_response::status

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

- [debug/gdb.h](#)

5.93 kbd_state Struct Reference

Data Fields

- uint8_t [buf](#) [KBD_MAX_BUF]
- unsigned long [buf_count](#)
- unsigned long [buf_head](#)
- unsigned long [buf_tail](#)
- FILE * [rxfs](#)
- int [ccmd](#)
- uint8_t [kcmd](#)
- uint8_t [ccmdbyte](#)
- unsigned long [kresp](#)
- long [slowdown](#)
- int [enabled](#)
- int [irq](#)
- oraddr_t [baseaddr](#)
- char * [rxfile](#)

5.93.1 Field Documentation

5.93.1.1 uint8_t kbd_state::buf[KBD_MAX_BUF]

5.93.1.2 unsigned long kbd_state::buf_count

5.93.1.3 unsigned long kbd_state::buf_head

5.93.1.4 unsigned long kbd_state::buf_tail

5.93.1.5 FILE* kbd_state::rxfs

5.93.1.6 int kbd_state::ccmd

5.93.1.7 uint8_t kbd_state::kcmd

5.93.1.8 uint8_t kbd_state::ccmdbyte

5.93.1.9 unsigned long kbd_state::kresp

5.93.1.10 long kbd_state::slowdown

5.93.1.11 int kbd_state::enabled

5.93.1.12 int kbd_state::irq

5.93.1.13 oraddr_t kbd_state::baseaddr

5.93.1.14 char* kbd_state::rxfile

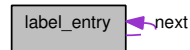
The documentation for this struct was generated from the following file:

- [peripheral/ps2kbd.c](#)

5.94 label_entry Struct Reference

```
#include <labels.h>
```

Collaboration diagram for label_entry:



Data Fields

- char * [name](#)
- [oraddr_t](#) [addr](#)
- struct [label_entry](#) * [next](#)

5.94.1 Detailed Description

Structure for holding one label per particular memory location

5.94.2 Field Documentation

5.94.2.1 char* [label_entry::name](#)

5.94.2.2 [oraddr_t](#) [label_entry::addr](#)

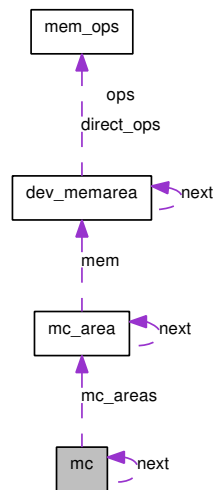
5.94.2.3 struct [label_entry](#)* [label_entry::next](#) [read]

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

- [cpu/common/labels.h](#)

5.95 mc Struct Reference

Collaboration diagram for mc:



Data Fields

- `uint32_t csr`
- `uint32_t poc`
- `uint32_t ba_mask`
- `uint32_t csc [N_CE]`
- `uint32_t tms [N_CE]`
- `oraddr_t baseaddr`
- `int enabled`
- `int index`
- `struct mc_area * mc_areas`
- `struct mc * next`

5.95.1 Field Documentation

5.95.1.1 `uint32_t mc::csr`

5.95.1.2 `uint32_t mc::poc`

5.95.1.3 `uint32_t mc::ba_mask`

5.95.1.4 `uint32_t mc::csc[N_CE]`

5.95.1.5 `uint32_t mc::tms[N_CE]`

5.95.1.6 `oraddr_t mc::baseaddr`

5.95.1.7 `int mc::enabled`

5.95.1.8 `int mc::index`

5.95.1.9 `struct mc_area* mc::mc_areas` [read]

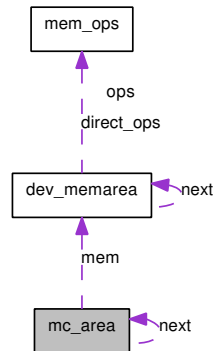
5.95.1.10 `struct mc* mc::next` [read]

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

- [peripheral/mc.c](#)

5.96 mc_area Struct Reference

Collaboration diagram for mc_area:



Data Fields

- struct [dev_memarea](#) * [mem](#)
- unsigned int [cs](#)
- int [mc](#)
- struct [mc_area](#) * [next](#)

5.96.1 Field Documentation

5.96.1.1 struct [dev_memarea](#)* [mc_area::mem](#) [read]

5.96.1.2 unsigned int [mc_area::cs](#)

5.96.1.3 int [mc_area::mc](#)

5.96.1.4 struct [mc_area](#)* [mc_area::next](#) [read]

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

- [peripheral/mc.c](#)

5.97 mem_config Struct Reference

Public Types

- enum { [MT_UNKNOWN](#), [MT_PATTERN](#), [MT_RANDOM](#) }

Data Fields

- int [ce](#)
- int [mc](#)
- [oraddr_t](#) [baseaddr](#)
- unsigned int [size](#)
- char * [name](#)
- char * [log](#)
- int [delayr](#)
- int [delayw](#)
- void * [mem](#)
- int [pattern](#)
- int [random_seed](#)
- enum mem_config:: { ... } [type](#)

5.97.1 Member Enumeration Documentation

5.97.1.1 anonymous enum

Enumerator:

MT_UNKNOWN

MT_PATTERN

MT_RANDOM

5.97.2 Field Documentation

5.97.2.1 `int mem_config::ce`

5.97.2.2 `int mem_config::mc`

5.97.2.3 `oraddr_t mem_config::baseaddr`

5.97.2.4 `unsigned int mem_config::size`

5.97.2.5 `char* mem_config::name`

5.97.2.6 `char* mem_config::log`

5.97.2.7 `int mem_config::delayr`

5.97.2.8 `int mem_config::delayw`

5.97.2.9 `void* mem_config::mem`

5.97.2.10 `int mem_config::pattern`

5.97.2.11 `int mem_config::random_seed`

5.97.2.12 `enum { ... } mem_config::type`

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

- [peripheral/memory.c](#)

5.98 mem_ops Struct Reference

```
#include <abstract.h>
```

Data Fields

- uint32_t(* [readfunc32](#))(oraddr_t, void *)
- uint16_t(* [readfunc16](#))(oraddr_t, void *)
- uint8_t(* [readfunc8](#))(oraddr_t, void *)
- void * [read_dat8](#)
- void * [read_dat16](#)
- void * [read_dat32](#)
- void(* [writefunc32](#))(oraddr_t, uint32_t, void *)
- void(* [writefunc16](#))(oraddr_t, uint16_t, void *)
- void(* [writefunc8](#))(oraddr_t, uint8_t, void *)
- void * [write_dat8](#)
- void * [write_dat16](#)
- void * [write_dat32](#)
- void(* [writeprog32](#))(oraddr_t, uint32_t, void *)
- void(* [writeprog8](#))(oraddr_t, uint8_t, void *)
- void * [writeprog32_dat](#)
- void * [writeprog8_dat](#)
- int [delayr](#)
- int [delayw](#)
- const char * [log](#)

5.98.1 Detailed Description

All the memory operations possible

5.98.2 Field Documentation

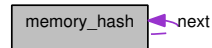
- 5.98.2.1 `uint32_t(* mem_ops::readfunc32)(oraddr_t, void *)`
- 5.98.2.2 `uint16_t(* mem_ops::readfunc16)(oraddr_t, void *)`
- 5.98.2.3 `uint8_t(* mem_ops::readfunc8)(oraddr_t, void *)`
- 5.98.2.4 `void* mem_ops::read_dat8`
- 5.98.2.5 `void* mem_ops::read_dat16`
- 5.98.2.6 `void* mem_ops::read_dat32`
- 5.98.2.7 `void(* mem_ops::writefunc32)(oraddr_t, uint32_t, void *)`
- 5.98.2.8 `void(* mem_ops::writefunc16)(oraddr_t, uint16_t, void *)`
- 5.98.2.9 `void(* mem_ops::writefunc8)(oraddr_t, uint8_t, void *)`
- 5.98.2.10 `void* mem_ops::write_dat8`
- 5.98.2.11 `void* mem_ops::write_dat16`
- 5.98.2.12 `void* mem_ops::write_dat32`
- 5.98.2.13 `void(* mem_ops::writeprog32)(oraddr_t, uint32_t, void *)`
- 5.98.2.14 `void(* mem_ops::writeprog8)(oraddr_t, uint8_t, void *)`
- 5.98.2.15 `void* mem_ops::writeprog32_dat`
- 5.98.2.16 `void* mem_ops::writeprog8_dat`
- 5.98.2.17 `int mem_ops::delayr`
- 5.98.2.18 `int mem_ops::delayw`
- 5.98.2.19 `const char* mem_ops::log`

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

- [cpu/common/abstract.h](#)

5.99 memory_hash Struct Reference

Collaboration diagram for memory_hash:



Data Fields

- struct [memory_hash](#) * [next](#)
- [oraddr_t](#) [addr](#)
- unsigned long [cnt](#) [3]

5.99.1 Detailed Description

Hash table data structure

5.99.2 Field Documentation

5.99.2.1 struct [memory_hash](#)* [memory_hash::next](#) [read]

5.99.2.2 [oraddr_t](#) [memory_hash::addr](#)

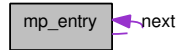
5.99.2.3 unsigned long [memory_hash::cnt](#)[3]

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

- [mprofiler.c](#)

5.100 mp_entry Struct Reference

Collaboration diagram for mp_entry:



Data Fields

- enum [mp_type](#) `type`
- unsigned long int [addr](#)
- unsigned long int [instr](#)
- struct [mp_entry](#) * `next`

5.100.1 Detailed Description

Data structure for a matchpoint hash table entry

5.100.2 Field Documentation

5.100.2.1 enum mp_type mp_entry::type

Type of matchpoint

5.100.2.2 unsigned long int mp_entry::addr

Address with the matchpoint

5.100.2.3 unsigned long int mp_entry::instr

Substituted instruction

5.100.2.4 struct mp_entry* mp_entry::next [read]

Next entry with this hash

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

- [debug/rsp-server.c](#)

5.101 mprofentry_struct Struct Reference

```
#include <profile.h>
```

Data Fields

- [oraddr_t addr](#)
- unsigned char [type](#)

5.101.1 Field Documentation

5.101.1.1 oraddr_t mprofentry_struct::addr

5.101.1.2 unsigned char mprofentry_struct::type

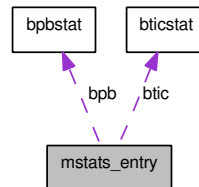
The documentation for this struct was generated from the following file:

- support/[profile.h](#)

5.102 mstats_entry Struct Reference

```
#include <stats.h>
```

Collaboration diagram for mstats_entry:



Data Fields

- int [byteadd](#)
- int [bf](#) [2][2]
- int [bnf](#) [2][2]
- struct [bpbstat](#) [bpb](#)
- struct [bticstat](#) [btic](#)

5.102.1 Field Documentation

5.102.1.1 int [mstats_entry::byteadd](#)

5.102.1.2 int [mstats_entry::bf](#)[2][2]

5.102.1.3 int [mstats_entry::bnf](#)[2][2]

5.102.1.4 struct [bpbstat](#) [mstats_entry::bpb](#) [read]

5.102.1.5 struct [bticstat](#) [mstats_entry::btic](#) [read]

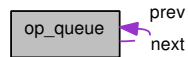
The documentation for this struct was generated from the following file:

- [cpu/common/stats.h](#)

5.103 op_queue Struct Reference

```
#include <dyn32-defs.h>
```

Collaboration diagram for op_queue:



Data Fields

- unsigned int [num_ops](#)
- unsigned int [ops_len](#)
- unsigned int * [ops](#)
- unsigned int [num_ops_param](#)
- unsigned int [ops_param_len](#)
- unsigned int * [ops_param](#)
- unsigned int [jump_local](#)
- [oraddr_t](#) [jump_local_loc](#)
- unsigned int [not_jump_loc](#)
- int [xref](#)
- [oraddr_t](#) [insn_addr](#)
- unsigned int [reg_t](#) [3]
- unsigned int [tflags](#) [3]
- int [insn_index](#)
- unsigned int [param_type](#) [5]
- [orreg_t](#) [param](#) [5]
- unsigned int [param_num](#)
- uint32_t [insn](#)
- struct [op_queue](#) * [prev](#)
- struct [op_queue](#) * [next](#)

5.103.1 Field Documentation

- 5.103.1.1 unsigned int op_queue::num_ops
- 5.103.1.2 unsigned int op_queue::ops_len
- 5.103.1.3 unsigned int* op_queue::ops
- 5.103.1.4 unsigned int op_queue::num_ops_param
- 5.103.1.5 unsigned int op_queue::ops_param_len
- 5.103.1.6 unsigned int* op_queue::ops_param
- 5.103.1.7 unsigned int op_queue::jump_local
- 5.103.1.8 oraddr_t op_queue::jump_local_loc
- 5.103.1.9 unsigned int op_queue::not_jump_loc
- 5.103.1.10 int op_queue::xref
- 5.103.1.11 oraddr_t op_queue::insn_addr
- 5.103.1.12 unsigned int op_queue::reg_t[3]
- 5.103.1.13 unsigned int op_queue::tflags[3]
- 5.103.1.14 int op_queue::insn_index
- 5.103.1.15 unsigned int op_queue::param_type[5]
- 5.103.1.16 orreg_t op_queue::param[5]
- 5.103.1.17 unsigned int op_queue::param_num
- 5.103.1.18 uint32_t op_queue::insn
- 5.103.1.19 struct op_queue* op_queue::prev [read]
- 5.103.1.20 struct op_queue* op_queue::next [read]

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

- [cpu/or32/dyn32-defs.h](#)

5.104 param_val Union Reference

```
#include <sim-config.h>
```

Data Fields

- char * [str_val](#)
- int [int_val](#)
- long long int [longlong_val](#)
- [oraddr_t](#) [addr_val](#)

5.104.1 Detailed Description

Union of all possible paramter values

5.104.2 Field Documentation

5.104.2.1 char* param_val::str_val

5.104.2.2 int param_val::int_val

5.104.2.3 long long int param_val::longlong_val

5.104.2.4 oraddr_t param_val::addr_val

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

- [sim-config.h](#)

5.105 raw_stats Struct Reference

```
#include <stats.h>
```

Data Fields

- int [reg](#) [64]
- int [range](#) [RAW_RANGE]

5.105.1 Field Documentation

5.105.1.1 int raw_stats::reg[64]

5.105.1.2 int raw_stats::range[RAW_RANGE]

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

- [cpu/common/stats.h](#)

5.106 reloc Struct Reference

```
#include <dyngen.h>
```

Data Fields

- unsigned int [func_offset](#)
- unsigned int [addend](#)
- int [type](#)
- const char * [name](#)

5.106.1 Field Documentation

5.106.1.1 unsigned int `reloc::func_offset`

5.106.1.2 unsigned int `reloc::addend`

5.106.1.3 int `reloc::type`

5.106.1.4 const char* `reloc::name`

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

- [cpu/or32/dyngen.h](#)

5.107 `rsp_buf` Struct Reference

Data Fields

- char `data` [GDB_BUF_MAX]
- int `len`

5.107.1 Detailed Description

Data structure for RSP buffers. Can't be null terminated, since it may include zero bytes

5.107.2 Field Documentation

5.107.2.1 char `rsp_buf::data`[GDB_BUF_MAX]

5.107.2.2 int `rsp_buf::len`

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

- [debug/rsp-server.c](#)

5.108 runtime Struct Reference

```
#include <sim-config.h>
```

Data Fields

- struct {
 - FILE * [fprof](#)
 - FILE * [fmprof](#)
 - FILE * [fexe_log](#)
 - FILE * [fout](#)
 - char * [filename](#)
 - int [iprompt](#)
 - int [iprompt_run](#)
 - long long [cycles](#)
 - long long int [end_cycles](#)
 - double [time_point](#)
 - unsigned long int [ext_int](#)
 - int [mem_cycles](#)
 - int [loadcycles](#)
 - int [storecycles](#)
 - long long [reset_cycles](#)
 - int [hush](#)
- } [sim](#)
- struct {
 - long long [instructions](#)
 - long long [reset_instructions](#)
 - int [stalled](#)
 - int [hazardwait](#)
 - int [supercycles](#)
- } [cpu](#)
- struct {
 - int [enabled](#)
 - FILE * [vapi_file](#)
 - int [server_port](#)
- } [vapi](#)
- struct {
 - int [mdelay](#) [4]
 - double [cycle_duration](#)
- } [cuc](#)

5.108.1 Detailed Description

Data structure for run time data

5.108.2 Field Documentation

5.108.2.1 FILE* runtime::fprof

5.108.2.2 FILE* runtime::fmprof

5.108.2.3 FILE* runtime::fexe_log

5.108.2.4 FILE* runtime::fout

5.108.2.5 char* runtime::filename

5.108.2.6 int runtime::iprompt

5.108.2.7 int runtime::iprompt_run

5.108.2.8 long long runtime::cycles

5.108.2.9 long long int runtime::end_cycles

5.108.2.10 double runtime::time_point

5.108.2.11 unsigned long int runtime::ext_int

5.108.2.12 int runtime::mem_cycles

5.108.2.13 int runtime::loadcycles

5.108.2.14 int runtime::storecycles

5.108.2.15 long long runtime::reset_cycles

5.108.2.16 int runtime::hush

5.108.2.17 struct { ... } runtime::sim

5.108.2.18 long long runtime::instructions

5.108.2.19 long long runtime::reset_instructions

5.108.2.20 int runtime::stalled

5.108.2.21 int runtime::hazardwait

5.108.2.22 int runtime::supercycles

5.108.2.23 struct { ... } runtime::cpu

5.108.2.24 int runtime::enabled

5.108.2.25 FILE* runtime::vapi_file

5.108.2.26 int runtime::server_port

5.108.2.27 struct { ... } runtime::xapi

5.108.2.28 int runtime::mdelay[4]

5.108.2.29 double runtime::cycle_duration

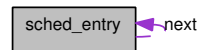
5.108.2.30 struct { ... } runtime::cuc

- [sim-config.h](#)

5.109 sched_entry Struct Reference

```
#include <sched.h>
```

Collaboration diagram for sched_entry:



Data Fields

- [int32_t time](#)
- [void * param](#)
- [void\(* func\)\(void *\)](#)
- [struct sched_entry * next](#)

5.109.1 Detailed Description

Structure for holding one job entry

5.109.2 Field Documentation

5.109.2.1 [int32_t sched_entry::time](#)

5.109.2.2 [void* sched_entry::param](#)

5.109.2.3 [void\(* sched_entry::func\)\(void *\)](#)

5.109.2.4 [struct sched_entry* sched_entry::next](#) [read]

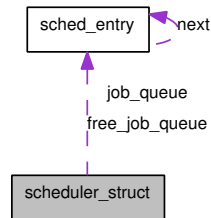
The documentation for this struct was generated from the following file:

- [support/sched.h](#)

5.110 scheduler_struct Struct Reference

```
#include <sched.h>
```

Collaboration diagram for scheduler_struct:



Data Fields

- struct `sched_entry` * `free_job_queue`
- struct `sched_entry` * `job_queue`

5.110.1 Detailed Description

Heap of jobs

5.110.2 Field Documentation

5.110.2.1 struct `sched_entry`* `scheduler_struct::free_job_queue` [read]

5.110.2.2 struct `sched_entry`* `scheduler_struct::job_queue` [read]

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

- support/[sched.h](#)

5.111 `sim_command` Struct Reference

Data Fields

- `const char * name`
- `int(* cmd_handle)(int argc, char **argv)`

5.111.1 Field Documentation

5.111.1.1 `const char* sim_command::name`

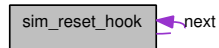
5.111.1.2 `int(* sim_command::cmd_handle)(int argc, char **argv)`

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

- [sim-cmd.c](#)

5.112 `sim_reset_hook` Struct Reference

Collaboration diagram for `sim_reset_hook`:



Data Fields

- `void * dat`
- `void(* reset_hook)(void *)`
- `struct sim_reset_hook * next`

5.112.1 Detailed Description

Struct for list of reset hooks

5.112.2 Field Documentation

5.112.2.1 `void* sim_reset_hook::dat`

5.112.2.2 `void(* sim_reset_hook::reset_hook)(void *)`

5.112.2.3 `struct sim_reset_hook* sim_reset_hook::next` [read]

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

- [toplevel-support.c](#)

5.113 sim_stat Struct Reference

Collaboration diagram for sim_stat:



Data Fields

- void(* [stat_func](#))(void *[dat](#))
- void * [dat](#)
- struct [sim_stat](#) * [next](#)

5.113.1 Field Documentation

5.113.1.1 void(* [sim_stat::stat_func](#))(void *[dat](#))

5.113.1.2 void* [sim_stat::dat](#)

5.113.1.3 struct [sim_stat](#)* [sim_stat::next](#) [read]

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

- [sim-cmd.c](#)

5.114 spr_bit_def Struct Reference

Data Fields

- `const char * name`
- `uorreg_t mask`

5.114.1 Field Documentation

5.114.1.1 `const char* spr_bit_def::name`

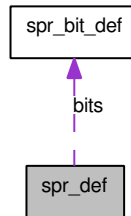
5.114.1.2 `uorreg_t spr_bit_def::mask`

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

- `cpu/or1k/spr-dump.c`

5.115 spr_def Struct Reference

Collaboration diagram for spr_def:



Data Fields

- [uint16_t from_spr](#)
- [uint16_t to_spr](#)
- [const char * name](#)
- [struct spr_bit_def * bits](#)

5.115.1 Field Documentation

5.115.1.1 [uint16_t spr_def::from_spr](#)

5.115.1.2 [uint16_t spr_def::to_spr](#)

5.115.1.3 [const char* spr_def::name](#)

5.115.1.4 [struct spr_bit_def* spr_def::bits](#) [read]

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

- [cpu/or1k/spr-dump.c](#)

5.116 sstats_entry Struct Reference

Data Fields

- int [insn](#)
- int [cnt_dynamic](#)

5.116.1 Field Documentation

5.116.1.1 int sstats_entry::insn

5.116.1.2 int sstats_entry::cnt_dynamic

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

- [cpu/common/stats.c](#)

5.117 `stack_struct` Struct Reference

Data Fields

- unsigned int [addr](#)
- unsigned int [cycles](#)
- unsigned int [raddr](#)
- char [name](#) [33]

5.117.1 Detailed Description

Data structure representing information about a stack frame

5.117.2 Field Documentation

5.117.2.1 unsigned int `stack_struct::addr`

Function address

5.117.2.2 unsigned int `stack_struct::cycles`

Cycles of func start; subfuncs added later

5.117.2.3 unsigned int `stack_struct::raddr`

Return address

5.117.2.4 char `stack_struct::name[33]`

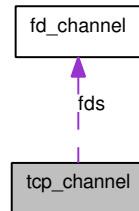
Name of the function

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

- [profiler.c](#)

5.118 tcp_channel Struct Reference

Collaboration diagram for tcp_channel:



Data Fields

- struct [fd_channel](#) `fds`
- int [socket_fd](#)
- int [port_number](#)
- int [connected](#)
- int [nonblocking](#)

5.118.1 Detailed Description

Structure to represent a TCP/IP [channel](#)

5.118.2 Field Documentation

5.118.2.1 struct [fd_channel](#) `tcp_channel::fds` [`read`]

5.118.2.2 int `tcp_channel::socket_fd`

5.118.2.3 int `tcp_channel::port_number`

5.118.2.4 int `tcp_channel::connected`

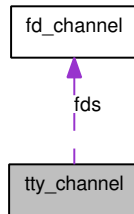
5.118.2.5 int `tcp_channel::nonblocking`

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

- [peripheral/channels/tcp.c](#)

5.119 tty_channel Struct Reference

Collaboration diagram for tty_channel:



Data Fields

- struct [fd_channel](#) `fds`

5.119.1 Detailed Description

Data structure representing a TTY [channel](#)

5.119.2 Field Documentation

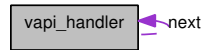
5.119.2.1 struct `fd_channel` `tty_channel::fds` [read]

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

- [peripheral/channels/tty.c](#)

5.120 vapi_handler Struct Reference

Collaboration diagram for vapi_handler:



Data Fields

- int [fd](#)
- unsigned long [base_id](#)
- unsigned long [num_ids](#)
- void(* [read_func](#))(unsigned long, unsigned long, void *)
- void * [priv_dat](#)
- struct [vapi_handler](#) * [next](#)
- int [temp](#)

5.120.1 Field Documentation

5.120.1.1 int `vapi_handler::fd`

5.120.1.2 unsigned long `vapi_handler::base_id`

5.120.1.3 unsigned long `vapi_handler::num_ids`

5.120.1.4 void(* `vapi_handler::read_func`)(unsigned long, unsigned long, void *)

5.120.1.5 void* `vapi_handler::priv_dat`

5.120.1.6 struct `vapi_handler*` `vapi_handler::next` [`read`]

5.120.1.7 int `vapi_handler::temp`

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

- [vapi/vapi.c](#)

5.121 vga_state Struct Reference

Data Fields

- int [enabled](#)
- int [pics](#)
- unsigned long [ctrl](#)
- unsigned long [stat](#)
- unsigned long [htim](#)
- unsigned long [vtim](#)
- int [vbindx](#)
- unsigned long [vbar](#) [2]
- unsigned [hlen](#)
- unsigned [vlen](#)
- int [pindex](#)
- unsigned long [palette](#) [2][256]
- [oraddr_t](#) [baseaddr](#)
- int [refresh_rate](#)
- int [irq](#)
- char * [filename](#)

5.121.1 Field Documentation

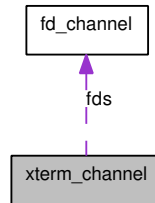
- 5.121.1.1 `int vga_state::enabled`
- 5.121.1.2 `int vga_state::pics`
- 5.121.1.3 `unsigned long vga_state::ctrl`
- 5.121.1.4 `unsigned long vga_state::stat`
- 5.121.1.5 `unsigned long vga_state::htim`
- 5.121.1.6 `unsigned long vga_state::vtim`
- 5.121.1.7 `int vga_state::vbindx`
- 5.121.1.8 `unsigned long vga_state::vbar[2]`
- 5.121.1.9 `unsigned vga_state::hlen`
- 5.121.1.10 `unsigned vga_state::vlen`
- 5.121.1.11 `int vga_state::pindex`
- 5.121.1.12 `unsigned long vga_state::palette[2][256]`
- 5.121.1.13 `oraddr_t vga_state::baseaddr`
- 5.121.1.14 `int vga_state::refresh_rate`
- 5.121.1.15 `int vga_state::irq`
- 5.121.1.16 `char* vga_state::filename`

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

- [peripheral/vga.c](#)

5.122 xterm_channel Struct Reference

Collaboration diagram for xterm_channel:



Data Fields

- struct [fd_channel](#) `fds`
- int `pid`
- char ** `argv`

5.122.1 Detailed Description

Data structure to represent the connection to the xterm

5.122.2 Field Documentation

5.122.2.1 struct `fd_channel` `xterm_channel::fds` [read]

5.122.2.2 int `xterm_channel::pid`

5.122.2.3 char** `xterm_channel::argv`

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

- `peripheral/channels/xterm.c`

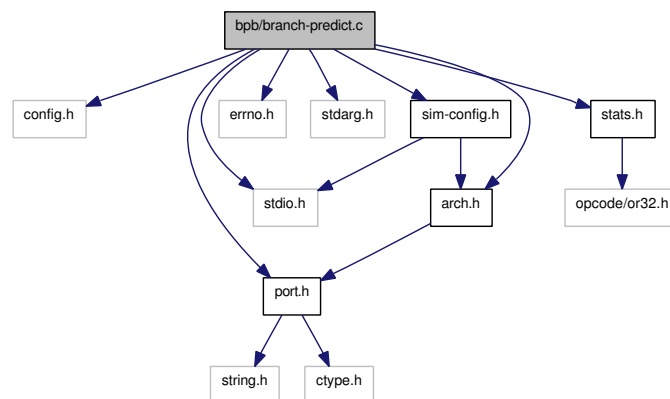
Chapter 6

File Documentation

6.1 bpb/branch-predict.c File Reference

```
#include "config.h"  
#include "port.h"  
#include <stdio.h>  
#include <errno.h>  
#include <stdarg.h>  
#include "sim-config.h"  
#include "arch.h"  
#include "stats.h"
```

Include dependency graph for branch-predict.c:



Data Structures

- struct `bpb_entry`
- struct `bpic_entry`

Defines

- #define `BPB_LEN` 64
- #define `BPB_WAYS` 1
- #define `BPB_PSTATES` 2
- #define `BPB_USTATES` 2
- #define `BTIC_LEN` 128
- #define `BTIC_WAYS` 2
- #define `BTIC_USTATES` 2
- #define `BTIC_BLOCKSIZE` 4

Functions

- void `bpb_info` ()
- void `bpb_update` (`oraddr_t` addr, int taken)
- void `btic_info` ()
- void `btic_update` (`oraddr_t` targetaddr)
- static void `bpb_enabled` (union `param_val` val, void *dat)
- static void `bpb_btic` (union `param_val` val, void *dat)
- static void `bpb_sbp_bnf_fwd` (union `param_val` val, void *dat)
- static void `bpb_sbp_bf_fwd` (union `param_val` val, void *dat)
- static void `bpb_missdelay` (union `param_val` val, void *dat)
- static void `bpb_hitdelay` (union `param_val` val, void *dat)
- void `reg_bpb_sec` ()

Variables

- struct `bpb_entry` `bpb` [`BPB_LEN`]
- struct `btic_entry` `btic` [`BTIC_LEN`]

6.1.1 Define Documentation

6.1.1.1 `#define BPB_LEN 64`

6.1.1.2 `#define BPB_PSTATES 2`

6.1.1.3 `#define BPB_USTATES 2`

6.1.1.4 `#define BPB_WAYS 1`

6.1.1.5 `#define BTIC_BLOCKSIZE 4`

6.1.1.6 `#define BTIC_LEN 128`

6.1.1.7 `#define BTIC_USTATES 2`

6.1.1.8 `#define BTIC_WAYS 2`

6.1.2 Function Documentation

6.1.2.1 `static void bpb_btict (union param_val val, void * dat) [static]`

6.1.2.2 `static void bpb_enabled (union param_val val, void * dat) [static]`

6.1.2.3 `static void bpb_hitdelay (union param_val val, void * dat) [static]`

6.1.2.4 `void bpb_info ()`

6.1.2.5 `static void bpb_missdelay (union param_val val, void * dat) [static]`

6.1.2.6 `static void bpb_sbp_bf_fwd (union param_val val, void * dat) [static]`

6.1.2.7 `static void bpb_sbp_bnf_fwd (union param_val val, void * dat) [static]`

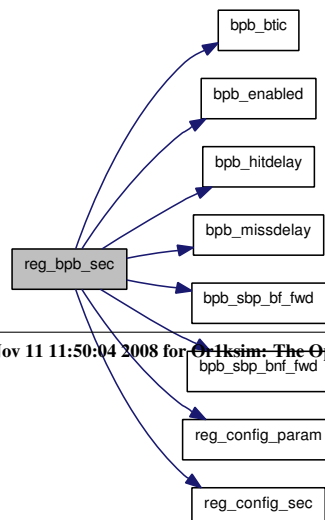
6.1.2.8 `void bpb_update (oraddr_t addr, int taken)`

6.1.2.9 `void btict_info ()`

6.1.2.10 `void btict_update (oraddr_t targetaddr)`

6.1.2.11 `void reg_bpb_sec ()`

Here is the call graph for this function:



6.1.3 Variable Documentation

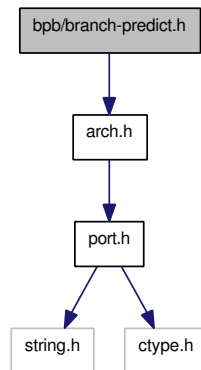
6.1.3.1 struct bpb_entry bpb[BPB_LEN]

6.1.3.2 struct btic_entry btic[BTIC_LEN]

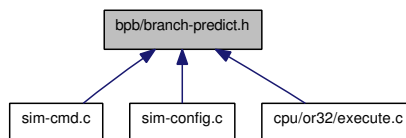
6.2 bpb/branch-predict.h File Reference

```
#include "arch.h"
```

Include dependency graph for branch-predict.h:



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



Functions

- void `bpb_info` ()
- void `bpb_update` (`oraddr_t` addr, int taken)
- void `btic_info` ()
- void `btic_update` (`oraddr_t` targetaddr)
- void `reg_bpb_sec` ()

6.2.1 Function Documentation

6.2.1.1 void `bpb_info` ()

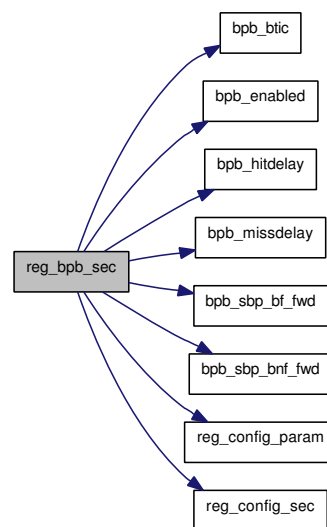
6.2.1.2 void `bpb_update` (`oraddr_t addr`, `int taken`)

6.2.1.3 void `btic_info` ()

6.2.1.4 void `btic_update` (`oraddr_t targetaddr`)

6.2.1.5 void `reg_bpb_sec` ()

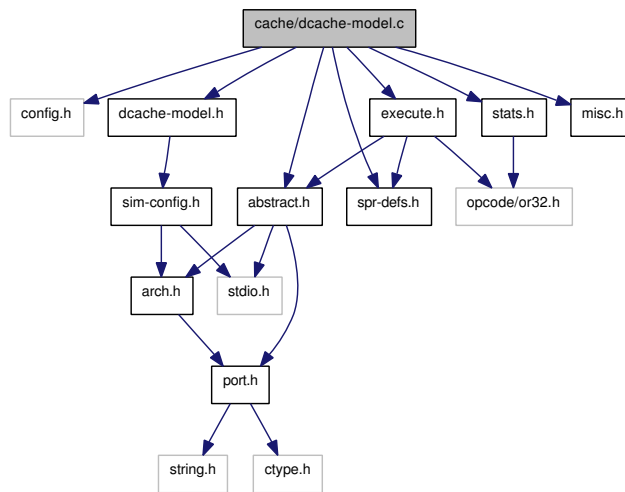
Here is the call graph for this function:



6.3 cache/dcache-model.c File Reference

```
#include "config.h"
#include "dcache-model.h"
#include "execute.h"
#include "spr-defs.h"
#include "abstract.h"
#include "stats.h"
#include "misc.h"
```

Include dependency graph for dcache-model.c:



Data Structures

- struct [dc_set](#)

Functions

- void [dc_info](#) (void)
- uint32_t [dc_simulate_read](#) (oraddr_t dataaddr, oraddr_t virt_addr, int width)
- void [dc_simulate_write](#) (oraddr_t dataaddr, oraddr_t virt_addr, uint32_t data, int width)
- void [dc_inv](#) (oraddr_t dataaddr)
- static void [dc_enabled](#) (union [param_val](#) val, void *dat)
- static void [dc_nsets](#) (union [param_val](#) val, void *dat)
- static void [dc_nways](#) (union [param_val](#) val, void *dat)
- static void [dc_blocksize](#) (union [param_val](#) val, void *dat)
- static void [dc_ustates](#) (union [param_val](#) val, void *dat)
- static void [dc_load_hitdelay](#) (union [param_val](#) val, void *dat)
- static void [dc_load_missdelay](#) (union [param_val](#) val, void *dat)
- static void [dc_store_hitdelay](#) (union [param_val](#) val, void *dat)
- static void [dc_store_missdelay](#) (union [param_val](#) val, void *dat)
- void [reg_dc_sec](#) (void)

Variables

- struct `dc_set dc` [MAX_DC_SETS]

6.3.1 Function Documentation

6.3.1.1 `static void dc_blocksize (union param_val val, void * dat)` [static]

Set the data cache block size

Value must be either MIN_DC_BLOCK_SIZE or MAX_DC_BLOCK_SIZE. If not issue a warning and ignore. Set the relevant field in the data cache `config` register

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure (not used here)

6.3.1.2 `static void dc_enabled (union param_val val, void * dat)` [static]

Enable or disable the data cache

Set the corresponding field in the UPR

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure (not used here)

6.3.1.3 `void dc_info (void)`

6.3.1.4 `void dc_inv (oraddr_t dataaddr)`

6.3.1.5 `static void dc_load_hitdelay (union param_val val, void * dat)` [static]

6.3.1.6 `static void dc_load_missdelay (union param_val val, void * dat)` [static]

6.3.1.7 `static void dc_nsets (union param_val val, void * dat)` [static]

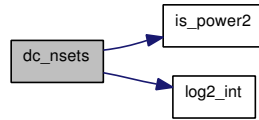
Set the number of data cache sets

Value must be a power of 2 <= MAX_DC_SETS. If not issue a warning and ignore. Set the relevant field in the data cache `config` register

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure (not used here)

Here is the call graph for this function:



6.3.1.8 static void dc_nways (union param_val val, void * dat) [static]

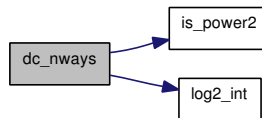
Set the number of data cache ways

Value must be a power of 2 <= MAX_DC_WAYS. If not issue a warning and ignore. Set the relevant field in the data cache [config](#) register

Parameters:

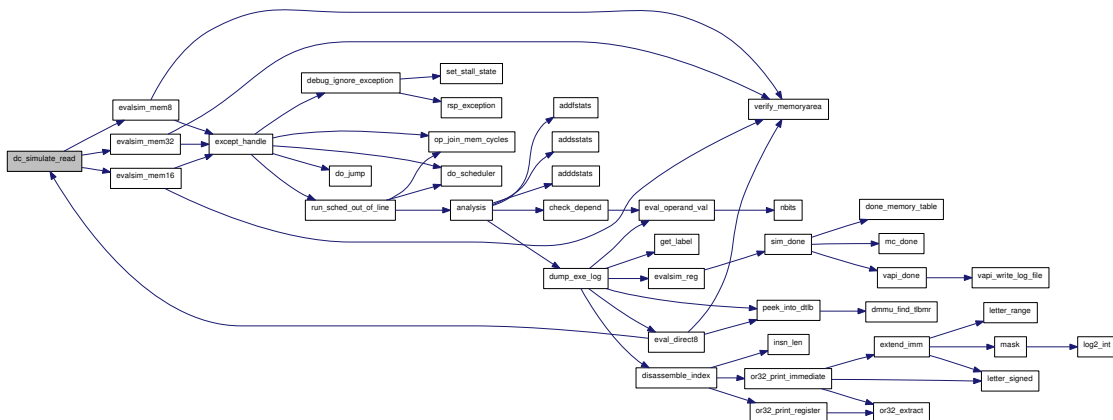
- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

Here is the call graph for this function:



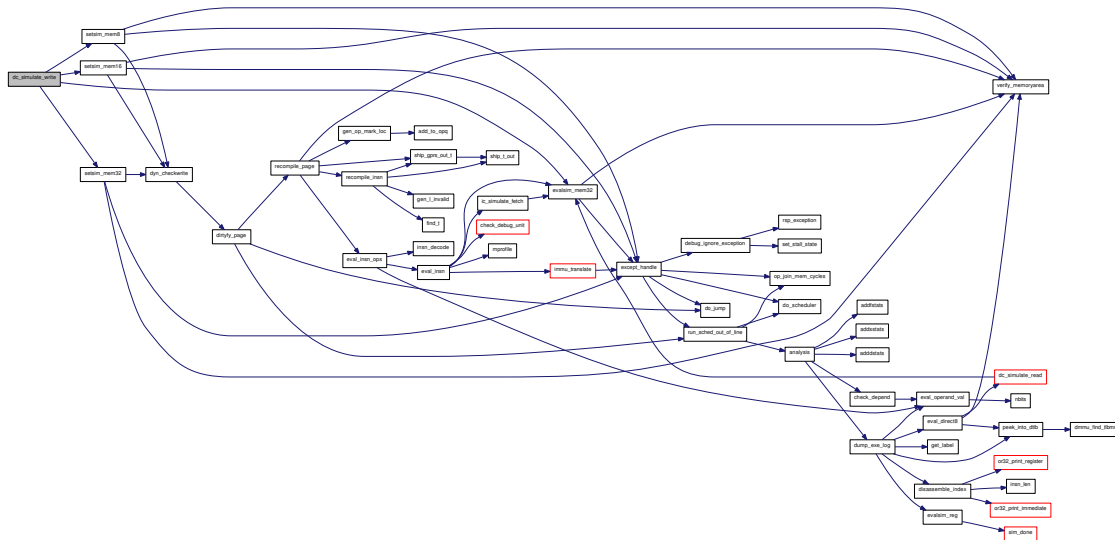
6.3.1.9 uint32_t dc_simulate_read (oraddr_t dataaddr, oraddr_t virt_addr, int width)

Here is the call graph for this function:



6.3.1.10 void dc_simulate_write (oraddr_t dataaddr, oraddr_t virt_addr, uint32_t data, int width)

Here is the call graph for this function:



6.3.1.11 static void dc_store_hitdelay (union param_val val, void * dat) [static]

6.3.1.12 static void dc_store_missdelay (union param_val val, void * dat) [static]

6.3.1.13 static void dc_ustates (union param_val val, void * dat) [static]

Set the number of data cache usage states

Value must be 2, 3 or 4. If not issue a warning and ignore.

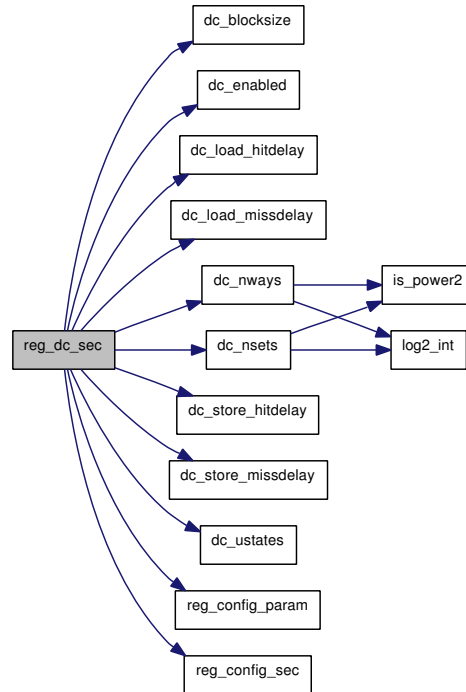
Parameters:

← *val* The value to use

← *dat* The [config](#) data structure (not used here)

6.3.1.14 void reg_dc_sec (void)

Here is the call graph for this function:



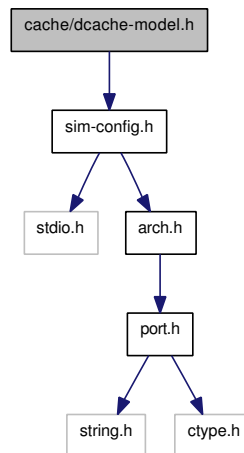
6.3.2 Variable Documentation

6.3.2.1 struct dc_set dc[MAX_DC_SETS]

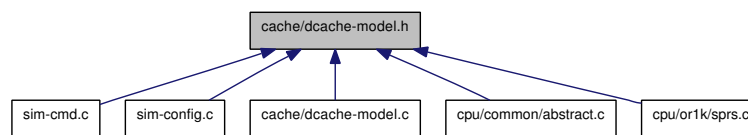
6.4 cache/dcache-model.h File Reference

```
#include "sim-config.h"
```

Include dependency graph for dcache-model.h:



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



Defines

- #define `MAX_DC_SETS` 1024
- #define `MAX_DC_WAYS` 32
- #define `MIN_DC_BLOCK_SIZE` 16
- #define `MAX_DC_BLOCK_SIZE` 32

Functions

- `uint32_t dc_simulate_read (oraddr_t dataaddr, oraddr_t virt_addr, int width)`
- `void dc_simulate_write (oraddr_t dataaddr, oraddr_t virt_addr, uint32_t data, int width)`
- `void dc_info ()`
- `void dc_inv (oraddr_t dataaddr)`
- `void reg_dc_sec ()`

6.4.1 Define Documentation

6.4.1.1 `#define MAX_DC_BLOCK_SIZE 32`

6.4.1.2 `#define MAX_DC_SETS 1024`

6.4.1.3 `#define MAX_DC_WAYS 32`

6.4.1.4 `#define MIN_DC_BLOCK_SIZE 16`

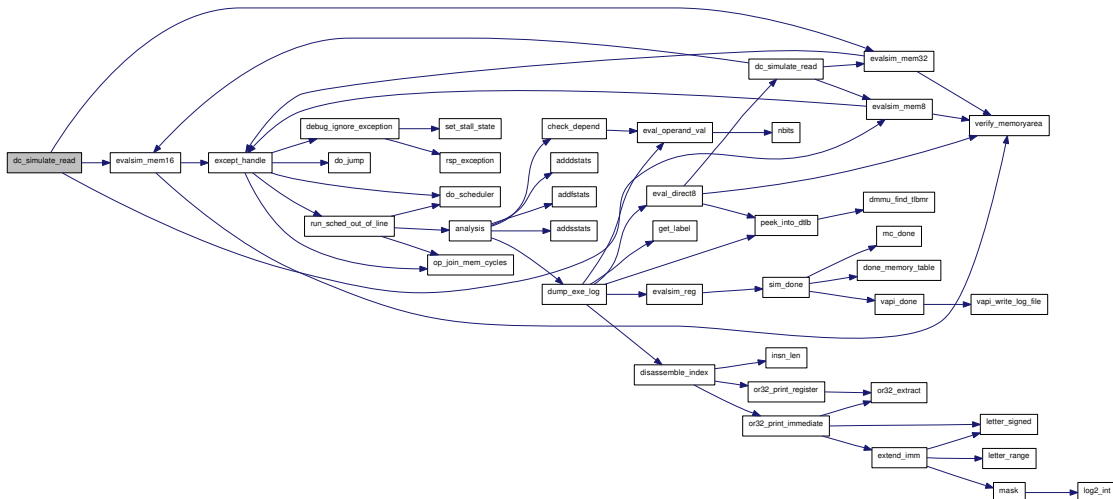
6.4.2 Function Documentation

6.4.2.1 `void dc_info ()`

6.4.2.2 `void dc_inv (oraddr_t dataaddr)`

6.4.2.3 `uint32_t dc_simulate_read (oraddr_t dataaddr, oraddr_t virt_addr, int width)`

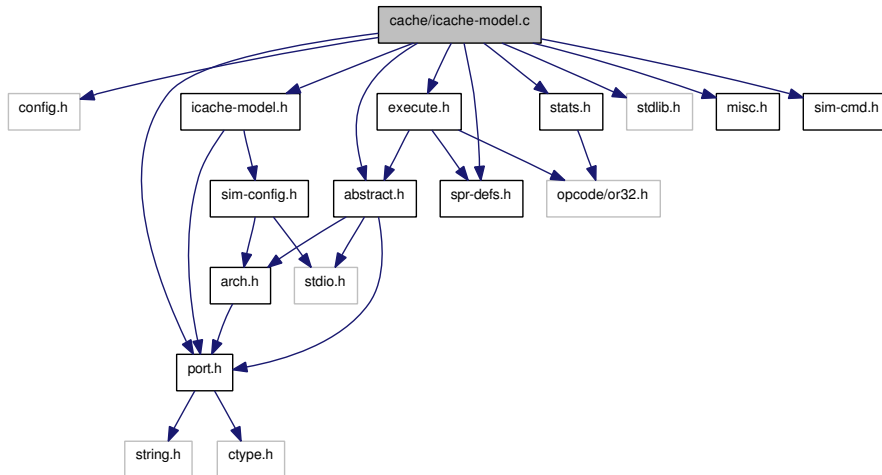
Here is the call graph for this function:



6.5 cache/icache-model.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "icache-model.h"
#include "execute.h"
#include "spr-defs.h"
#include "abstract.h"
#include "misc.h"
#include "stats.h"
#include "sim-cmd.h"
```

Include dependency graph for icache-model.c:



Defines

- #define [MAX_IC_SETS](#) 1024
- #define [MAX_IC_WAYS](#) 32
- #define [MIN_IC_BLOCK_SIZE](#) 16
- #define [MAX_IC_BLOCK_SIZE](#) 32

Functions

- static void [ic_info](#) (void *dat)
- uint32_t [ic_simulate_fetch](#) ([oraddr_t](#) fetchaddr, [oraddr_t](#) virt_addr)
- void [ic_inv](#) ([oraddr_t](#) dataaddr)
- static void [ic_enabled](#) (union [param_val](#) val, void *dat)
- static void [ic_nsets](#) (union [param_val](#) val, void *dat)
- static void [ic_nways](#) (union [param_val](#) val, void *dat)

- static void `ic_blocksize` (union `param_val` *val*, void **dat*)
- static void `ic_ustates` (union `param_val` *val*, void **dat*)
- static void `ic_hitdelay` (union `param_val` *val*, void **dat*)
- static void `ic_missdelay` (union `param_val` *val*, void **dat*)
- static void * `ic_start_sec` ()
- static void `ic_end_sec` (void **dat*)
- void `reg_ic_sec` (void)

Variables

- struct `ic` * `ic_state` = NULL

6.5.1 Define Documentation

6.5.1.1 `#define MAX_IC_BLOCK_SIZE 32`

6.5.1.2 `#define MAX_IC_SETS 1024`

6.5.1.3 `#define MAX_IC_WAYS 32`

6.5.1.4 `#define MIN_IC_BLOCK_SIZE 16`

6.5.2 Function Documentation

6.5.2.1 `static void ic_blocksize (union param_val val, void * dat)` [`static`]

Set the instruction cache block size

Value must be either `MIN_IC_BLOCK_SIZE` or `MAX_IC_BLOCK_SIZE`. If not issue a warning and ignore. Set the relevant field in the data cache `config` register

Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.5.2.2 `static void ic_enabled (union param_val val, void * dat)` [`static`]

Enable or disable the instruction cache

Set the corresponding field in the UPR

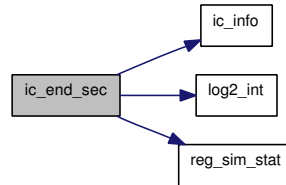
Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.5.2.3 static void ic_end_sec (void * *dat*) [static]

Here is the call graph for this function:



6.5.2.4 static void ic_hitdelay (union param_val *val*, void * *dat*) [static]

6.5.2.5 static void ic_info (void * *dat*) [static]

6.5.2.6 void ic_inv (oraddr_t *dataaddr*)

6.5.2.7 static void ic_missdelay (union param_val *val*, void * *dat*) [static]

6.5.2.8 static void ic_nsets (union param_val *val*, void * *dat*) [static]

Set the number of instruction cache sets

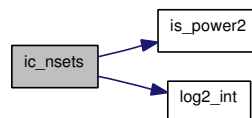
Set the corresponding field in the UPR

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



6.5.2.9 static void ic_nways (union param_val *val*, void * *dat*) [static]

Set the number of instruction cache ways

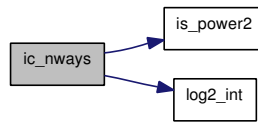
Set the corresponding field in the UPR

Parameters:

← *val* The value to use

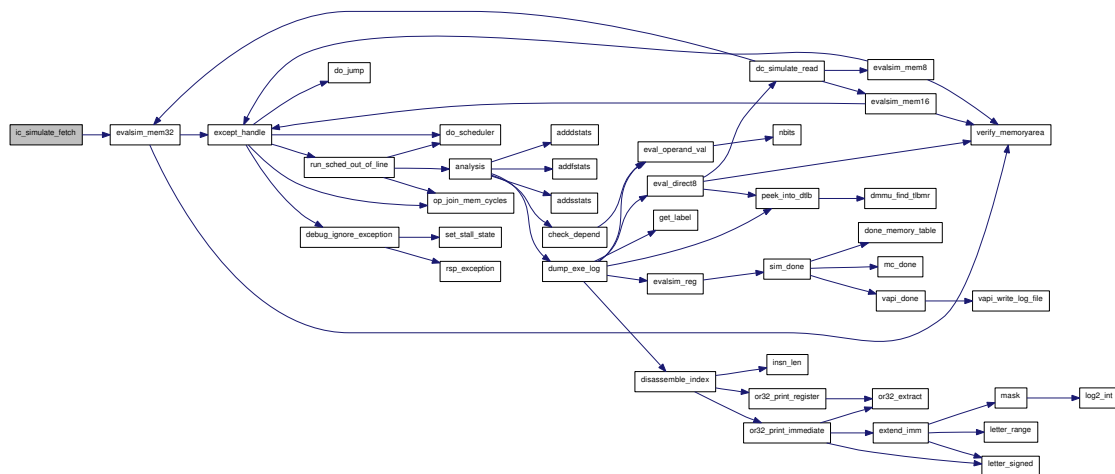
← *dat* The `config` data structure

Here is the call graph for this function:



6.5.2.10 uint32_t ic_simulate_fetch (oraddr_t fetchaddr, oraddr_t virt_addr)

Here is the call graph for this function:



6.5.2.11 static void* ic_start_sec () [static]

Initialize a new instruction cache configuration

ALL parameters are set explicitly to default values. Corresponding SPR flags are set as appropriate.

Returns:

The new memory configuration data structure

Here is the call graph for this function:



6.5.2.12 static void ic_ustates (union param_val val, void * dat) [static]

Set the number of instruction cache usage states

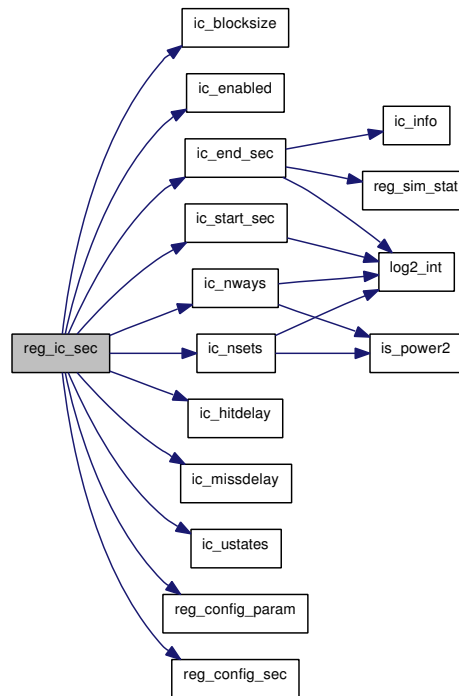
Value must be 2, 3 or 4. If not issue a warning and ignore.

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure

6.5.2.13 void reg_ic_sec (void)

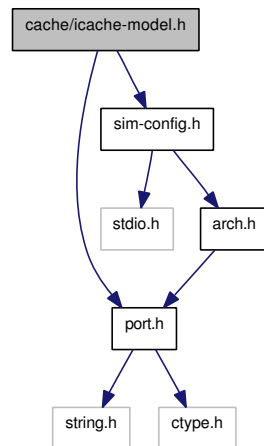
Here is the call graph for this function:

**6.5.3 Variable Documentation****6.5.3.1 struct ic* ic_state = NULL**

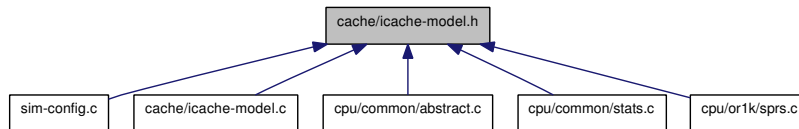
6.6 cache/icache-model.h File Reference

```
#include "port.h"
#include "sim-config.h"
```

Include dependency graph for icache-model.h:



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



Data Structures

- struct `ic`

Functions

- `uint32_t ic_simulate_fetch (oraddr_t fetchaddr, oraddr_t virt_addr)`
- `void ic_inv (oraddr_t dataaddr)`
- `void reg_ic_sec ()`

Variables

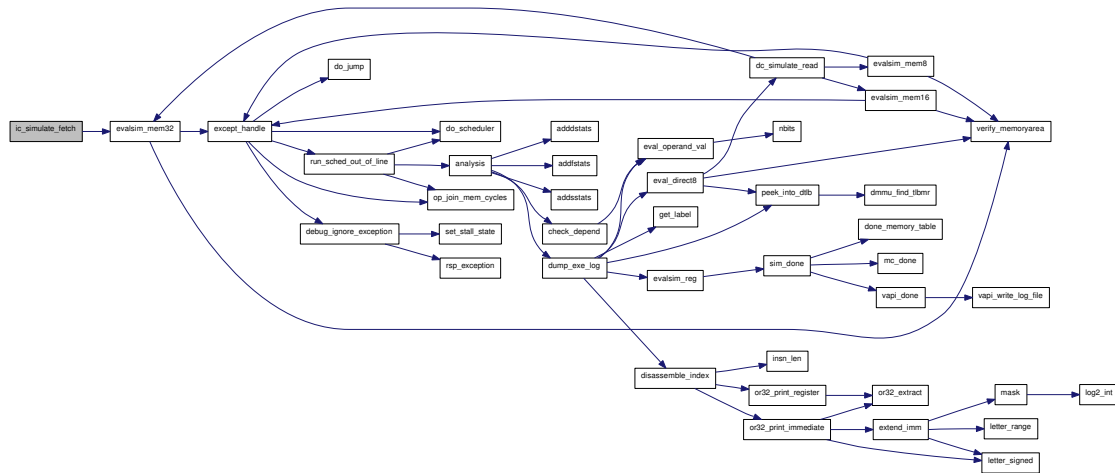
- struct `ic * ic_state`

6.6.1 Function Documentation

6.6.1.1 void ic_inv (oraddr_t dataaddr)

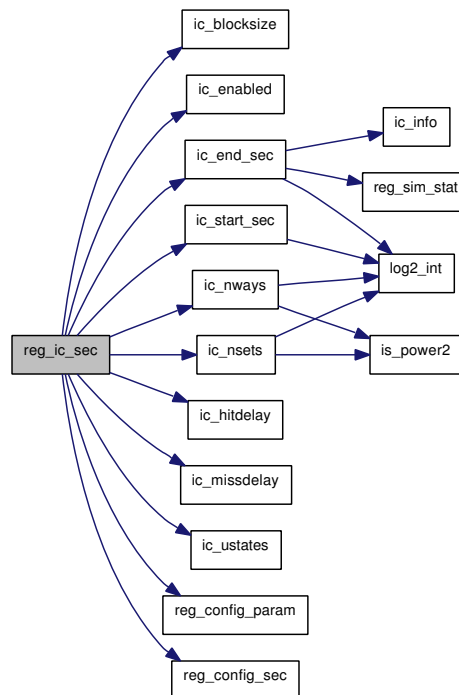
6.6.1.2 uint32_t ic_simulate_fetch (oraddr_t fetchaddr, oraddr_t virt_addr)

Here is the call graph for this function:



6.6.1.3 void reg_ic_sec ()

Here is the call graph for this function:



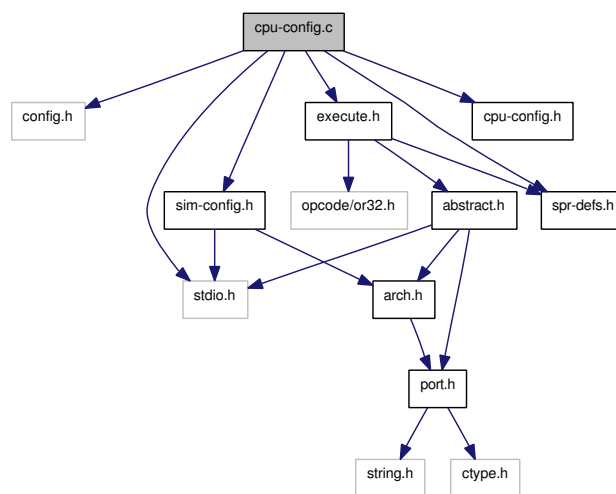
6.6.2 Variable Documentation

6.6.2.1 struct ic* ic_state

6.7 cpu-config.c File Reference

```
#include "config.h"
#include <stdio.h>
#include "cpu-config.h"
#include "sim-config.h"
#include "spr-defs.h"
#include "execute.h"
```

Include dependency graph for `cpu-config.c`:



Defines

- #define `WARNING(s)` `fprintf(stderr, "Warning: config.%s: %s\n", cur_section → name, (s))`

Functions

- static void `cpu_ver` (union `param_val` val, void *dat)
- static void `cpu_cfg` (union `param_val` val, void *dat)
- static void `cpu_rev` (union `param_val` val, void *dat)
- static void `cpu_upr` (union `param_val` val, void *dat)
- static void `cpu_cfgr` (union `param_val` val, void *dat)
- static void `cpu_sr` (union `param_val` val, void *dat)
- static void `cpu_hazards` (union `param_val` val, void *dat)
- static void `cpu_superscalar` (union `param_val` val, void *dat)
- static void `cpu_dependstats` (union `param_val` val, void *dat)
- static void `cpu_sbuf_len` (union `param_val` val, void *dat)
- void `reg_cpu_sec` ()

6.7.1 Define Documentation

6.7.1.1 `#define WARNING(s) fprintf(stderr, "Warning: config.%s: %s\n", cur_section → name, (s))`

6.7.2 Function Documentation

6.7.2.1 `static void cpu_cfg (union param_val val, void * dat) [static]`

Set the CPU configuration

Value must be an 8-bit integer. Larger values are truncated with a warning.

Parameters:

← *val* The value to use

← *dat* The [config](#) data structure (not used here)

6.7.2.2 `static void cpu_cfgr (union param_val val, void * dat) [static]`

Set the CPU configuration

Value must be just the OB32S instruction set bit. Nothing else is currently supported. If other values are specified, they will be set, but with a warning.

Parameters:

← *val* The value to use

← *dat* The [config](#) data structure (not used here)

6.7.2.3 `static void cpu_dependstats (union param_val val, void * dat) [static]`

6.7.2.4 `static void cpu_hazards (union param_val val, void * dat) [static]`

6.7.2.5 `static void cpu_rev (union param_val val, void * dat) [static]`

Set the CPU revision

Value must be an 6-bit integer. Larger values are truncated with a warning.

Parameters:

← *val* The value to use

← *dat* The [config](#) data structure (not used here)

6.7.2.6 `static void cpu_sbuf_len (union param_val val, void * dat) [static]`

6.7.2.7 `static void cpu_sr (union param_val val, void * dat) [static]`

Set the CPU supervision register

Only the lowest 17 bits may be set. The top 4 bits are for context ID's (not currently supported), the rest are reserved and should not be set.

If such values are specified, the value will be set (it has no effect), but with a warning.

Parameters:

← *val* The value to use

← *dat* The [config](#) data structure (not used here)

6.7.2.8 `static void cpu_superscalar (union param_val val, void * dat) [static]`

6.7.2.9 `static void cpu_upr (union param_val val, void * dat) [static]`

6.7.2.10 `static void cpu_ver (union param_val val, void * dat) [static]`

Set the CPU version

Value must be an 8-bit integer. Larger values are truncated with a warning.

Parameters:

← *val* The value to use

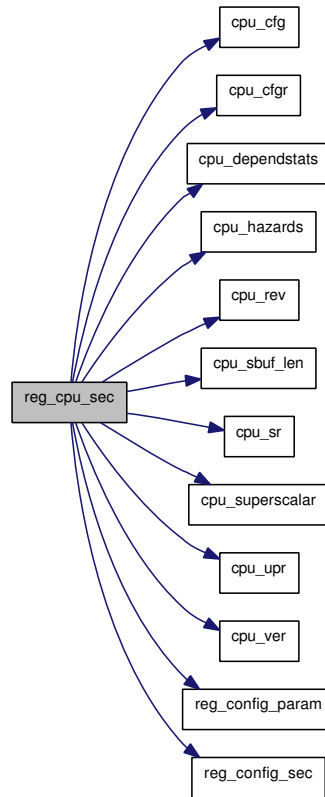
← *dat* The [config](#) data structure (not used here)

6.7.2.11 `void reg_cpu_sec ()`

Register the functions to handle a section cpu

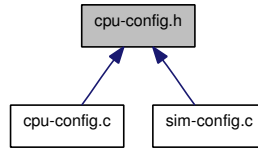
This section does not allocate dynamically a data structure holding its [config](#) information. It's all in the global [config.sim](#) data structure. Therefore it does not need a start and end function to initialize default values (although it might be clearer to do so). The default values are set in [init_defconfig\(\)](#).

Here is the call graph for this function:



6.8 cpu-config.h File Reference

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



Functions

- void [reg_cpu_sec](#) ()

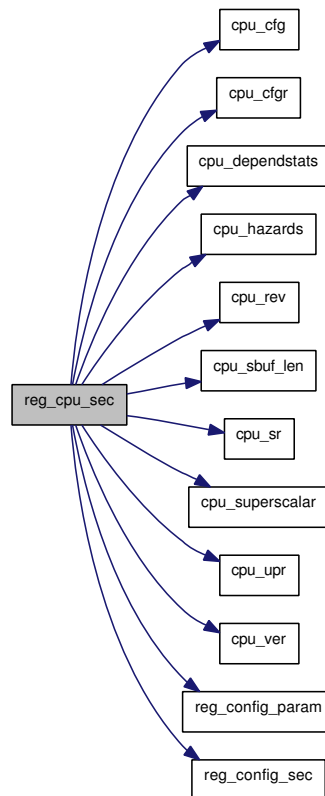
6.8.1 Function Documentation

6.8.1.1 void [reg_cpu_sec](#) ()

Register the functions to handle a section cpu

This section does not allocate dynamically a data structure holding its [config](#) information. It's all in the global [config.sim](#) data structure. Therefore it does not need a start and end function to initialize default values (although it might be clearer to do so). The default values are set in [init_defconfig](#) ().

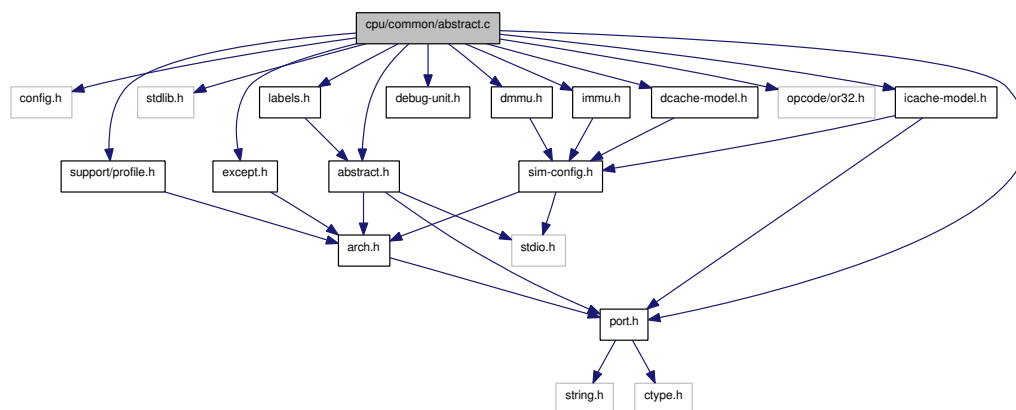
Here is the call graph for this function:



6.9 cpu/common/abstract.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "abstract.h"
#include "except.h"
#include "support/profile.h"
#include "debug-unit.h"
#include "icache-model.h"
#include "dcache-model.h"
#include "labels.h"
#include "opcode/or32.h"
#include "dmmu.h"
#include "immu.h"
```

Include dependency graph for abstract.c:



Functions

- static uint32_t [eval_mem_32_inv](#) (oraddr_t, void *)
- static uint16_t [eval_mem_16_inv](#) (oraddr_t, void *)
- static uint8_t [eval_mem_8_inv](#) (oraddr_t, void *)
- static uint32_t [eval_mem_32_inv_direct](#) (oraddr_t, void *)
- static uint16_t [eval_mem_16_inv_direct](#) (oraddr_t, void *)
- static uint8_t [eval_mem_8_inv_direct](#) (oraddr_t, void *)
- static void [set_mem_32_inv](#) (oraddr_t, uint32_t, void *)
- static void [set_mem_16_inv](#) (oraddr_t, uint16_t, void *)
- static void [set_mem_8_inv](#) (oraddr_t, uint8_t, void *)
- static void [set_mem_32_inv_direct](#) (oraddr_t, uint32_t, void *)
- static void [set_mem_16_inv_direct](#) (oraddr_t, uint16_t, void *)
- static void [set_mem_8_inv_direct](#) (oraddr_t, uint8_t, void *)

- static unsigned int `bit_mask` (uint32_t data)
- static struct `dev_memarea` * `register_memoryarea_mask` (oraddr_t addr_mask, oraddr_t addr_-, compare, uint32_t size, unsigned mc_dev)
- struct `dev_memarea` * `reg_mem_area` (oraddr_t addr, uint32_t size, unsigned mc_dev, struct `mem_ops` *ops)
- struct `dev_memarea` * `verify_memoryarea` (oraddr_t addr)
- void `set_mem_valid` (struct `dev_memarea` *mem, int valid)
- void `adjust_rw_delay` (struct `dev_memarea` *mem, int delayr, int delayw)
- uint32_t `evalsim_mem32` (oraddr_t memaddr, oraddr_t vaddr)
- uint16_t `evalsim_mem16` (oraddr_t memaddr, oraddr_t vaddr)
- uint8_t `evalsim_mem8` (oraddr_t memaddr, oraddr_t vaddr)
- uint32_t `eval_mem32` (oraddr_t memaddr, int *breakpoint)
- uint32_t `eval_direct32` (oraddr_t memaddr, int through_mmu, int through_dc)
- uint32_t `eval_insn` (oraddr_t memaddr, int *breakpoint)
- uint16_t `eval_mem16` (oraddr_t memaddr, int *breakpoint)
- uint16_t `eval_direct16` (oraddr_t memaddr, int through_mmu, int through_dc)
- uint8_t `eval_mem8` (oraddr_t memaddr, int *breakpoint)
- uint8_t `eval_direct8` (oraddr_t memaddr, int through_mmu, int through_dc)
- void `setsim_mem32` (oraddr_t memaddr, oraddr_t vaddr, uint32_t value)
- void `setsim_mem16` (oraddr_t memaddr, oraddr_t vaddr, uint16_t value)
- void `setsim_mem8` (oraddr_t memaddr, oraddr_t vaddr, uint8_t value)
- void `set_mem32` (oraddr_t memaddr, uint32_t value, int *breakpoint)
- void `set_direct32` (oraddr_t memaddr, uint32_t value, int through_mmu, int through_dc)
- void `set_mem16` (oraddr_t memaddr, uint16_t value, int *breakpoint)
- void `set_direct16` (oraddr_t memaddr, uint16_t value, int through_mmu, int through_dc)
- void `set_mem8` (oraddr_t memaddr, uint8_t value, int *breakpoint)
- void `set_direct8` (oraddr_t memaddr, uint8_t value, int through_mmu, int through_dc)
- void `set_program32` (oraddr_t memaddr, uint32_t value)
- void `set_program8` (oraddr_t memaddr, uint8_t value)
- void `dump_memory` (oraddr_t from, oraddr_t to)
- void `disassemble_memory` (oraddr_t from, oraddr_t to, int nl)
- void `done_memory_table` ()
- void `memory_table_status` (void)
- char * `generate_time_pretty` (char *dest, long time_ps)

Variables

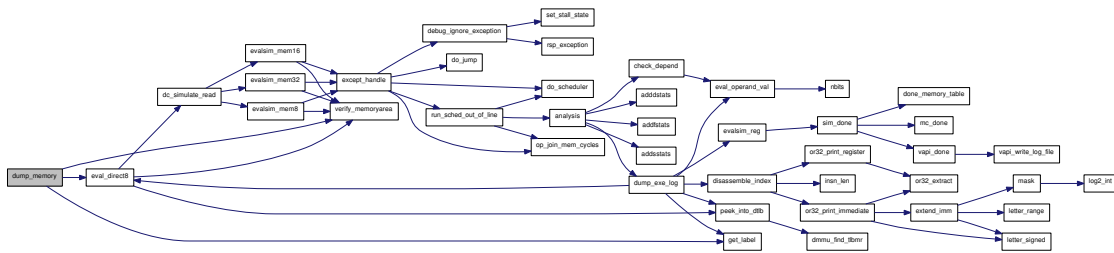
- struct `dev_memarea` * `cur_area`
- int `data_ci`
- int `insn_ci`
- static struct `dev_memarea` * `dev_list`
- static struct `dev_memarea` * `mc_area` = NULL
- static oraddr_t `cur_vadd`

Parameters:

← *from* Start address of the area of memory

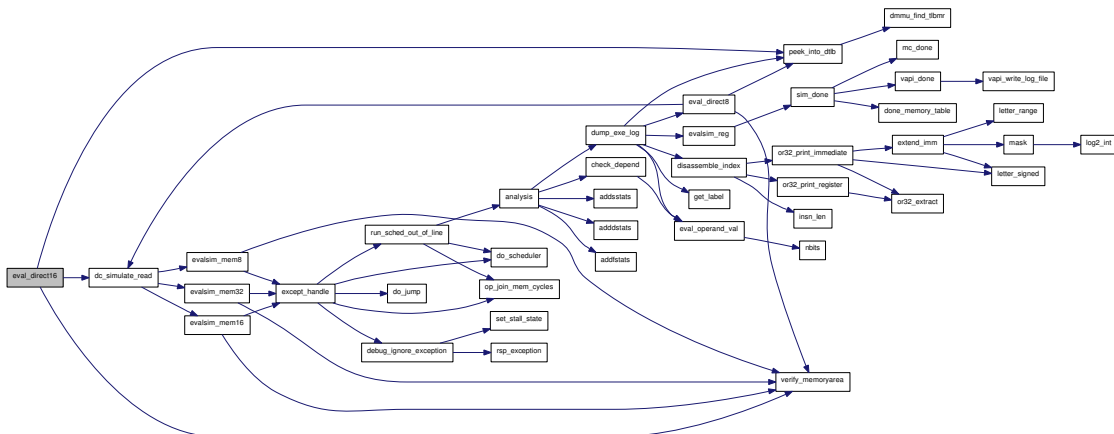
← *to* End address of the area of memory

Here is the call graph for this function:



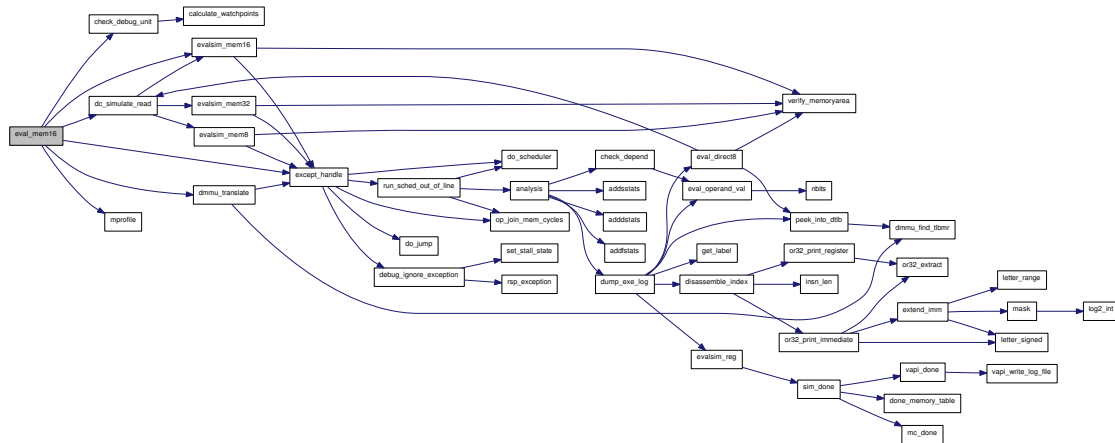
6.9.1.6 uint16_t eval_direct16 (oraddr_t memaddr, int through_mmu, int through_dc)

Here is the call graph for this function:



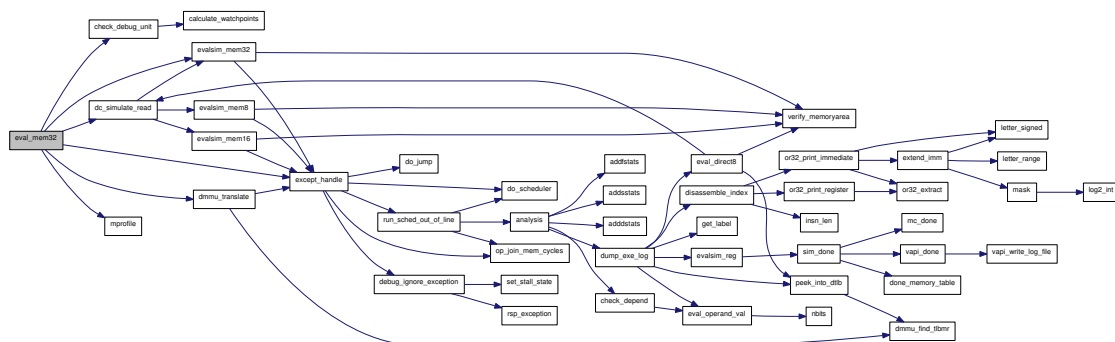
6.9.1.10 uint16_t eval_mem16 (oraddr_t memaddr, int * breakpoint)

Here is the call graph for this function:



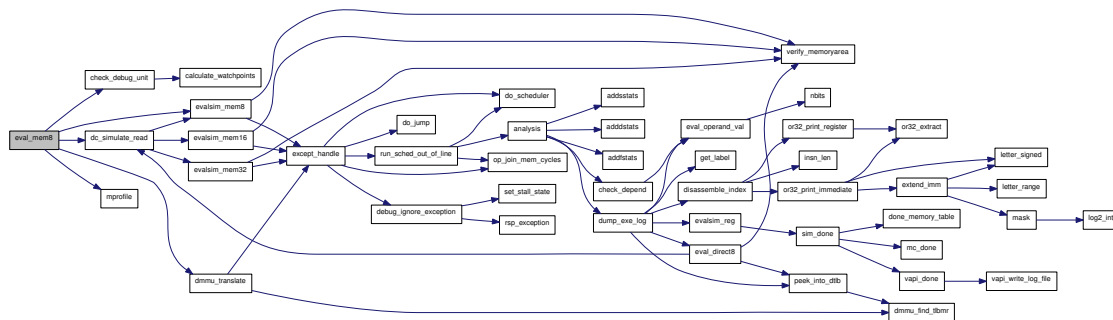
6.9.1.11 uint32_t eval_mem32 (oraddr_t memaddr, int * breakpoint)

Here is the call graph for this function:



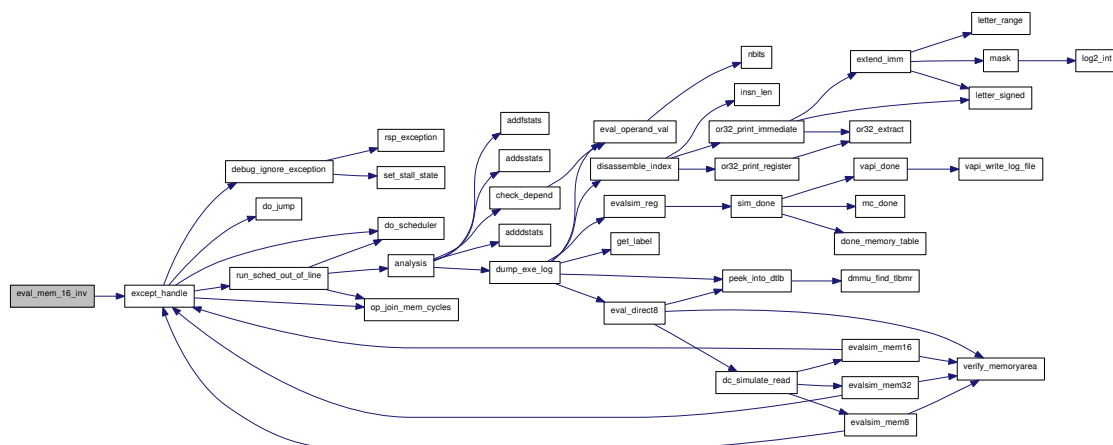
6.9.1.12 uint8_t eval_mem8 (oraddr_t memaddr, int * breakpoint)

Here is the call graph for this function:



6.9.1.13 static uint16_t eval_mem_16_inv (oraddr_t memaddr, void * dat) [static]

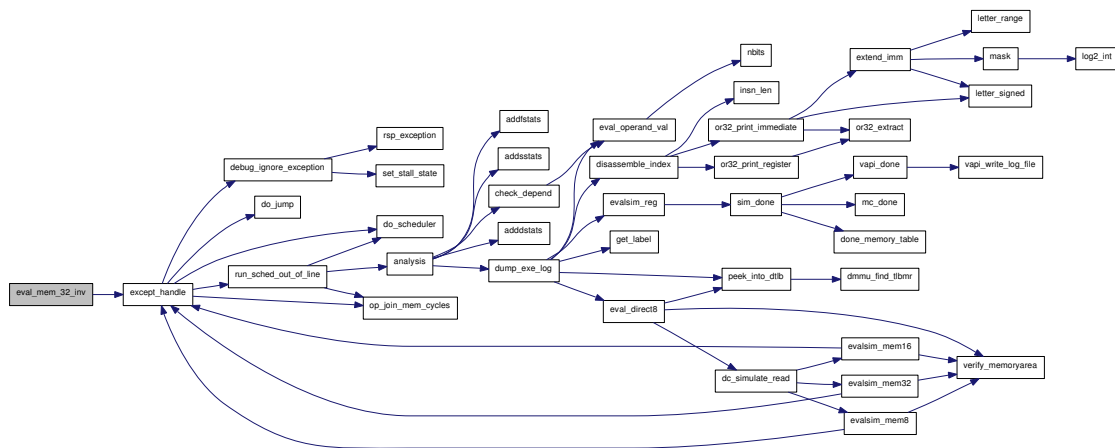
Here is the call graph for this function:



6.9.1.14 `uint16_t eval_mem_16_inv_direct (oraddr_t memaddr, void * dat)` [static]

6.9.1.15 `static uint32_t eval_mem_32_inv (oraddr_t memaddr, void * dat)` [static]

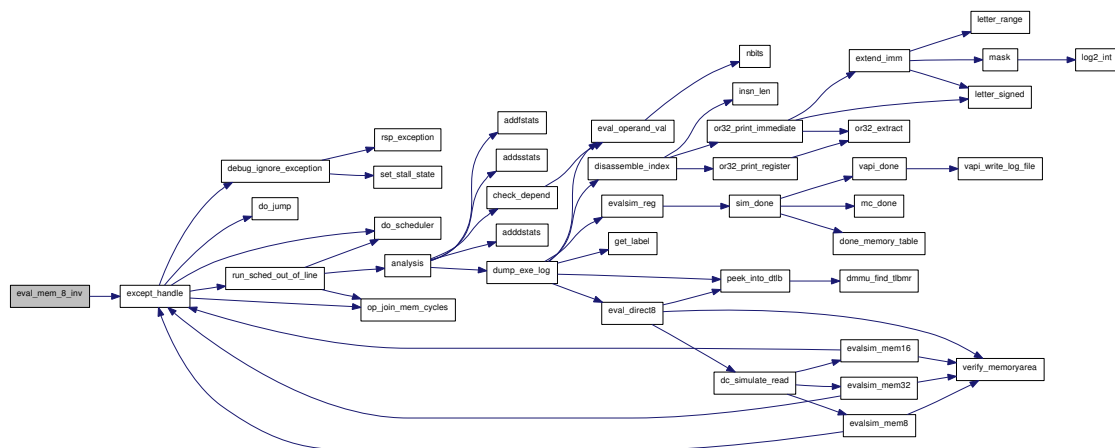
Here is the call graph for this function:



6.9.1.16 `uint32_t eval_mem_32_inv_direct (oraddr_t memaddr, void * dat)` [static]

6.9.1.17 `static uint8_t eval_mem_8_inv (oraddr_t memaddr, void * dat)` [static]

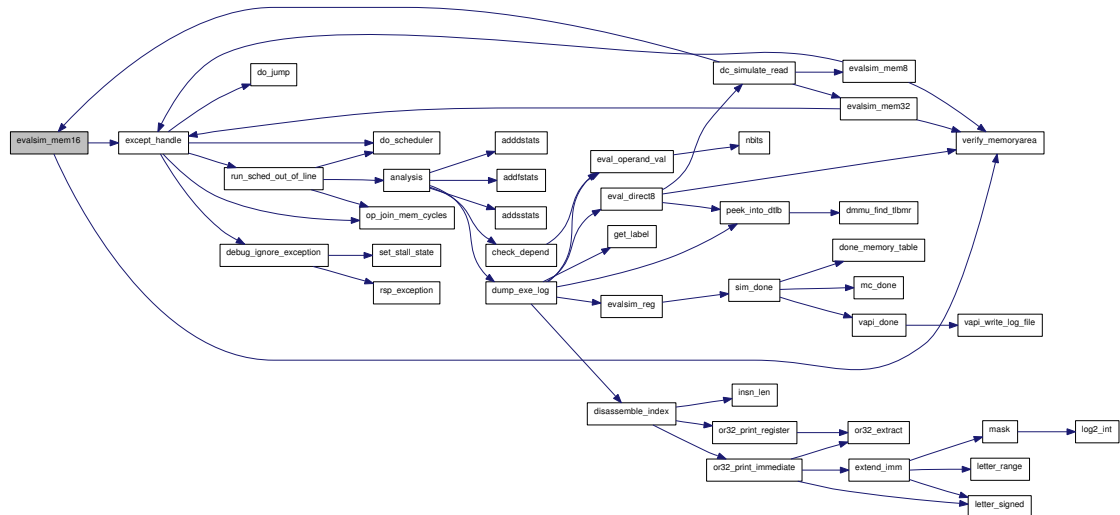
Here is the call graph for this function:



6.9.1.18 `uint8_t eval_mem_8_inv_direct (oraddr_t memaddr, void * dat)` [static]

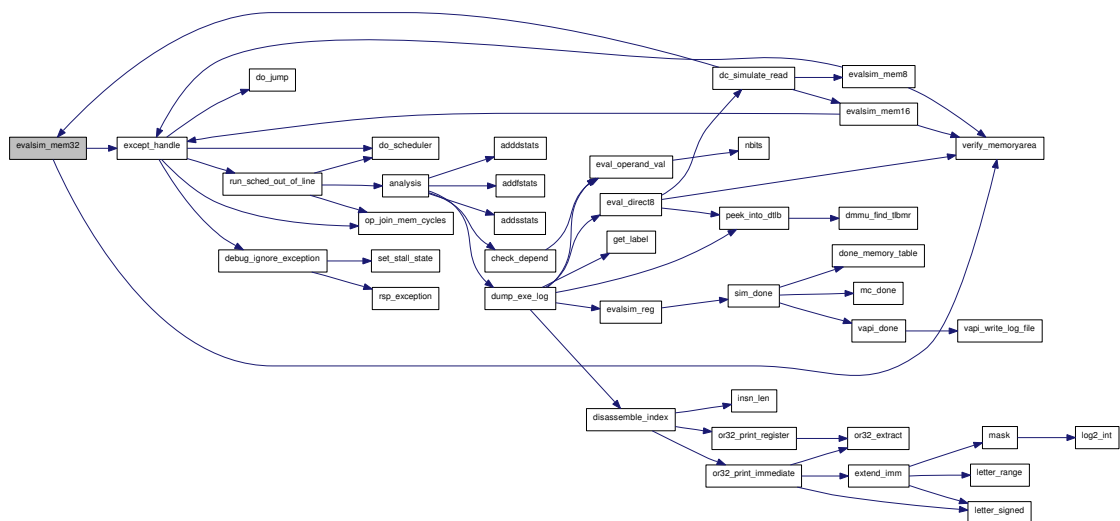
6.9.1.19 `uint16_t evalsim_mem16 (oraddr_t memaddr, oraddr_t vaddr)`

Here is the call graph for this function:



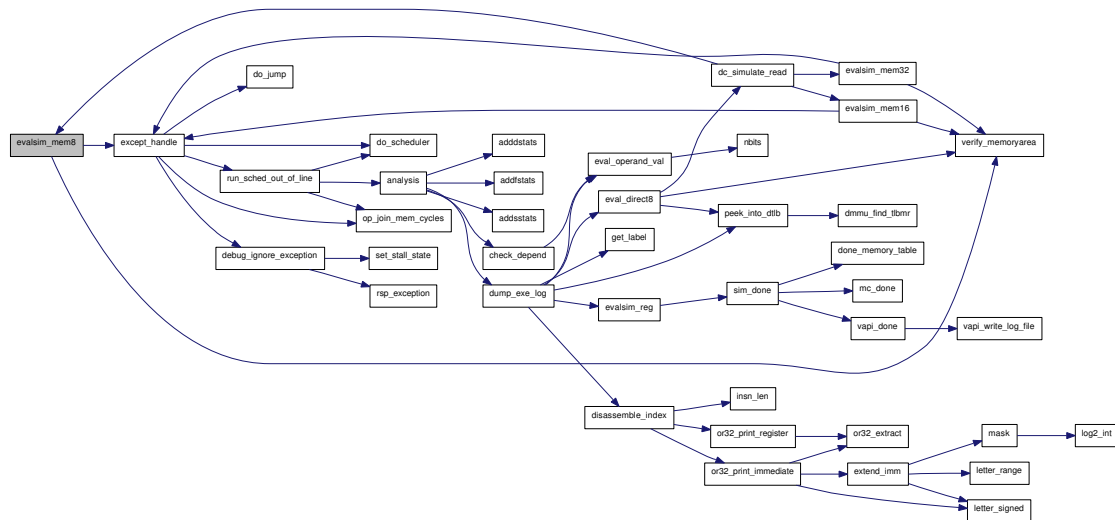
6.9.1.20 `uint32_t evalsim_mem32 (oraddr_t memaddr, oraddr_t vaddr)`

Here is the call graph for this function:



6.9.1.21 uint8_t evalsim_mem8 (oraddr_t memaddr, oraddr_t vaddr)

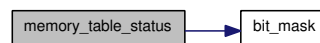
Here is the call graph for this function:



6.9.1.22 char* generate_time_pretty (char * dest, long time_ps)

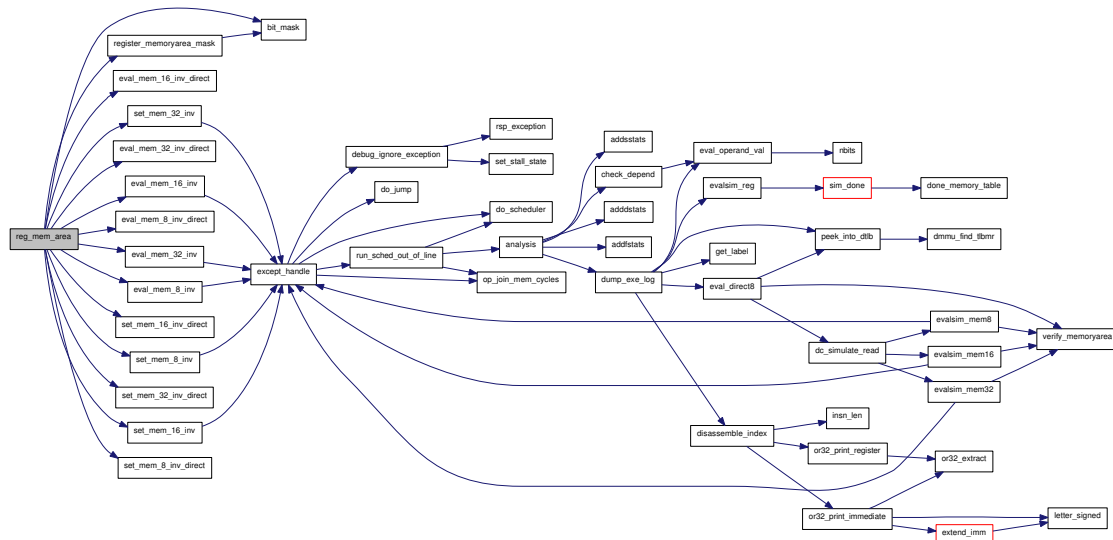
6.9.1.23 void memory_table_status (void)

Here is the call graph for this function:



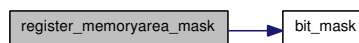
6.9.1.24 struct dev_memarea* reg_mem_area (oraddr_t addr, uint32_t size, unsigned mc_dev, struct mem_ops * ops) [read]

Here is the call graph for this function:



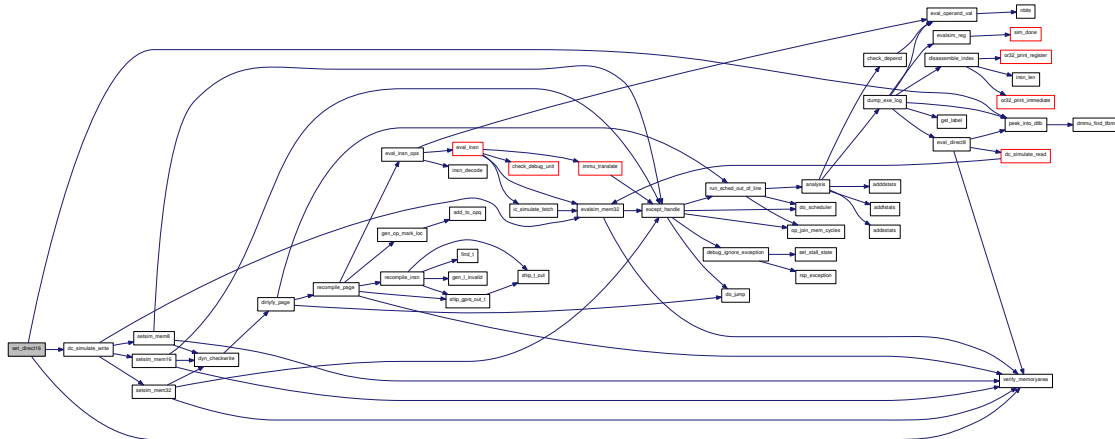
6.9.1.25 static struct dev_memarea* register_memoryarea_mask (oraddr_t addr_mask, oraddr_t addr_compare, uint32_t size, unsigned mc_dev) [static, read]

Here is the call graph for this function:



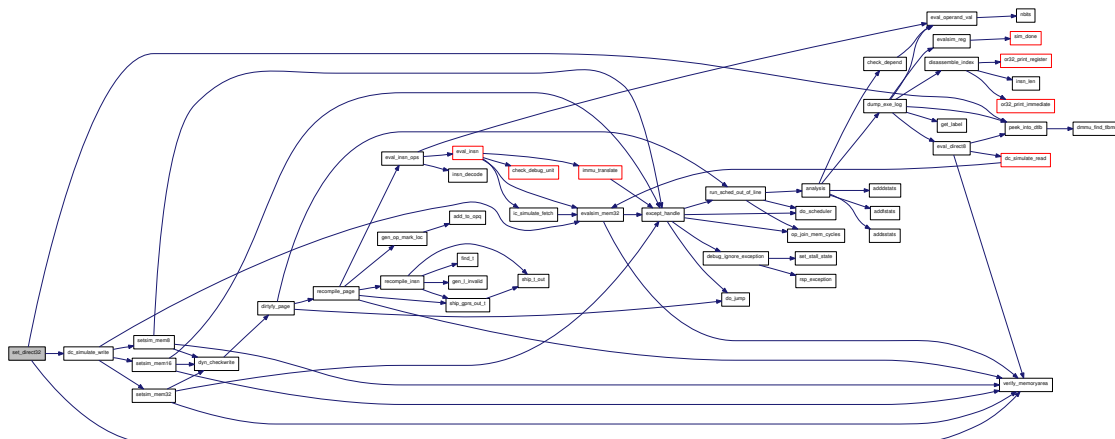
6.9.1.26 void set_direct16 (oraddr_t memaddr, uint16_t value, int through_mmu, int through_dc)

Here is the call graph for this function:



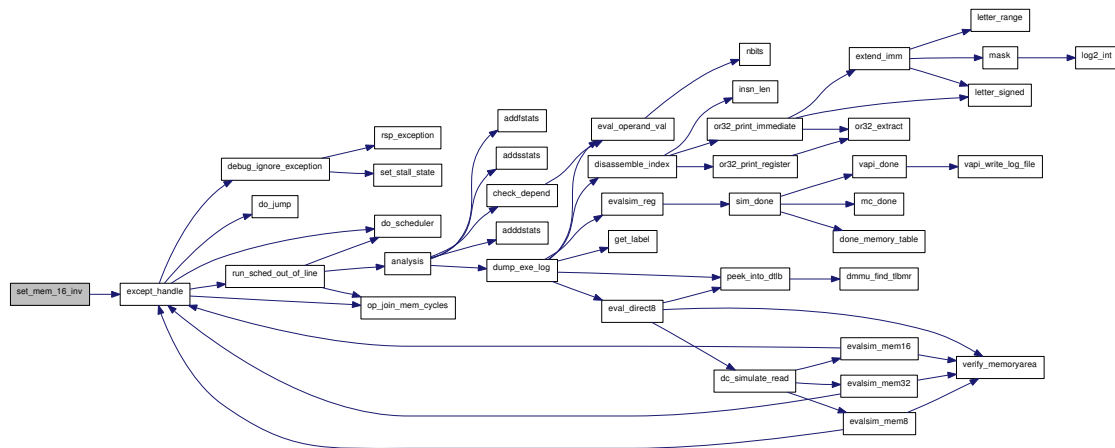
6.9.1.27 void set_direct32 (oraddr_t memaddr, uint32_t value, int through_mmu, int through_dc)

Here is the call graph for this function:



6.9.1.32 static void set_mem_16_inv (oraddr_t memaddr, uint16_t val, void * dat) [static]

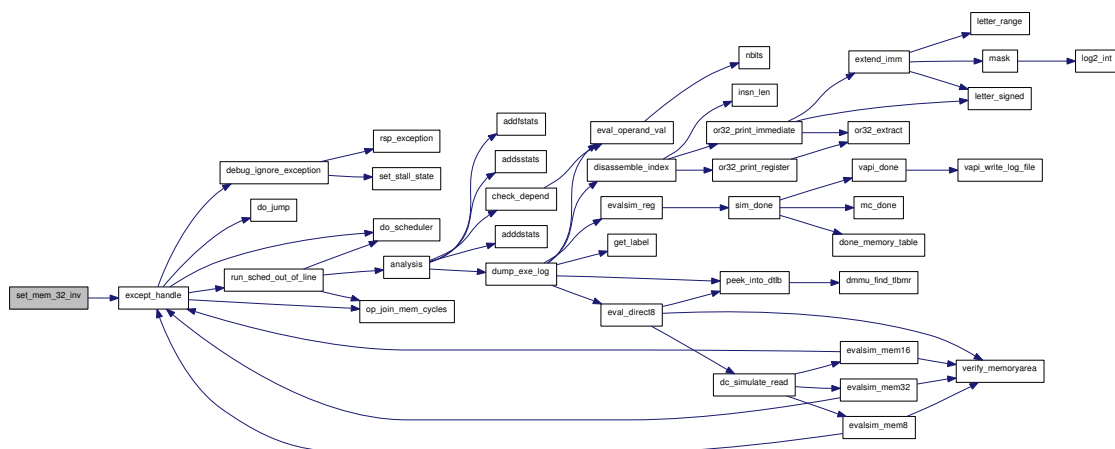
Here is the call graph for this function:



6.9.1.33 void set_mem_16_inv_direct (oraddr_t memaddr, uint16_t val, void * dat) [static]

6.9.1.34 static void set_mem_32_inv (oraddr_t memaddr, uint32_t val, void * dat) [static]

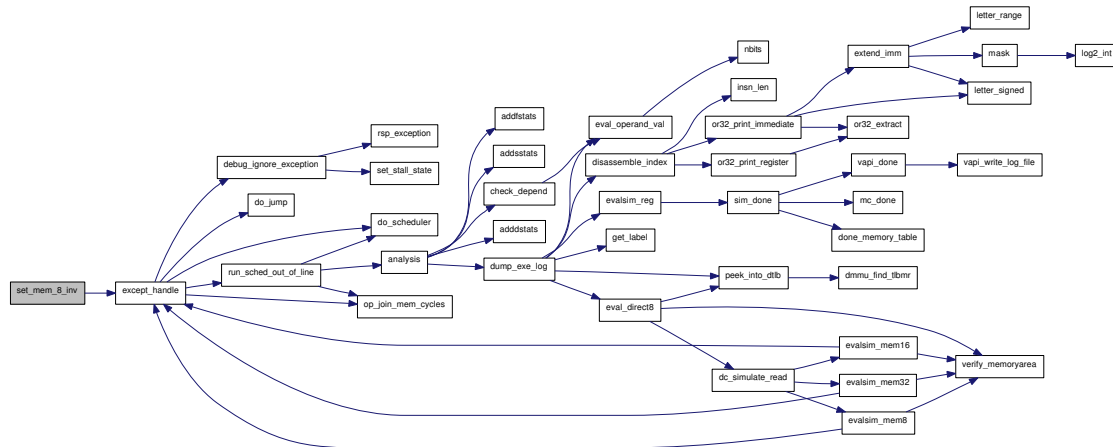
Here is the call graph for this function:



6.9.1.35 void set_mem_32_inv_direct (oraddr_t memaddr, uint32_t val, void * dat) [static]

6.9.1.36 static void set_mem_8_inv (oraddr_t memaddr, uint8_t val, void * dat) [static]

Here is the call graph for this function:



6.9.1.37 void set_mem_8_inv_direct (oraddr_t memaddr, uint8_t val, void * dat) [static]

6.9.1.38 void set_mem_valid (struct dev_memarea * mem, int valid)

6.9.1.39 void set_program32 (oraddr_t memaddr, uint32_t value)

Here is the call graph for this function:



6.9.1.40 void set_program8 (oraddr_t memaddr, uint8_t value)

Here is the call graph for this function:



6.9.1.44 `struct dev_memarea* verify_memoryarea (oraddr_t addr)` [read]

6.9.2 Variable Documentation

6.9.2.1 `struct dev_memarea* cur_area`

Global temporary variable to increase speed.

6.9.2.2 `oraddr_t cur_vadd` [static]

6.9.2.3 `int data_ci`

Global var: data cache inhibit bit set

6.9.2.4 `struct dev_memarea* dev_list` [static]

6.9.2.5 `int insn_ci`

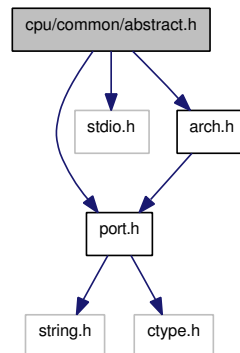
Global var: instr cache inhibit bit set

6.9.2.6 `struct dev_memarea* mc_area = NULL` [static]

6.10 cpu/common/abstract.h File Reference

```
#include "port.h"
#include <stdio.h>
#include "arch.h"
```

Include dependency graph for abstract.h:



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



Data Structures

- struct [iqueue_entry](#)
- struct [mem_ops](#)
- struct [dev_memarea](#)

Defines

- #define [DEFAULT_MEMORY_START](#) 0
- #define [DEFAULT_MEMORY_LEN](#) 0x800000
- #define [STACK_SIZE](#) 20
- #define [LABELNAME_LEN](#) 50
- #define [INSNAME_LEN](#) 15
- #define [OPERANDNAME_LEN](#) 50
- #define [MAX_OPERANDS](#) 5
- #define [OP_MEM_ACCESS](#) 0x80000000
- #define [CT_NONE](#) 0
- #define [CT_VIRTUAL](#) 1
- #define [CT_PHYSICAL](#) 2
- #define [HISTEXEC_LEN](#) 200
- #define [LONGEST](#) long long
- #define [ULONGEST](#) unsigned long long
- #define [LE16\(x\)](#) bswap_16(x)

Functions

- `uint32_t eval_mem32 (oraddr_t memaddr, int *)`
- `uint16_t eval_mem16 (oraddr_t memaddr, int *)`
- `uint8_t eval_mem8 (oraddr_t memaddr, int *)`
- `void set_mem32 (oraddr_t memaddr, uint32_t value, int *)`
- `void set_mem16 (oraddr_t memaddr, uint16_t value, int *)`
- `void set_mem8 (oraddr_t memaddr, uint8_t value, int *)`
- `void dump_memory (oraddr_t from, oraddr_t to)`
- `void disassemble_memory (oraddr_t from, oraddr_t to, int nl)`
- `uint32_t evalsim_mem32 (oraddr_t, oraddr_t)`
- `uint16_t evalsim_mem16 (oraddr_t, oraddr_t)`
- `uint8_t evalsim_mem8 (oraddr_t, oraddr_t)`
- `void setsim_mem32 (oraddr_t, oraddr_t, uint32_t)`
- `void setsim_mem16 (oraddr_t, oraddr_t, uint16_t)`
- `void setsim_mem8 (oraddr_t, oraddr_t, uint8_t)`
- `void done_memory_table ()`
- `void memory_table_status (void)`
- `struct dev_memarea * reg_mem_area (oraddr_t addr, uint32_t size, unsigned mc_dev, struct mem_ops *ops)`
- `void adjust_rw_delay (struct dev_memarea *mem, int delayr, int delayw)`
- `void set_mem_valid (struct dev_memarea *mem, int valid)`
- `struct dev_memarea * verify_memoryarea (oraddr_t addr)`
- `char * generate_time_pretty (char *dest, long time_ps)`
- `uint32_t eval_insn (oraddr_t, int *)`
- `uint32_t eval_direct32 (oraddr_t addr, int through_mmu, int through_dc)`
- `uint16_t eval_direct16 (oraddr_t memaddr, int through_mmu, int through_dc)`
- `uint8_t eval_direct8 (oraddr_t memaddr, int through_mmu, int through_dc)`
- `void set_direct8 (oraddr_t, uint8_t, int, int)`
- `void set_direct16 (oraddr_t, uint16_t, int, int)`
- `void set_direct32 (oraddr_t, uint32_t, int, int)`
- `void set_program32 (oraddr_t memaddr, uint32_t value)`
- `void set_program8 (oraddr_t memaddr, uint8_t value)`

Variables

- `struct dev_memarea * cur_area`
- `int data_ci`
- `int insn_ci`
- `struct hist_exec * hist_exec_tail`

6.10.1 Define Documentation

- 6.10.1.1 `#define CT_NONE 0`
- 6.10.1.2 `#define CT_PHYSICAL 2`
- 6.10.1.3 `#define CT_VIRTUAL 1`
- 6.10.1.4 `#define DEFAULT_MEMORY_LEN 0x800000`
- 6.10.1.5 `#define DEFAULT_MEMORY_START 0`
- 6.10.1.6 `#define HISTEXEC_LEN 200`
- 6.10.1.7 `#define INSNAME_LEN 15`
- 6.10.1.8 `#define LABELNAME_LEN 50`
- 6.10.1.9 `#define LE16(x) bswap_16(x)`
- 6.10.1.10 `#define LONGEST long long`
- 6.10.1.11 `#define MAX_OPERANDS 5`
- 6.10.1.12 `#define OP_MEM_ACCESS 0x80000000`
- 6.10.1.13 `#define OPERANDNAME_LEN 50`
- 6.10.1.14 `#define STACK_SIZE 20`
- 6.10.1.15 `#define ULONGEST unsigned long long`

6.10.2 Function Documentation

- 6.10.2.1 `void adjust_rw_delay (struct dev_memarea * mem, int delayr, int delayw)`
- 6.10.2.2 `void disassemble_memory (oraddr_t from, oraddr_t to, int nl)`

Disassemble memory to the current output

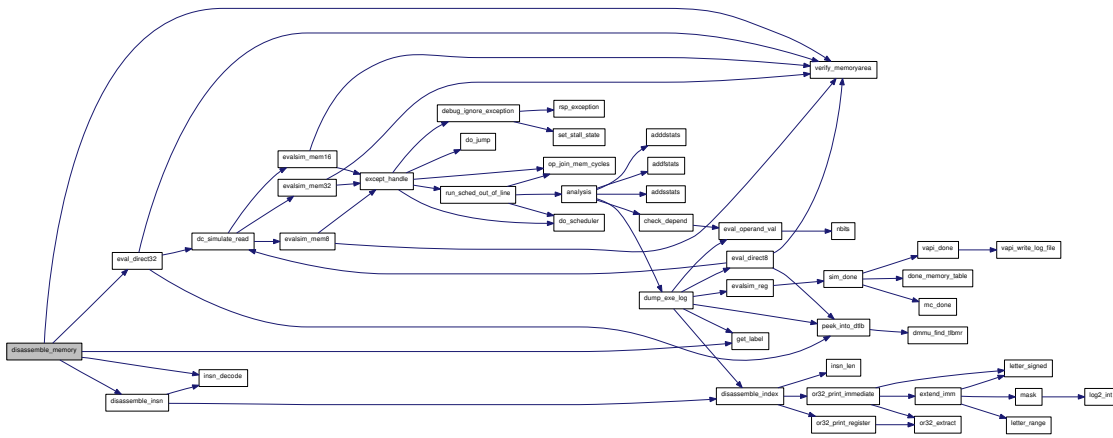
Output format is symbolic disassembly, one instruction per line. Start each line with its address and (optionally) its symbolic name.

There are all sorts of ways to trip this up, but they are unlikely. The validity of a memory area is taken from the address of the start of a line to be printed, so assumes the following 3 bytes are present. This could be fooled by ridiculous memory declarations.

Parameters:

- ← *from* Start address of the area of memory
- ← *to* End address of the area of memory
- ← *nl* If non-zero (true) print a newline at the end of each line

Here is the call graph for this function:



6.10.2.3 void done_memory_table ()

6.10.2.4 void dump_memory (oraddr_t from, oraddr_t to)

Dump memory to the current output

Output format is hex bytes, 16 bytes per line. Start each line with its address and (optionally) its symbolic name. Always end with a newline.

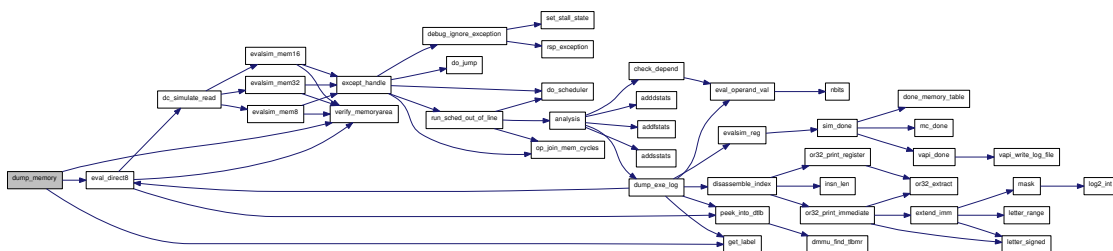
There are all sorts of ways to trip this up, but they are unlikely. The validity of a memory area is taken from the address of the start of a line to be printed, so assumes the following 15 bytes are present. This could be fooled by ridiculous memory declarations.

Parameters:

← *from* Start address of the area of memory

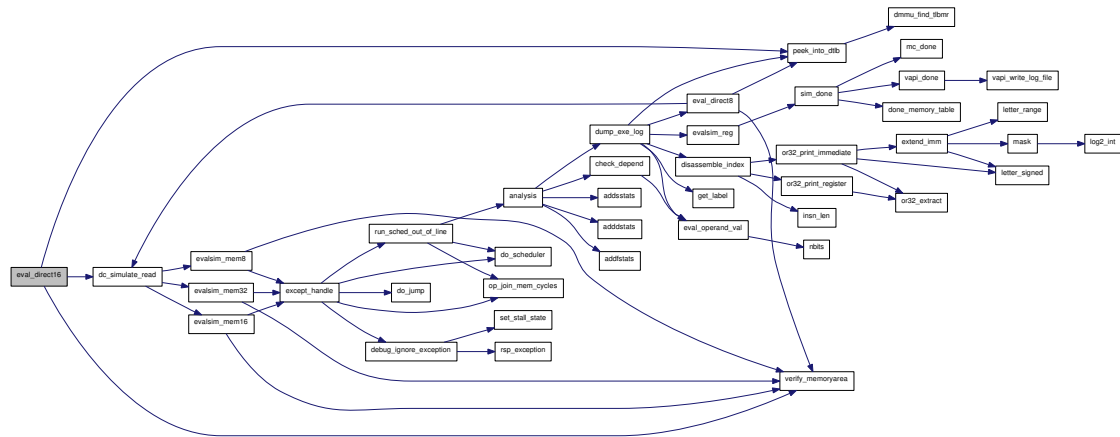
← *to* End address of the area of memory

Here is the call graph for this function:



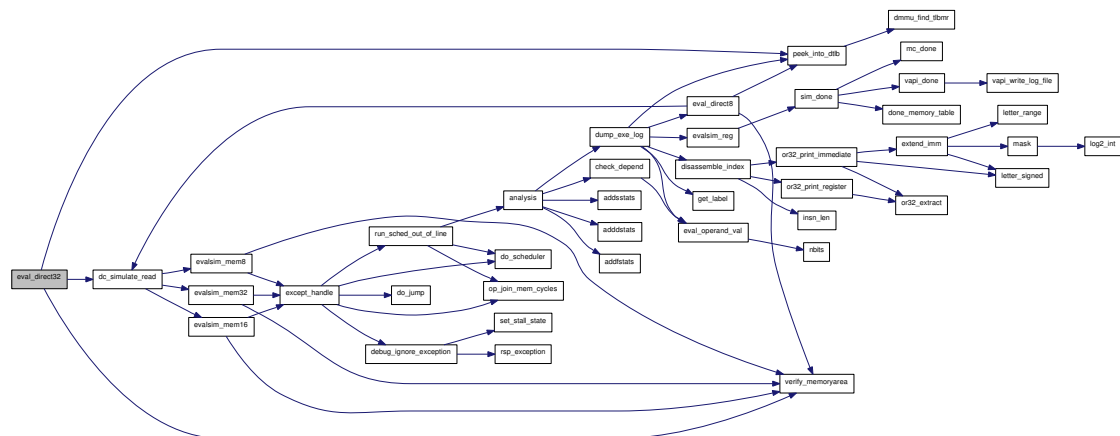
6.10.2.5 uint16_t eval_direct16 (oraddr_t memaddr, int through_mmu, int through_dc)

Here is the call graph for this function:



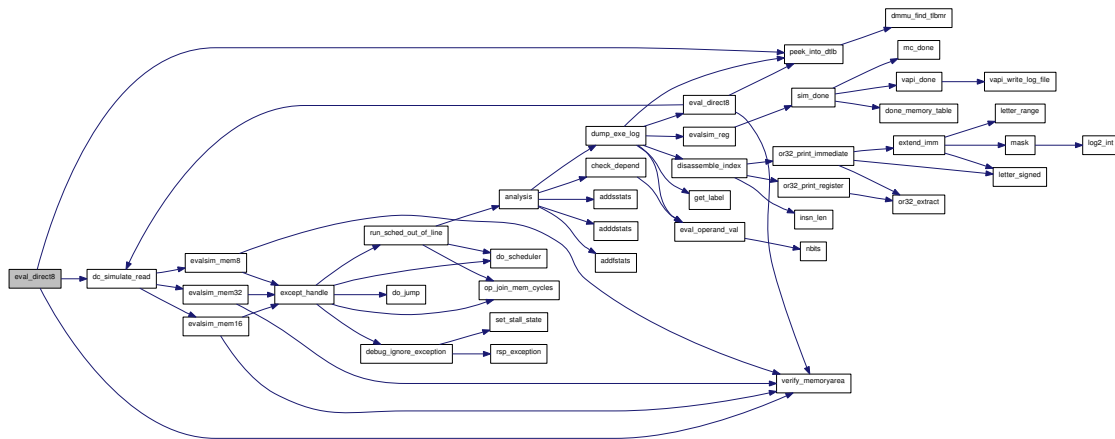
6.10.2.6 uint32_t eval_direct32 (oraddr_t addr, int through_mmu, int through_dc)

Here is the call graph for this function:



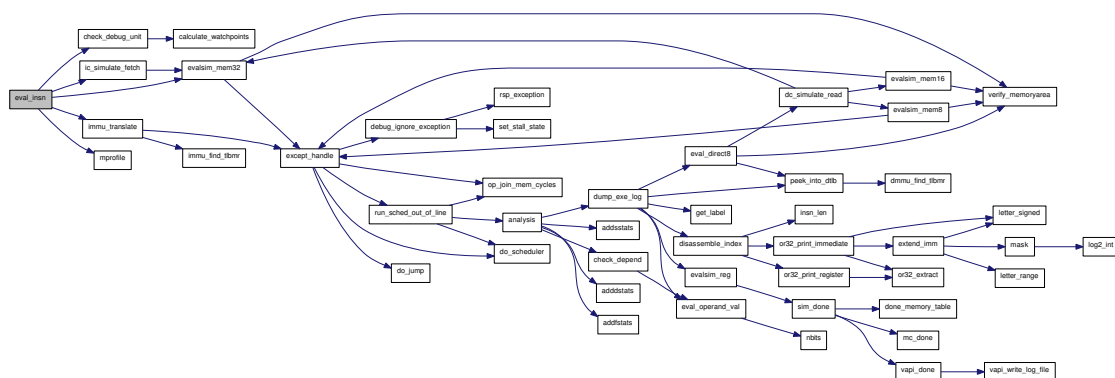
6.10.2.7 uint8_t eval_direct8 (oraddr_t memaddr, int through_mmu, int through_dc)

Here is the call graph for this function:



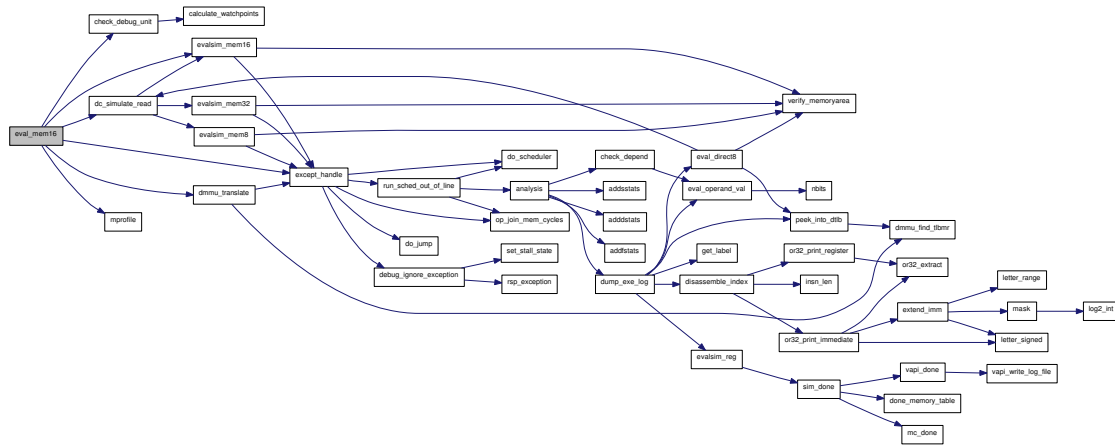
6.10.2.8 uint32_t eval_insn (oraddr_t, int *)

Here is the call graph for this function:



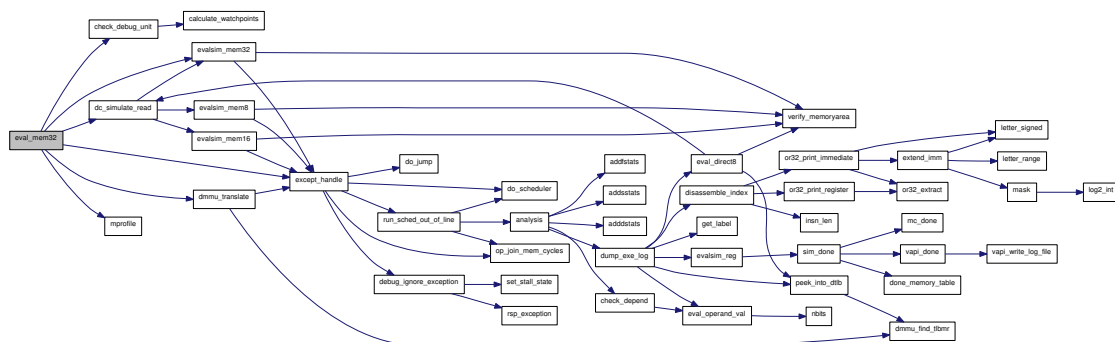
6.10.2.9 uint16_t eval_mem16 (oraddr_t memaddr, int *)

Here is the call graph for this function:



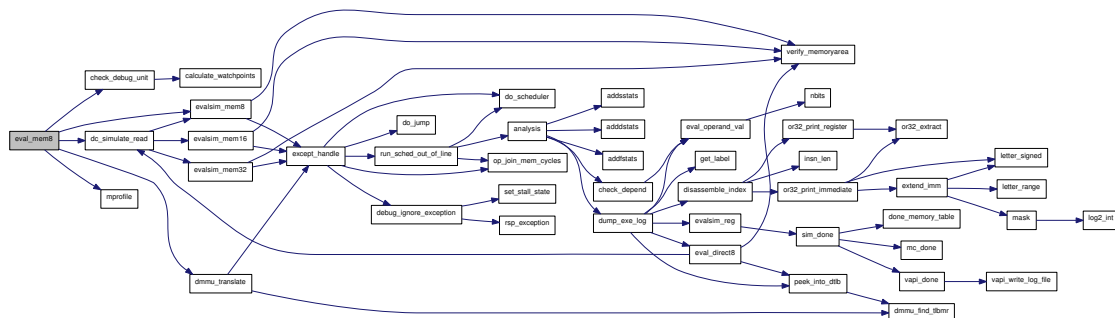
6.10.2.10 uint32_t eval_mem32 (oraddr_t memaddr, int *)

Here is the call graph for this function:



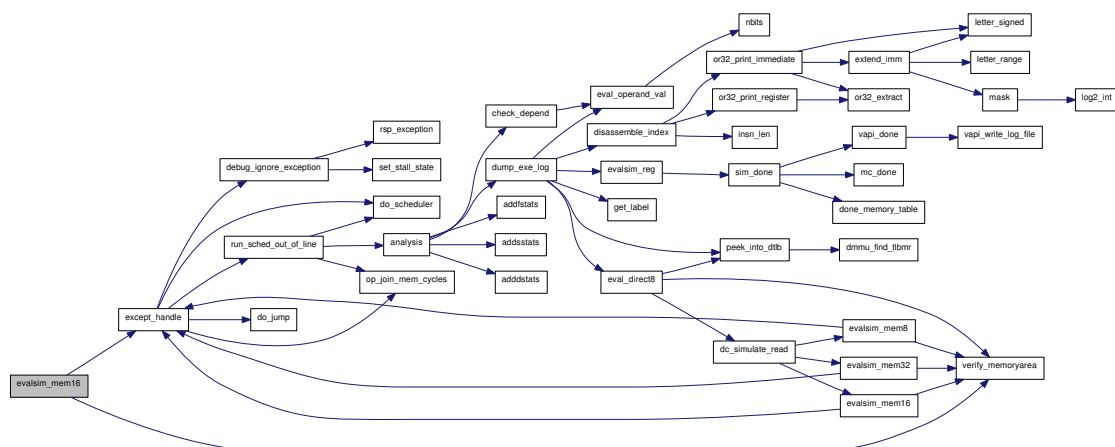
6.10.2.11 uint8_t eval_mem8 (oraddr_t memaddr, int *)

Here is the call graph for this function:



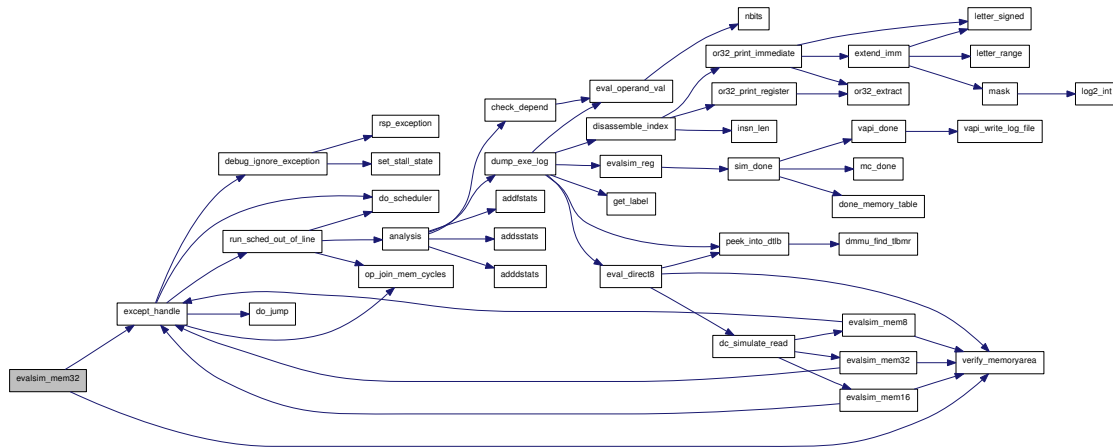
6.10.2.12 uint16_t evalsim_mem16 (oraddr_t, oraddr_t)

Here is the call graph for this function:



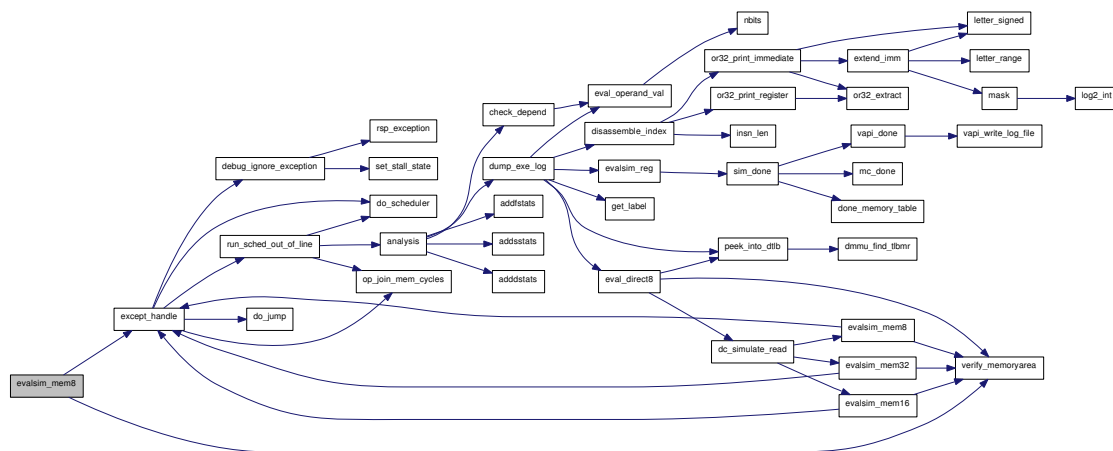
6.10.2.13 uint32_t evalsim_mem32 (oraddr_t, oraddr_t)

Here is the call graph for this function:



6.10.2.14 uint8_t evalsim_mem8 (oraddr_t, oraddr_t)

Here is the call graph for this function:



6.10.2.15 char* generate_time_pretty (char * dest, long time_ps)

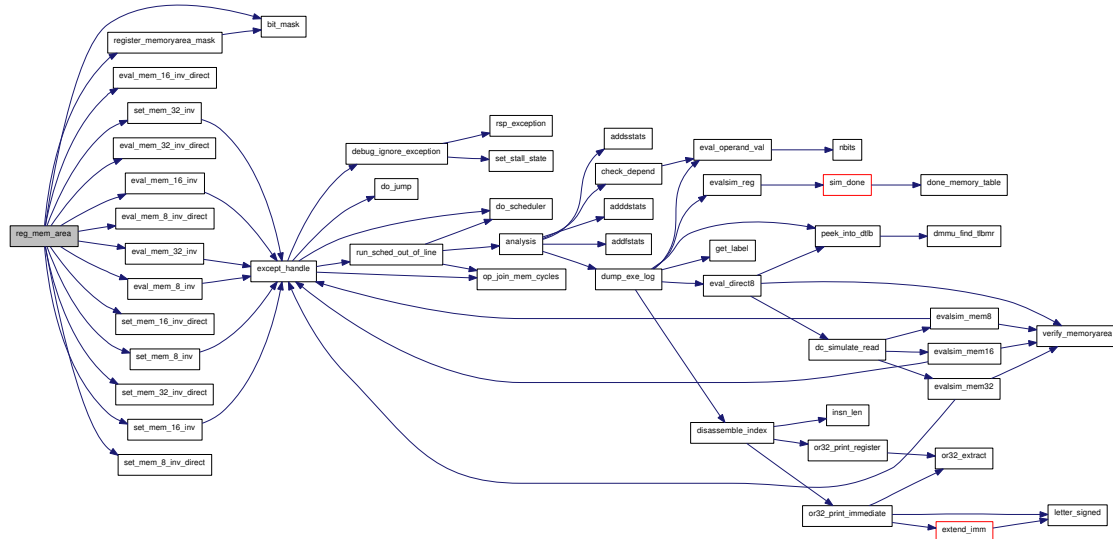
6.10.2.16 void memory_table_status (void)

Here is the call graph for this function:



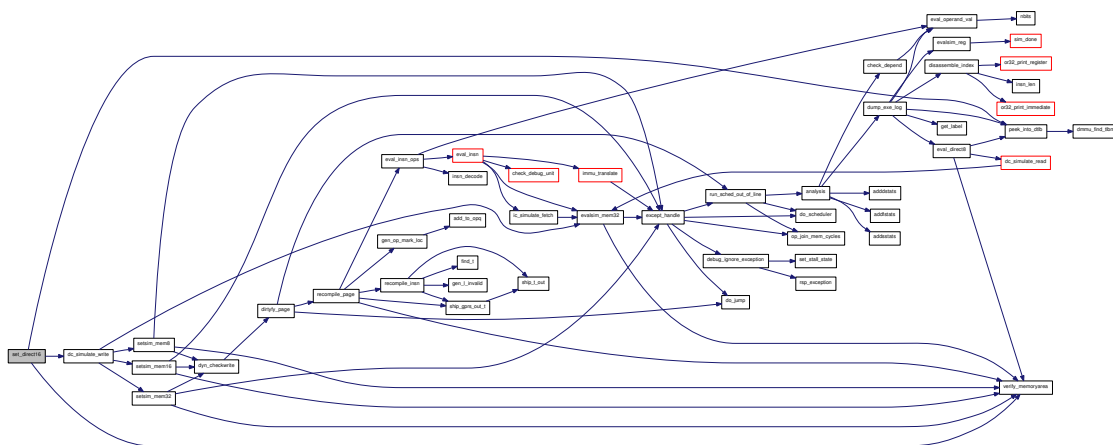
6.10.2.17 `struct dev_memarea* reg_mem_area (oraddr_t addr, uint32_t size, unsigned mc_dev, struct mem_ops * ops) [read]`

Here is the call graph for this function:



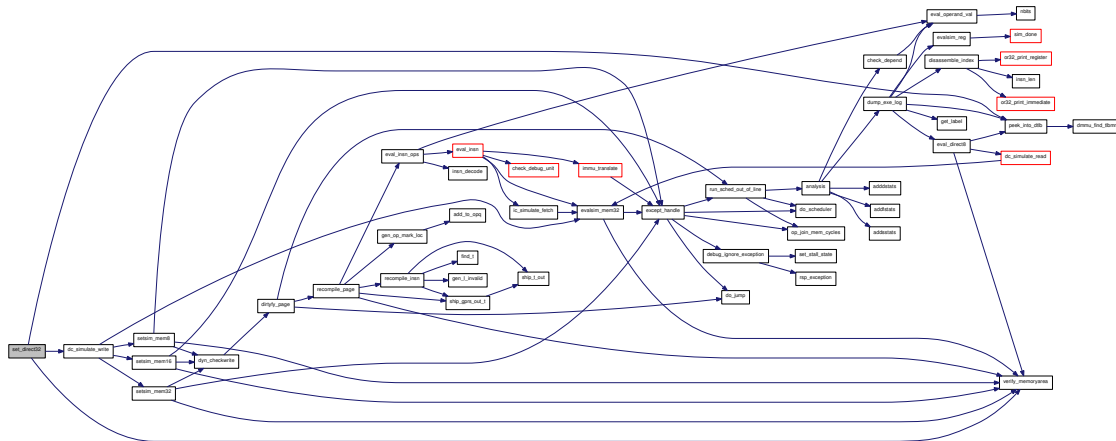
6.10.2.18 `void set_direct16 (oraddr_t, uint16_t, int, int)`

Here is the call graph for this function:



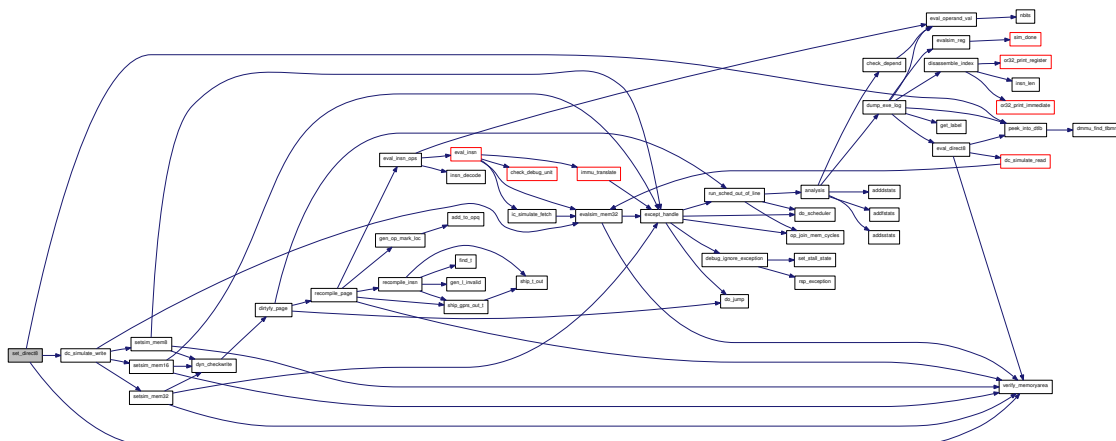
6.10.2.19 void set_direct32 (oraddr_t, uint32_t, int, int)

Here is the call graph for this function:



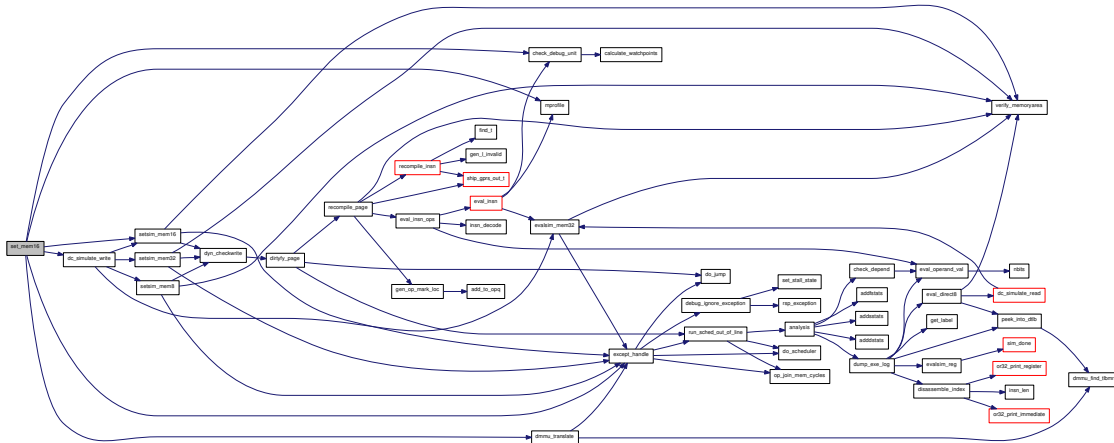
6.10.2.20 void set_direct8 (oraddr_t, uint8_t, int, int)

Here is the call graph for this function:



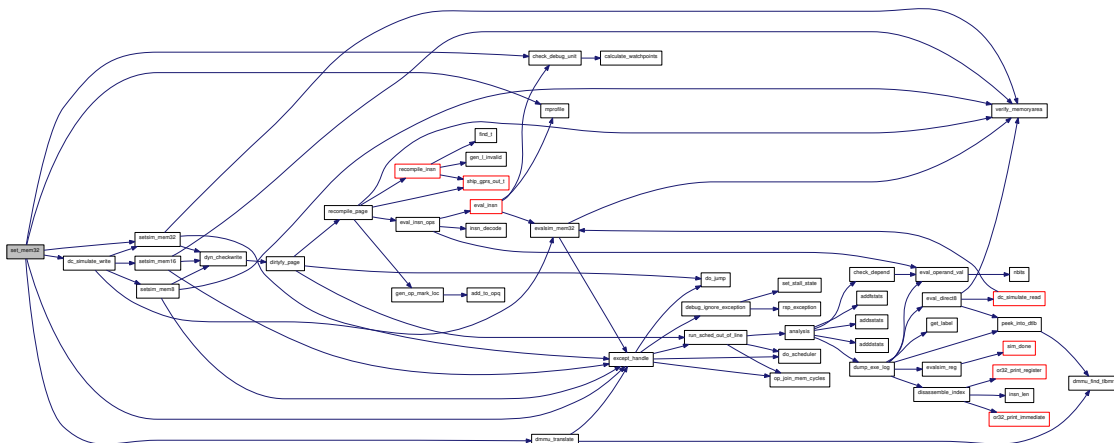
6.10.2.21 void set_mem16 (oraddr_t memaddr, uint16_t value, int *)

Here is the call graph for this function:



6.10.2.22 void set_mem32 (oraddr_t memaddr, uint32_t value, int *)

Here is the call graph for this function:



6.10.2.30 `struct dev_memarea* verify_memoryarea (oraddr_t addr)` [read]

6.10.3 Variable Documentation

6.10.3.1 `struct dev_memarea* cur_area`

Global temporary variable to increase speed.

6.10.3.2 `int data_ci`

Global var: data cache inhibit bit set

6.10.3.3 `struct hist_exec* hist_exec_tail`

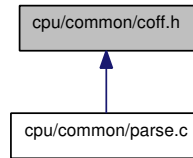
History of execution. Globally available

6.10.3.4 `int insn_ci`

Global var: instr cache inhibit bit set

6.11 cpu/common/coff.h File Reference

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



Data Structures

- struct [COFF_filehdr](#)
- struct [COFF_AOUTHDR](#)
- struct [COFF_scnhdr](#)
- struct [COFF_slib](#)
- struct [COFF_lineno](#)
- struct [COFF_syment](#)
- union [COFF_auxent](#)
- struct [COFF_reloc](#)

Defines

- #define [E_SYMNMLEN](#) 8
- #define [E_FILNMLEN](#) 14
- #define [E_DIMNUM](#) 4
- #define [COFF_SHORT_L](#) SWAP_ENDIAN_SHORT
- #define [COFF_LONG_L](#) SWAP_ENDIAN_LONG
- #define [COFF_SHORT_H](#) KEEP_ENDIAN_SHORT
- #define [COFF_LONG_H](#) KEEP_ENDIAN_LONG
- #define [SWAP_ENDIAN_SHORT](#)(ps)
- #define [SWAP_ENDIAN_LONG](#)(ps)
- #define [KEEP_ENDIAN_SHORT](#)(ps)
- #define [KEEP_ENDIAN_LONG](#)(ps)
- #define [COFF_LONG](#)(v) [COFF_LONG_L](#)(v)
- #define [COFF_SHORT](#)(v) [COFF_SHORT_L](#)(v)
- #define [COFF_F_RELFLG](#) 0000001
- #define [COFF_F_EXEC](#) 0000002
- #define [COFF_F_LNNO](#) 0000004
- #define [COFF_F_LSYMS](#) 0000010
- #define [COFF_F_MINMAL](#) 0000020
- #define [COFF_F_UPDATE](#) 0000040
- #define [COFF_F_SWABD](#) 0000100
- #define [COFF_F_AR16WR](#) 0000200
- #define [COFF_F_AR32WR](#) 0000400
- #define [COFF_F_AR32W](#) 0001000
- #define [COFF_F_PATCH](#) 0002000
- #define [COFF_F_NODF](#) 0002000

- #define [COFF_I386MAGIC](#) 0x14c
- #define [COFF_I386BADMAG\(x\)](#) (COFF_SHORT((x).f_magic) != COFF_I386MAGIC)
- #define [COFF_FILHDR](#) struct [COFF_filehdr](#)
- #define [COFF_FILHSZ](#) sizeof(COFF_FILHDR)
- #define [COFF_AOUTSZ](#) (sizeof(COFF_AOUTHDR))
- #define [COFF_STMAGIC](#) 0401
- #define [COFF_OMAGIC](#) 0404
- #define [COFF_JMAGIC](#) 0407
- #define [COFF_DMAGIC](#) 0410
- #define [COFF_ZMAGIC](#) 0413
- #define [COFF_SHMAGIC](#) 0443
- #define [C_EFCN](#) 0xff
- #define [C_NULL](#) 0
- #define [C_AUTO](#) 1
- #define [C_EXT](#) 2
- #define [C_STAT](#) 3
- #define [C_REG](#) 4
- #define [C_EXTDEF](#) 5
- #define [C_LABEL](#) 6
- #define [C_ULABEL](#) 7
- #define [C_MOS](#) 8
- #define [C_ARG](#) 9
- #define [C_STRTAG](#) 10
- #define [C_MOU](#) 11
- #define [C_UNTAG](#) 12
- #define [C_TPDEF](#) 13
- #define [C_USTATIC](#) 14
- #define [C_ENTAG](#) 15
- #define [C_MOE](#) 16
- #define [C_REGPARAM](#) 17
- #define [C_FIELD](#) 18
- #define [C_AUTOARG](#) 19
- #define [C_LASTENT](#) 20
- #define [C_BLOCK](#) 100
- #define [C_FCN](#) 101
- #define [C_EOS](#) 102
- #define [C_FILE](#) 103
- #define [C_LINE](#) 104
- #define [C_ALIAS](#) 105
- #define [C_HIDDEN](#) 106
- #define [C_WEAKEXT](#) 127
- #define [C_UEXT](#) 19
- #define [C_STATLAB](#) 20
- #define [C_EXTLAB](#) 21
- #define [C_SYSTEM](#) 23
- #define [C_SECTION](#) 104
- #define [C_NT_WEAK](#) 105
- #define [C_LEAFPROC](#) 108
- #define [C_SCALL](#) 107
- #define [C_LEAFEXT](#) 108

- #define C_LEAFSTAT 113
- #define C_OPTVAR 109
- #define C_DEFINE 110
- #define C_PRAGMA 111
- #define C_SEGMENT 112
- #define C_SHADOW 107
- #define C_VERSION 108
- #define C_HIDEXT 107
- #define C_BINCL 108
- #define C_EINCL 109
- #define C_GSYM (0x80)
- #define C_LSYM (0x81)
- #define C_PSYM (0x82)
- #define C_RSYM (0x83)
- #define C_RPSYM (0x84)
- #define C_STSYM (0x85)
- #define C_TCSYM (0x86)
- #define C_BCOMM (0x87)
- #define C_ECOML (0x88)
- #define C_ECOMM (0x89)
- #define C_DECL (0x8c)
- #define C_ENTRY (0x8d)
- #define C_FUN (0x8e)
- #define C_BSTAT (0x8f)
- #define C_ESTAT (0x90)
- #define C_THUMBEXT (128 + C_EXT)
- #define C_THUMBSTAT (128 + C_STAT)
- #define C_THUMBLABEL (128 + C_LABEL)
- #define C_THUMBEXTFUNC (C_THUMBEXT + 20)
- #define C_THUMBSTATFUNC (C_THUMBSTAT + 20)
- #define COFF_SCNHDR struct COFF_scnhdr
- #define COFF_SCNHSZ sizeof(COFF_SCNHDR)
- #define COFF_TEXT ".text"
- #define COFF_DATA ".data"
- #define COFF_BSS ".bss"
- #define COFF_COMMENT ".comment"
- #define COFF_LIB ".lib"
- #define COFF_SECT_TEXT 0
- #define COFF_SECT_DATA 1
- #define COFF_SECT_BSS 2
- #define COFF_SECT_REQD 3
- #define COFF_STYP_REG 0x00
- #define COFF_STYP_DSECT 0x01
- #define COFF_STYP_NOLOAD 0x02
- #define COFF_STYP_GROUP 0x04
- #define COFF_STYP_PAD 0x08
- #define COFF_STYP_COPY 0x10
- #define COFF_STYP_TEXT 0x20
- #define COFF_STYP_DATA 0x40
- #define COFF_STYP_BSS 0x80

- #define `COFF_STYP_INFO` 0x200
- #define `COFF_STYP_OVER` 0x400
- #define `COFF_STYP_LIB` 0x800
- #define `COFF_SLIBHD` struct `COFF_slib`
- #define `COFF_SLIBSZ` sizeof(COFF_SLIBHD)
- #define `COFF_LINENO` struct `COFF_lineno`
- #define `COFF_LINESZ` 6
- #define `COFF_E_SYMNMLEN` 8
- #define `COFF_E_FILNMLEN` 14
- #define `COFF_E_DIMNUM` 4
- #define `COFF_N_BTMASK` (0xf)
- #define `COFF_N_TMASK` (0x30)
- #define `COFF_N_BTSHFT` (4)
- #define `COFF_N_TSHIFT` (2)
- #define `COFF_SYMENT` struct `COFF_syment`
- #define `COFF_SYMESZ` 18
- #define `COFF_AUXENT` union `COFF_auxent`
- #define `COFF_AUXESZ` 18
- #define `COFF_ETEXT` "etext"
- #define `COFF_RELOC` struct `COFF_reloc`
- #define `COFF_RELSZ` 10
- #define `COFF_DEF_DATA_SECTION_ALIGNMENT` 4
- #define `COFF_DEF_BSS_SECTION_ALIGNMENT` 4
- #define `COFF_DEF_TEXT_SECTION_ALIGNMENT` 4
- #define `COFF_DEF_SECTION_ALIGNMENT` 4

6.11.1 Define Documentation

6.11.1.1 `#define C_ALIAS 105`

6.11.1.2 `#define C_ARG 9`

6.11.1.3 `#define C_AUTO 1`

6.11.1.4 `#define C_AUTOARG 19`

6.11.1.5 `#define C_BCOMM (0x87)`

6.11.1.6 `#define C_BINCL 108`

6.11.1.7 `#define C_BLOCK 100`

6.11.1.8 `#define C_BSTAT (0x8f)`

6.11.1.9 `#define C_DECL (0x8c)`

6.11.1.10 `#define C_DEFINE 110`

6.11.1.11 `#define C_ECOML (0x88)`

6.11.1.12 `#define C_ECOMM (0x89)`

6.11.1.13 `#define C_EFCN 0xff`

6.11.1.14 `#define C_EINCL 109`

6.11.1.15 `#define C_ENTAG 15`

6.11.1.16 `#define C_ENTRY (0x8d)`

6.11.1.17 `#define C_EOS 102`

6.11.1.18 `#define C_ESTAT (0x90)`

6.11.1.19 `#define C_EXT 2`

6.11.1.20 `#define C_EXTDEF 5`

6.11.1.21 `#define C_EXTLAB 21`

6.11.1.22 `#define C_FCN 101`

6.11.1.23 `#define C_FIELD 18`

6.11.1.24 `#define C_FILE 103`

6.11.1.25 `#define C_FUN (0x8e)`

6.11.1.26 `#define C_GSYM (0x80)`

6.11.1.27 `#define C_HIDDEN 106`

6.11.1.28 `#define C_HIDEXT 107`

6.11.1.29 `#define C_LABEL 6`

6.11.1.30 `#define C_LASTENT 20`

```
((long) (((unsigned long) ((unsigned char)ps[0])<<24) |\
          ((unsigned long) ((unsigned char)ps[1])<<16) |\
          ((unsigned long) ((unsigned char)ps[2])<<8)  |\
          ((unsigned long) ((unsigned char)ps[3])))))
```

6.11.1.148 #define KEEP_ENDIAN_SHORT(ps)

Value:

```
((short) (((unsigned short) ((unsigned char)ps[0])<<8) |\
          ((unsigned short) ((unsigned char)ps[1]))))
```

6.11.1.149 #define SWAP_ENDIAN_LONG(ps)

Value:

```
((long) (((unsigned long) ((unsigned char)ps[3])<<24) |\
          ((unsigned long) ((unsigned char)ps[2])<<16) |\
          ((unsigned long) ((unsigned char)ps[1])<<8)  |\
          ((unsigned long) ((unsigned char)ps[0]))))
```

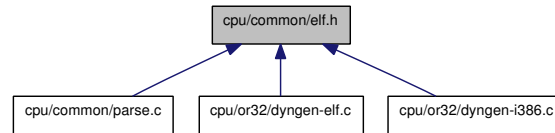
6.11.1.150 #define SWAP_ENDIAN_SHORT(ps)

Value:

```
((short) (((unsigned short) ((unsigned char)ps[1])<<8) |\
          ((unsigned short) ((unsigned char)ps[0]))))
```

6.12 cpu/common/elf.h File Reference

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



Data Structures

- struct [dynamic](#)
- struct [Elf64_Dyn](#)
- struct [elf32_rel](#)
- struct [elf64_rel](#)
- struct [elf32_rela](#)
- struct [elf64_rela](#)
- struct [elf32_sym](#)
- struct [elf64_sym](#)
- struct [elf32_hdr](#)
- struct [elf64_hdr](#)
- struct [elf32_phdr](#)
- struct [elf64_phdr](#)
- struct [elf32_shdr](#)
- struct [elf64_shdr](#)
- struct [elf32_note](#)
- struct [elf64_note](#)

Defines

- #define [ELF_SHORT_H](#)(ps)
- #define [ELF_LONG_H](#)(ps)
- #define [PT_NULL](#) 0
- #define [PT_LOAD](#) 1
- #define [PT_DYNAMIC](#) 2
- #define [PT_INTERP](#) 3
- #define [PT_NOTE](#) 4
- #define [PT_SHLIB](#) 5
- #define [PT_PHDR](#) 6
- #define [PT_LOPROC](#) 0x70000000
- #define [PT_HIPROC](#) 0x7fffffff
- #define [ET_NONE](#) 0
- #define [ET_REL](#) 1
- #define [ET_EXEC](#) 2
- #define [ET_DYN](#) 3
- #define [ET_CORE](#) 4
- #define [ET_LOPROC](#) 5
- #define [ET_HIPROC](#) 6

- #define EM_NONE 0
- #define EM_M32 1
- #define EM_SPARC 2
- #define EM_386 3
- #define EM_68K 4
- #define EM_88K 5
- #define EM_486 6
- #define EM_860 7
- #define EM_MIPS 8
- #define EM_MIPS_RS4_BE 10
- #define EM_SPARC64 11
- #define EM_PARISC 15
- #define EM_SPARC32PLUS 18
- #define EM_PPC 20
- #define EM_ALPHA 0x9026
- #define DT_NULL 0
- #define DT_NEEDED 1
- #define DT_PLTRELSZ 2
- #define DT_PLTGOT 3
- #define DT_HASH 4
- #define DT_STRTAB 5
- #define DT_SYMTAB 6
- #define DT_RELA 7
- #define DT_RELASZ 8
- #define DT_RELAENT 9
- #define DT_STRSZ 10
- #define DT_SYMENT 11
- #define DT_INIT 12
- #define DT_FINI 13
- #define DT_SONAME 14
- #define DT_RPATH 15
- #define DT_SYMBOLIC 16
- #define DT_REL 17
- #define DT_RELSZ 18
- #define DT_RELENT 19
- #define DT_PLTREL 20
- #define DT_DEBUG 21
- #define DT_TEXTREL 22
- #define DT_JMPREL 23
- #define DT_LOPROC 0x70000000
- #define DT_HIPROC 0x7fffffff
- #define STB_LOCAL 0
- #define STB_GLOBAL 1
- #define STB_WEAK 2
- #define STT_NOTYPE 0
- #define STT_OBJECT 1
- #define STT_FUNC 2
- #define STT_SECTION 3
- #define STT_FILE 4
- #define ELF32_ST_BIND(x) ((x) >> 4)

- #define [ELF32_ST_TYPE\(x\)](#) (((unsigned int) x) & 0xf)
- #define [AT_NULL](#) 0
- #define [AT_IGNORE](#) 1
- #define [AT_EXECFD](#) 2
- #define [AT_PHDR](#) 3
- #define [AT_PHENT](#) 4
- #define [AT_PHNUM](#) 5
- #define [AT_PAGESZ](#) 6
- #define [AT_BASE](#) 7
- #define [AT_FLAGS](#) 8
- #define [AT_ENTRY](#) 9
- #define [AT_NOTELF](#) 10
- #define [AT_UID](#) 11
- #define [AT_EUID](#) 12
- #define [AT_GID](#) 13
- #define [AT_EGID](#) 14
- #define [ELF32_R_SYM\(x\)](#) ((x) >> 8)
- #define [ELF32_R_TYPE\(x\)](#) ((x) & 0xff)
- #define [R_386_NONE](#) 0
- #define [R_386_32](#) 1
- #define [R_386_PC32](#) 2
- #define [R_386_GOT32](#) 3
- #define [R_386_PLT32](#) 4
- #define [R_386_COPY](#) 5
- #define [R_386_GLOB_DAT](#) 6
- #define [R_386_JMP_SLOT](#) 7
- #define [R_386_RELATIVE](#) 8
- #define [R_386_GOTOFF](#) 9
- #define [R_386_GOTPC](#) 10
- #define [R_386_NUM](#) 11
- #define [R_68K_NONE](#) 0
- #define [R_68K_32](#) 1
- #define [R_68K_16](#) 2
- #define [R_68K_8](#) 3
- #define [R_68K_PC32](#) 4
- #define [R_68K_PC16](#) 5
- #define [R_68K_PC8](#) 6
- #define [R_68K_GOT32](#) 7
- #define [R_68K_GOT16](#) 8
- #define [R_68K_GOT8](#) 9
- #define [R_68K_GOT32O](#) 10
- #define [R_68K_GOT16O](#) 11
- #define [R_68K_GOT8O](#) 12
- #define [R_68K_PLT32](#) 13
- #define [R_68K_PLT16](#) 14
- #define [R_68K_PLT8](#) 15
- #define [R_68K_PLT32O](#) 16
- #define [R_68K_PLT16O](#) 17
- #define [R_68K_PLT8O](#) 18
- #define [R_68K_COPY](#) 19

- #define R_68K_GLOB_DAT 20
- #define R_68K_JMP_SLOT 21
- #define R_68K_RELATIVE 22
- #define EI_NIDENT 16
- #define PF_R 0x4
- #define PF_W 0x2
- #define PF_X 0x1
- #define SHT_NULL 0
- #define SHT_PROGBITS 1
- #define SHT_SYMTAB 2
- #define SHT_STRTAB 3
- #define SHT_RELA 4
- #define SHT_HASH 5
- #define SHT_DYNAMIC 6
- #define SHT_NOTE 7
- #define SHT_NOBITS 8
- #define SHT_REL 9
- #define SHT_SHLIB 10
- #define SHT_DYNSYM 11
- #define SHT_NUM 12
- #define SHT_LOPROC 0x70000000
- #define SHT_HIPROC 0x7fffffff
- #define SHT_LOUSER 0x80000000
- #define SHT_HIUSER 0xffffffff
- #define SHF_WRITE 0x1
- #define SHF_ALLOC 0x2
- #define SHF_EXECINSTR 0x4
- #define SHF_MASKPROC 0xf0000000
- #define SHN_UNDEF 0
- #define SHN_LORESERVE 0xff00
- #define SHN_LOPROC 0xff00
- #define SHN_HIPROC 0xff1f
- #define SHN_ABS 0xff1
- #define SHN_COMMON 0xff2
- #define SHN_HIRESERVE 0xffff
- #define EI_MAG0 0
- #define EI_MAG1 1
- #define EI_MAG2 2
- #define EI_MAG3 3
- #define EI_CLASS 4
- #define EI_DATA 5
- #define EI_VERSION 6
- #define EI_PAD 7
- #define ELFMAG0 0x7f
- #define ELFMAG1 'E'
- #define ELFMAG2 'L'
- #define ELFMAG3 'F'
- #define ELFMAG "\177ELF"
- #define SELFMAG 4
- #define ELFCLASSNONE 0

- #define [ELFCLASS32](#) 1
- #define [ELFCLASS64](#) 2
- #define [ELFCLASSNUM](#) 3
- #define [ELFDATANONE](#) 0
- #define [ELFDATA2LSB](#) 1
- #define [ELFDATA2MSB](#) 2
- #define [EV_NONE](#) 0
- #define [EV_CURRENT](#) 1
- #define [EV_NUM](#) 2
- #define [NT_PRSTATUS](#) 1
- #define [NT_PRFPREG](#) 2
- #define [NT_PRPSINFO](#) 3
- #define [NT_TASKSTRUCT](#) 4
- #define [elfhdr elf64_hdr](#)
- #define [elf_phdr elf64_phdr](#)
- #define [elf_note elf64_note](#)

Typedefs

- typedef unsigned long [Elf32_Addr](#)
- typedef unsigned short [Elf32_Half](#)
- typedef unsigned long [Elf32_Off](#)
- typedef long [Elf32_Sword](#)
- typedef unsigned long [Elf32_Word](#)
- typedef struct [dynamic](#) [Elf32_Dyn](#)
- typedef struct [elf32_rel](#) [Elf32_Rel](#)
- typedef struct [elf64_rel](#) [Elf64_Rel](#)
- typedef struct [elf32_rela](#) [Elf32_Rela](#)
- typedef struct [elf64_rela](#) [Elf64_Rela](#)
- typedef struct [elf32_sym](#) [Elf32_Sym](#)
- typedef struct [elf64_sym](#) [Elf64_Sym](#)
- typedef struct [elf32_hdr](#) [Elf32_Ehdr](#)
- typedef struct [elf64_hdr](#) [Elf64_Ehdr](#)
- typedef struct [elf32_phdr](#) [Elf32_Phdr](#)
- typedef struct [elf64_phdr](#) [Elf64_Phdr](#)
- typedef struct [elf32_shdr](#) [Elf32_Shdr](#)
- typedef struct [elf64_shdr](#) [Elf64_Shdr](#)
- typedef struct [elf32_note](#) [Elf32_Nhdr](#)
- typedef struct [elf64_note](#) [Elf64_Nhdr](#)

Variables

- [Elf64_Dyn _DYNAMIC](#) []

6.12.1 Define Documentation

- 6.12.1.1 `#define AT_BASE 7`
- 6.12.1.2 `#define AT_EGID 14`
- 6.12.1.3 `#define AT_ENTRY 9`
- 6.12.1.4 `#define AT_EUID 12`
- 6.12.1.5 `#define AT_EXECFD 2`
- 6.12.1.6 `#define AT_FLAGS 8`
- 6.12.1.7 `#define AT_GID 13`
- 6.12.1.8 `#define AT_IGNORE 1`
- 6.12.1.9 `#define AT_NOTELF 10`
- 6.12.1.10 `#define AT_NULL 0`
- 6.12.1.11 `#define AT_PAGESZ 6`
- 6.12.1.12 `#define AT_PHDR 3`
- 6.12.1.13 `#define AT_PHEENT 4`
- 6.12.1.14 `#define AT_PHNUM 5`
- 6.12.1.15 `#define AT_UID 11`
- 6.12.1.16 `#define DT_DEBUG 21`
- 6.12.1.17 `#define DT_FINI 13`
- 6.12.1.18 `#define DT_HASH 4`
- 6.12.1.19 `#define DT_HIPROC 0x7fffffff`
- 6.12.1.20 `#define DT_INIT 12`
- 6.12.1.21 `#define DT_JMPREL 23`
- 6.12.1.22 `#define DT_LOPROC 0x70000000`
- 6.12.1.23 `#define DT_NEEDED 1`
- 6.12.1.24 `#define DT_NULL 0`
- 6.12.1.25 `#define DT_PLTGOT 3`
- 6.12.1.26 `#define DT_PLTREL 20`
- 6.12.1.27 `#define DT_PLTRELSZ 2`
- 6.12.1.28 `#define DT_REL 17`
- 6.12.1.29 `#define DT_RELA 7`
- 6.12.1.30 `#define DT_RELAENT 9`

```
((((unsigned long) (ps) >> 24) & 0xff) |\
  ((unsigned long) (ps) >> 8) & 0xff00) |\
  ((unsigned long) (ps) << 8) & 0xff0000) |\
  ((unsigned long) (ps) << 24) & 0xff000000))
```

6.12.1.56 #define elf_note elf64_note

6.12.1.57 #define elf_phdr elf64_phdr

6.12.1.58 #define ELF_SHORT_H(ps)

Value:

```
((((unsigned short) (ps) >> 8) & 0xff) |\
  ((unsigned short) (ps) << 8) & 0xff00))
```


6.12.1.59 **#define ELFCLASS32 1**

6.12.1.60 **#define ELFCLASS64 2**

6.12.1.61 **#define ELFCLASSNONE 0**

6.12.1.62 **#define ELFCLASSNUM 3**

6.12.1.63 **#define ELFDATA2LSB 1**

6.12.1.64 **#define ELFDATA2MSB 2**

6.12.1.65 **#define ELFDATANONE 0**

6.12.1.66 **#define elfhdr elf64_hdr**

6.12.1.67 **#define ELFMAG "\177ELF"**

6.12.1.68 **#define ELFMAG0 0x7f**

6.12.1.69 **#define ELFMAG1 'E'**

6.12.1.70 **#define ELFMAG2 'L'**

6.12.1.71 **#define ELFMAG3 'F'**

6.12.1.72 **#define EM_386 3**

6.12.1.73 **#define EM_486 6**

6.12.1.74 **#define EM_68K 4**

6.12.1.75 **#define EM_860 7**

6.12.1.76 **#define EM_88K 5**

6.12.1.77 **#define EM_ALPHA 0x9026**

6.12.1.78 **#define EM_M32 1**

6.12.1.79 **#define EM_MIPS 8**

6.12.1.80 **#define EM_MIPS_RS4_BE 10**

6.12.1.81 **#define EM_NONE 0**

6.12.1.82 **#define EM_PARISC 15**

6.12.1.83 **#define EM_PPC 20**

6.12.1.84 **#define EM_SPARC 2**

6.12.1.85 **#define EM_SPARC32PLUS 18**

6.12.1.86 **#define EM_SPARC64 11**

6.12.1.87 **#define ET_CORE 4**

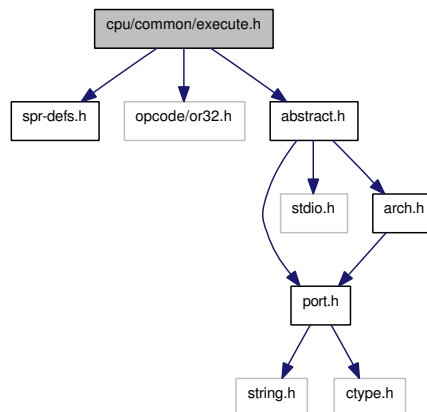
6.12.1.88 **#define ET_DYN 3**

6.12.1.89 **#define ET_EXEC 2**

6.13 `cpu/common/execute.h` File Reference

```
#include "spr-defs.h"
#include "opcode/or32.h"
#include "abstract.h"
```

Include dependency graph for `execute.h`:



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



Data Structures

- struct [cpu_state](#)
- struct [hist_exec](#)

Defines

- #define [CURINSN](#)(INSN) (`strcmp(cur → insn, (INSN)) == 0`)
- #define [SET_OV_FLAG_FN](#)(value)

Functions

- void [dumpreg](#) ()
- void [dump_exe_log](#) ()
- int [cpu_clock](#) ()
- void [cpu_reset](#) ()
- [uorreg_t evalsim_reg](#) (unsigned int regno)
- void [setsim_reg](#) (unsigned int regno, [uorreg_t](#) value)
- [uorreg_t eval_operand_val](#) (uint32_t insn, struct [insn_op_struct](#) *opd)
- void [analysis](#) (struct [iqueue_entry](#) *current)
- void [exec_main](#) ()
- int [depend_operands](#) (struct [iqueue_entry](#) *prev, struct [iqueue_entry](#) *next)

Variables

- struct `cpu_state` `cpu_state`
- oraddr_t `pcnext`
- int `sbuf_wait_cyc`
- int `sbuf_total_cyc`
- int `do_stats`
- struct `hist_exec` * `hist_exec_tail`

6.13.1 Define Documentation

6.13.1.1 `#define CURINSN(INSN) (strcmp(cur → insn, (INSN)) == 0)`

6.13.1.2 `#define SET_OV_FLAG_FN(value)`

Sets a new `SPR_SR_OV` value, based on next register value

6.13.2 Function Documentation

6.13.2.1 `void analysis (struct iqueue_entry * current)`

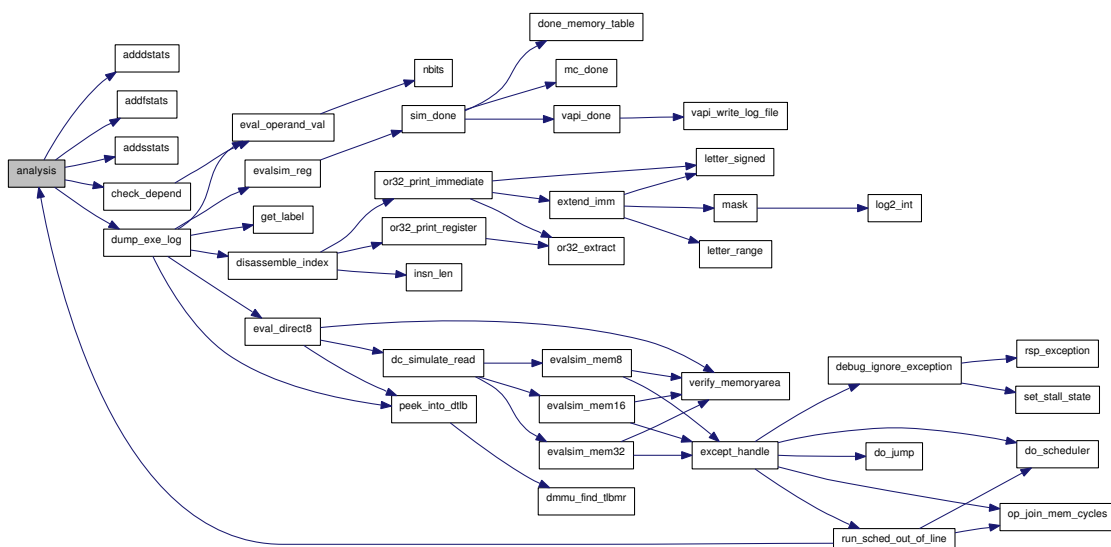
Perform analysis of the instruction being executed

This could be static for `SIMPLE_EXECUTION`, but made global for general use.

Parameters:

← *current* The instruction being executed

Here is the call graph for this function:



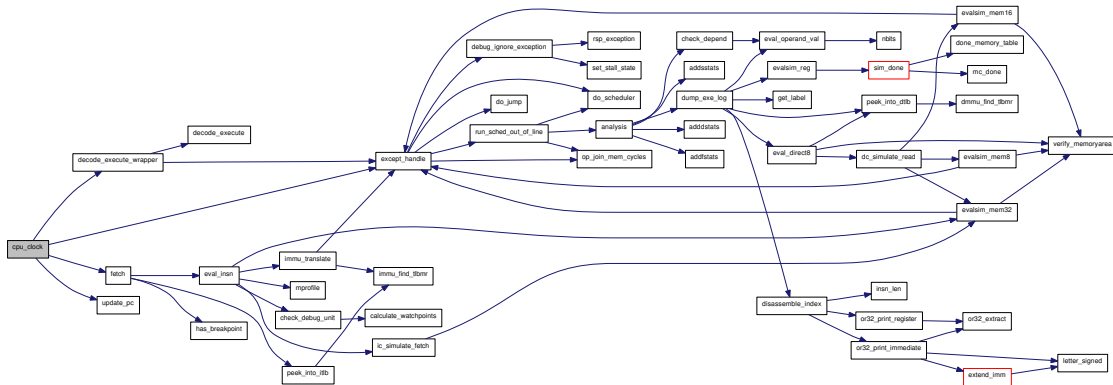
6.13.2.2 int cpu_clock ()

Simulates one CPU clock cycle

Returns:

non-zero if a breakpoint is hit, zero otherwise.

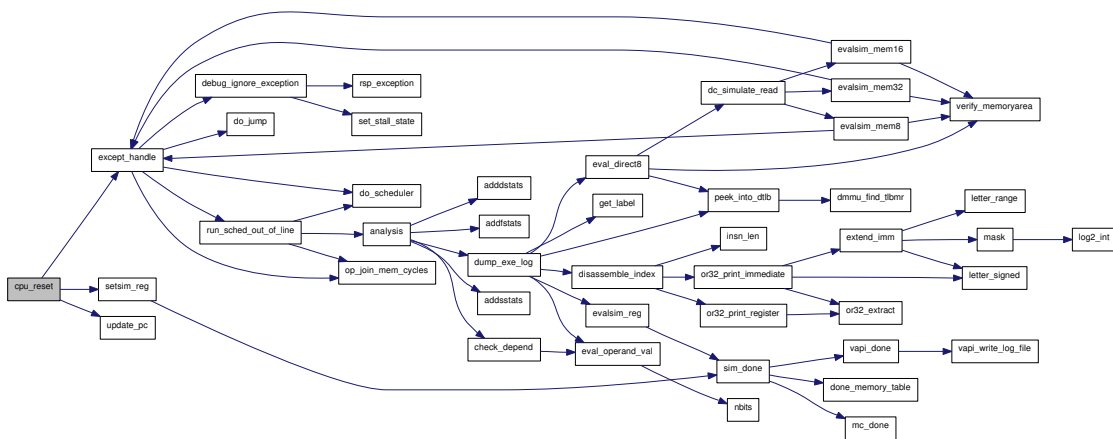
Here is the call graph for this function:



6.13.2.3 void cpu_reset ()

Reset the CPU

Here is the call graph for this function:

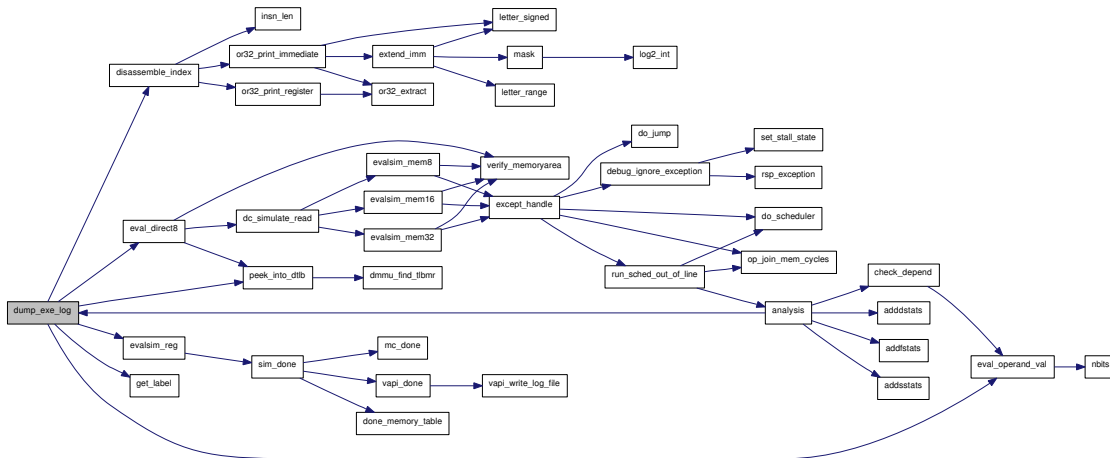


6.13.2.4 int depend_operands (struct iqueue_entry * prev, struct iqueue_entry * next)

6.13.2.5 void dump_exe_log ()

Outputs disassembled instruction

Here is the call graph for this function:

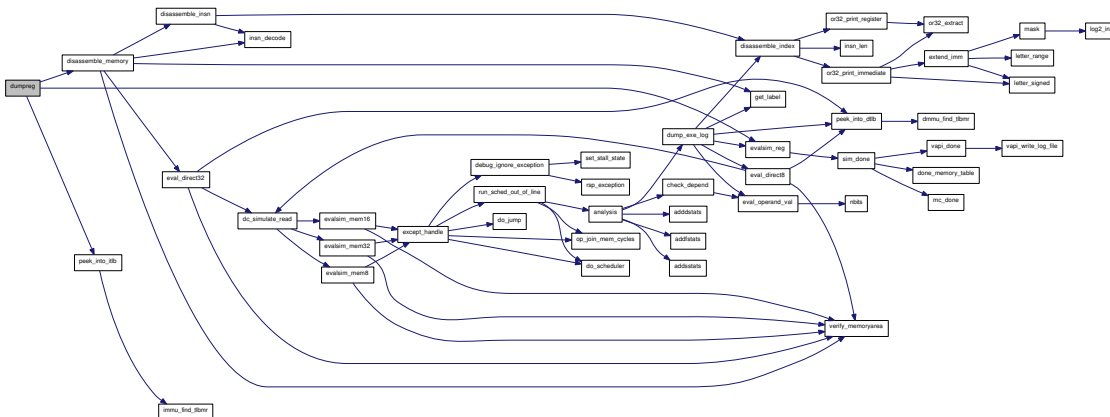


6.13.2.6 void dumpreg ()

Dump registers

Supports the CLI 'r' and 't' commands

Here is the call graph for this function:



6.13.2.7 uorreg_t eval_operand_val (uint32_t insn, struct insn_op_struct * opd)

Evaluates source operand operand

Implementation specific. Declared global, although this is only actually required for DYNAMIC_EXECUTION,

Parameters:

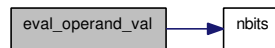
← *insn* The instruction

← *opd* The operand

Returns:

The value of the source operand

Here is the call graph for this function:

**6.13.2.8 uorreg_t evalsim_reg (unsigned int regno)**

Get an actual value of a specific register

Implementation specific. Abort if we are given a duff register. Only used externally to support [simprintf\(\)](#).

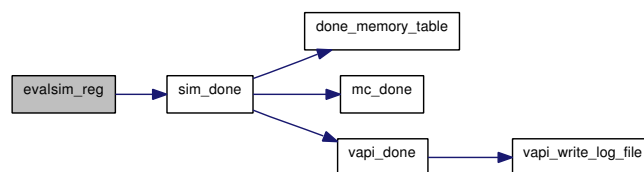
Parameters:

← *regno* The register of interest

Returns:

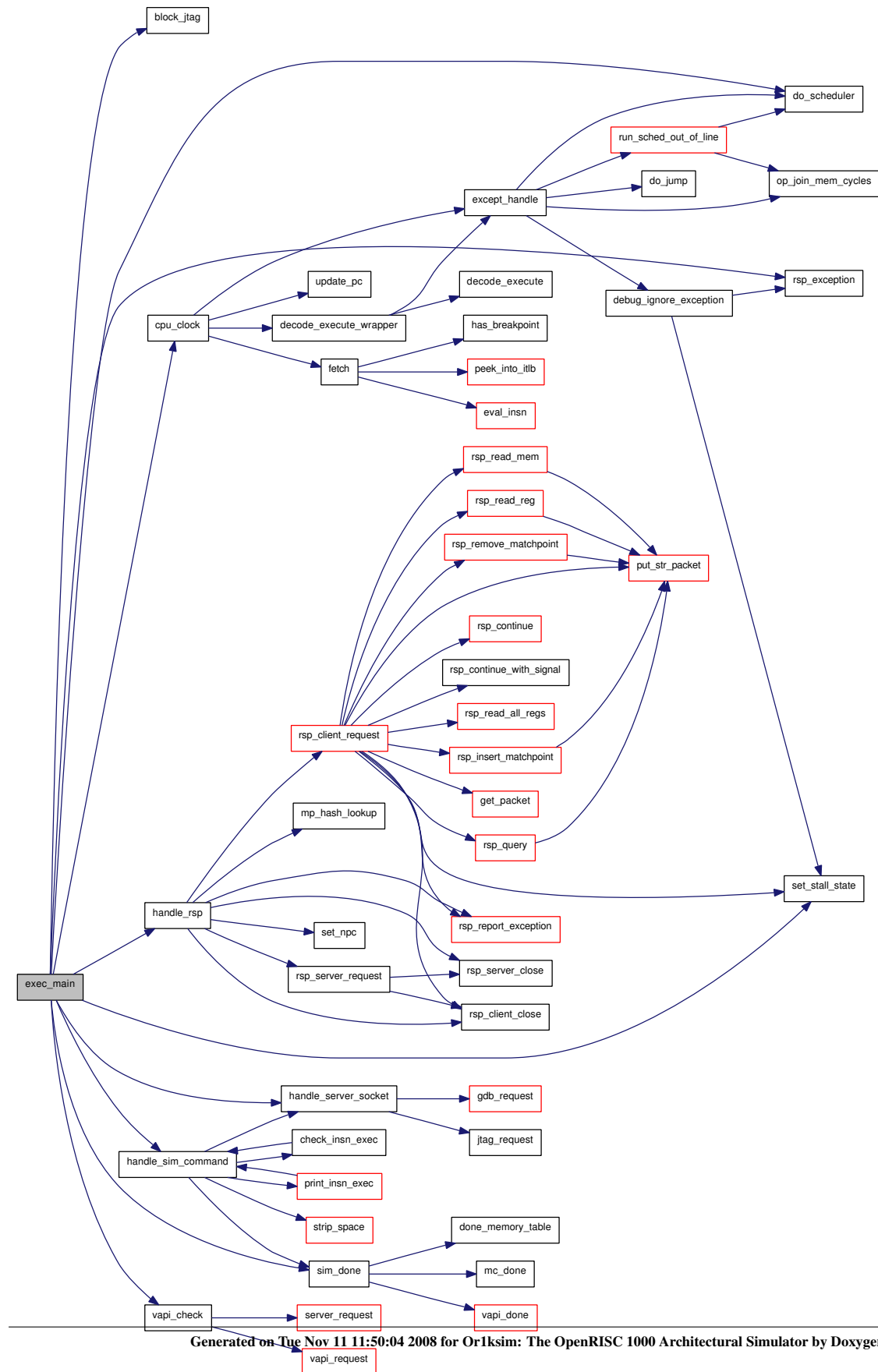
The value of the register

Here is the call graph for this function:

**6.13.2.9 void exec_main ()**

The main execution loop

Here is the call graph for this function:



6.13.2.10 void setsim_reg (unsigned int regno, uorreg_t value)

Set a specific register with value

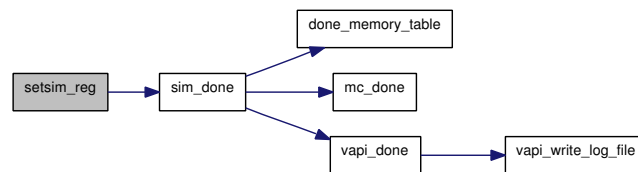
Implementation specific. Abort if we are given a duff register.

Parameters:

← *regno* The register of interest

← *value* The value to be set

Here is the call graph for this function:



6.13.3 Variable Documentation

6.13.3.1 struct cpu_state cpu_state

Current cpu state. Globally available.

6.13.3.2 int do_stats

Whether we are doing statistical analysis. Globally available

6.13.3.3 struct hist_exec* hist_exec_tail

History of execution. Globally available

6.13.3.4 oraddr_t pcnext

Temporary program counter. Globally available

6.13.3.5 int sbuf_total_cyc

Number of total store cycles. Globally available

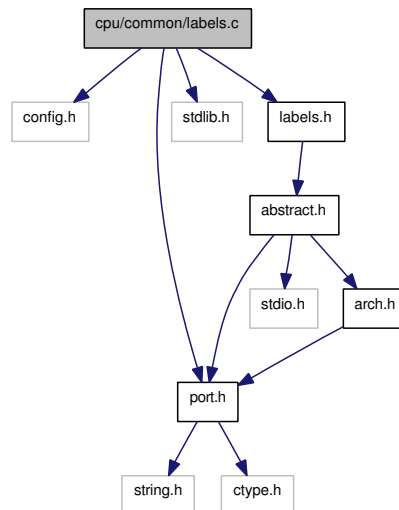
6.13.3.6 int sbuf_wait_cyc

Num cycles waiting for stores to complete. Globally available

6.14 cpu/common/labels.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "labels.h"
```

Include dependency graph for labels.c:



Defines

- `#define LABELS_HASH_SIZE 119`

Functions

- `void init_labels ()`
- `void add_label (oraddr_t addr, char *name)`
- `struct label_entry * get_label (oraddr_t addr)`
- `struct label_entry * find_label (char *name)`
- `oraddr_t eval_label (char *name)`
- `void init_breakpoints ()`
- `void add_breakpoint (oraddr_t addr)`
- `void remove_breakpoint (oraddr_t addr)`
- `void print_breakpoints ()`
- `int has_breakpoint (oraddr_t addr)`

Variables

- `struct breakpoint_entry * breakpoints`
- `static struct label_entry * label_hash [LABELS_HASH_SIZE]`

6.14.1 Define Documentation

6.14.1.1 `#define LABELS_HASH_SIZE 119`

6.14.2 Function Documentation

6.14.2.1 `void add_breakpoint (oraddr_t addr)`

6.14.2.2 `void add_label (oraddr_t addr, char * name)`

6.14.2.3 `oraddr_t eval_label (char * name)`

Here is the call graph for this function:



6.14.2.4 `struct label_entry* find_label (char * name)` [read]

6.14.2.5 `struct label_entry* get_label (oraddr_t addr)` [read]

6.14.2.6 `int has_breakpoint (oraddr_t addr)`

6.14.2.7 `void init_breakpoints ()`

6.14.2.8 `void init_labels ()`

6.14.2.9 `void print_breakpoints ()`

6.14.2.10 `void remove_breakpoint (oraddr_t addr)`

6.14.3 Variable Documentation

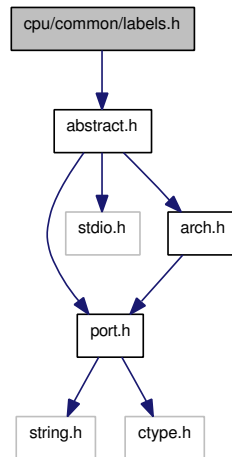
6.14.3.1 `struct breakpoint_entry* breakpoints`

6.14.3.2 `struct label_entry* label_hash[LABELS_HASH_SIZE]` [static]

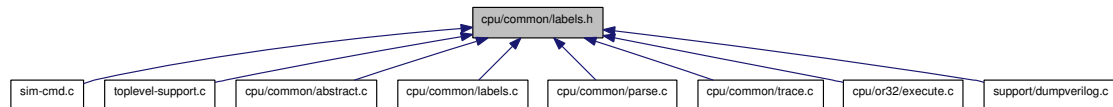
6.15 cpu/common/labels.h File Reference

```
#include "abstract.h"
```

Include dependency graph for labels.h:



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



Data Structures

- struct [label_entry](#)
- struct [breakpoint_entry](#)

Functions

- void [init_labels](#) ()
- void [add_label](#) ([oraddr_t](#) addr, char *name)
- struct [label_entry](#) * [get_label](#) ([oraddr_t](#) addr)
- struct [label_entry](#) * [find_label](#) (char *name)
- [oraddr_t](#) [eval_label](#) (char *name)
- void [add_breakpoint](#) ([oraddr_t](#) addr)
- void [remove_breakpoint](#) ([oraddr_t](#) addr)
- void [print_breakpoints](#) ()
- int [has_breakpoint](#) ([oraddr_t](#) addr)
- void [init_breakpoints](#) ()

Variables

- struct [breakpoint_entry](#) * [breakpoints](#)

6.15.1 Function Documentation

6.15.1.1 void `add_breakpoint` (`oraddr_t addr`)

6.15.1.2 void `add_label` (`oraddr_t addr`, `char * name`)

6.15.1.3 `oraddr_t eval_label` (`char * name`)

Here is the call graph for this function:



6.15.1.4 `struct label_entry*` `find_label` (`char * name`) [read]

6.15.1.5 `struct label_entry*` `get_label` (`oraddr_t addr`) [read]

6.15.1.6 int `has_breakpoint` (`oraddr_t addr`)

6.15.1.7 void `init_breakpoints` ()

6.15.1.8 void `init_labels` ()

6.15.1.9 void `print_breakpoints` ()

6.15.1.10 void `remove_breakpoint` (`oraddr_t addr`)

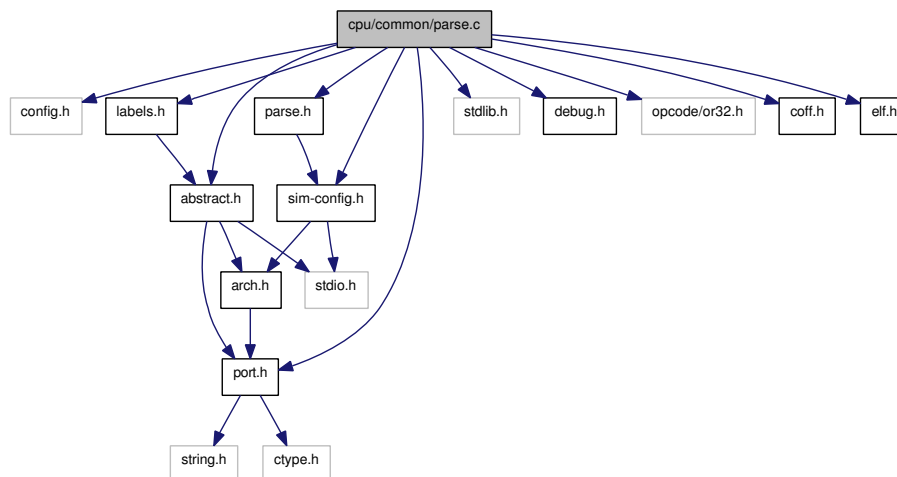
6.15.2 Variable Documentation

6.15.2.1 `struct breakpoint_entry*` `breakpoints`

6.16 cpu/common/parse.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "parse.h"
#include "sim-config.h"
#include "debug.h"
#include "abstract.h"
#include "opcode/or32.h"
#include "coff.h"
#include "elf.h"
#include "labels.h"
```

Include dependency graph for parse.c:



Defines

- #define [MEMORY_LEN](#) 0x100000000ULL
- #define [IMM_STATS](#) 0

Functions

- char * [rstrip](#) (char *dst, const char *src, int n)
- static [oraddr_t](#) [translate](#) ([oraddr_t](#) laddr, int *breakpoint)
- static void [addprogram](#) ([oraddr_t](#) address, [uint32_t](#) insn, int *breakpoint)
- static void [readfile_coff](#) (char *filename, short sections)
- static void [readsyms_coff](#) (char *filename, [uint32_t](#) symptr, [uint32_t](#) syms)
- static void [readfile_elf](#) (char *filename)
- static void [identifyfile](#) (char *filename)
- [uint32_t](#) [loadcode](#) (char *filename, [oraddr_t](#) startaddr, [oraddr_t](#) virtphy_transl)

Variables

- static unsigned int [freemem](#)
- static [oraddr_t](#) [transl_table](#)
- static uint32_t [transl_error](#)

6.16.1 Define Documentation

6.16.1.1 #define IMM_STATS 0

Whether to do immediate statistics. This seems to be for local debugging of [parse.c](#)

6.16.1.2 #define MEMORY_LEN 0x10000000ULL

6.16.2 Function Documentation

6.16.2.1 static void [addprogram](#) ([oraddr_t](#) *address*, [uint32_t](#) *insn*, [int](#) **breakpoint*) [static]

Add an instruction to the program

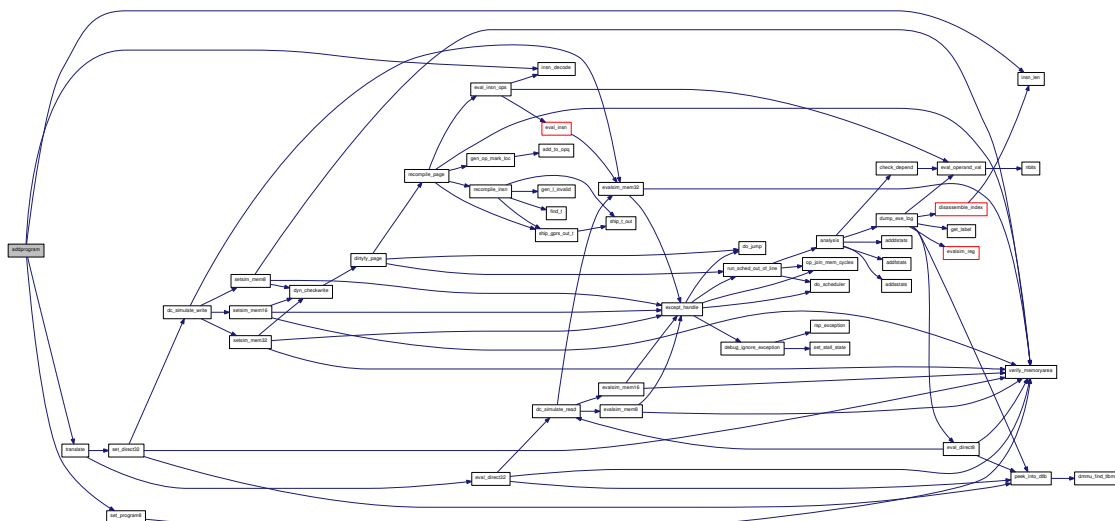
Note:

insn must be in big endian format

Parameters:

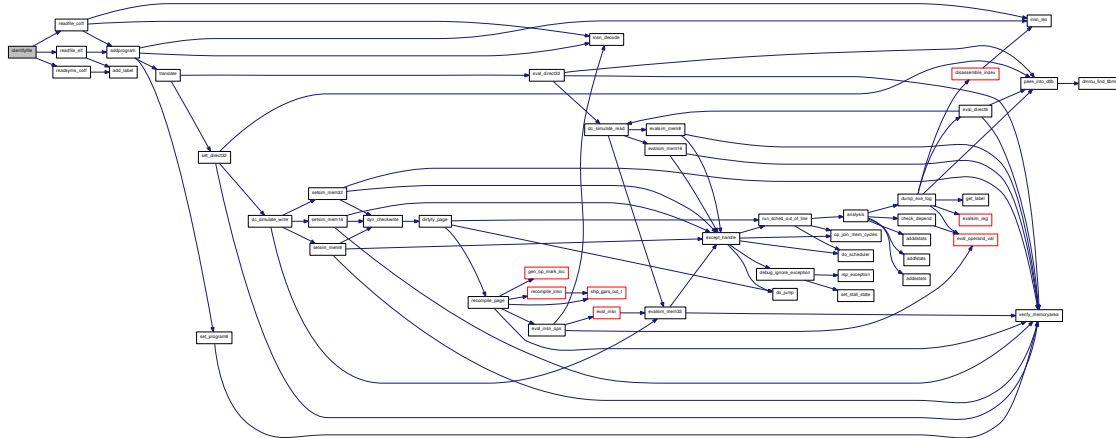
- ← *address* The address to use
- ← *insn* The instruction to add
- ← *breakpoint* Not used (it is passed to the [translate\(\)](#) function, which also does not use it.

Here is the call graph for this function:



6.16.2.2 static void identifyfile (char *filename) [static]

Here is the call graph for this function:



6.16.2.3 uint32_t loadcode (char *filename, oraddr_t startaddr, oraddr_t virtphy_transl)

Load file to memory

Loads file to memory starting at address startaddr and returns freemem.

Parameters:

← *filename* File to load

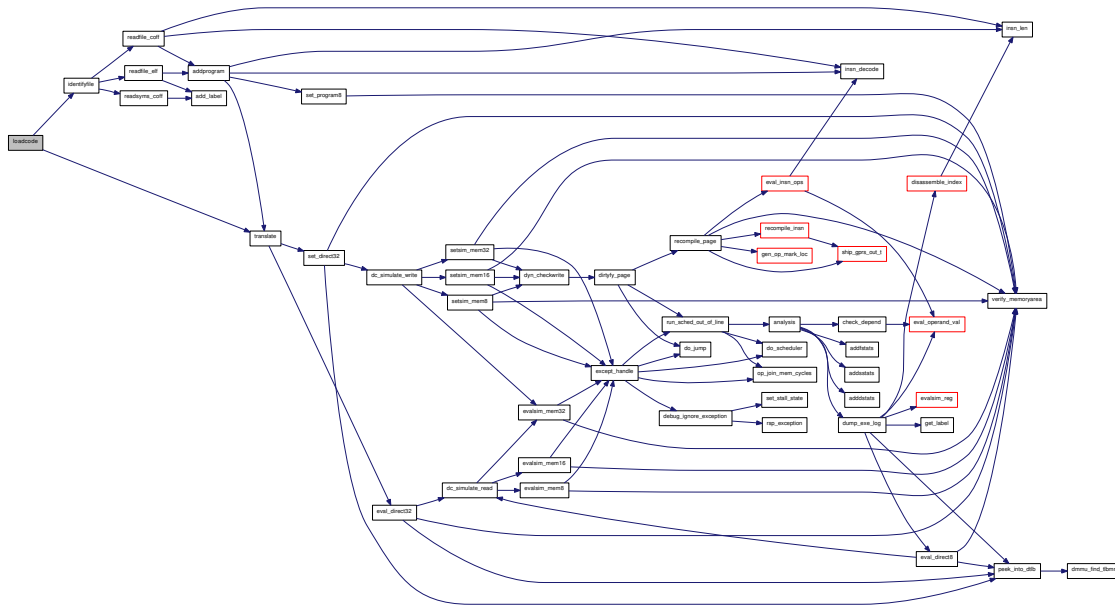
← *startaddr* Start address at which to load

← *virtphy_transl* Virtual to physical translation table if required. Only used for microkernel simulation, and not used in Ork1sim at present (set to NULL)

Returns:

zero on success, negative on failure.

Here is the call graph for this function:



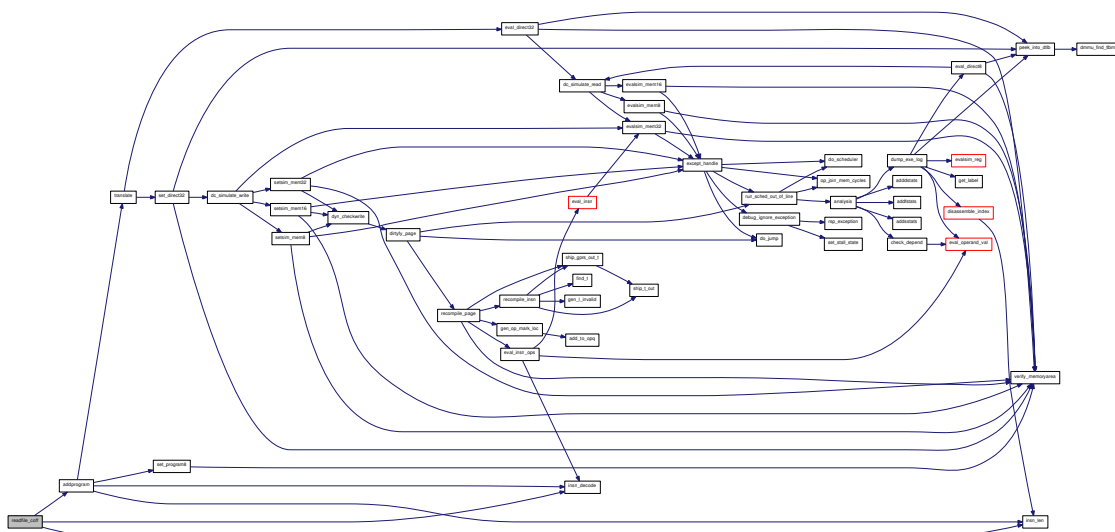
6.16.2.4 static void readfile_coff (char * *filename*, short *sections*) [static]

Load big-endian COFF file

Parameters:

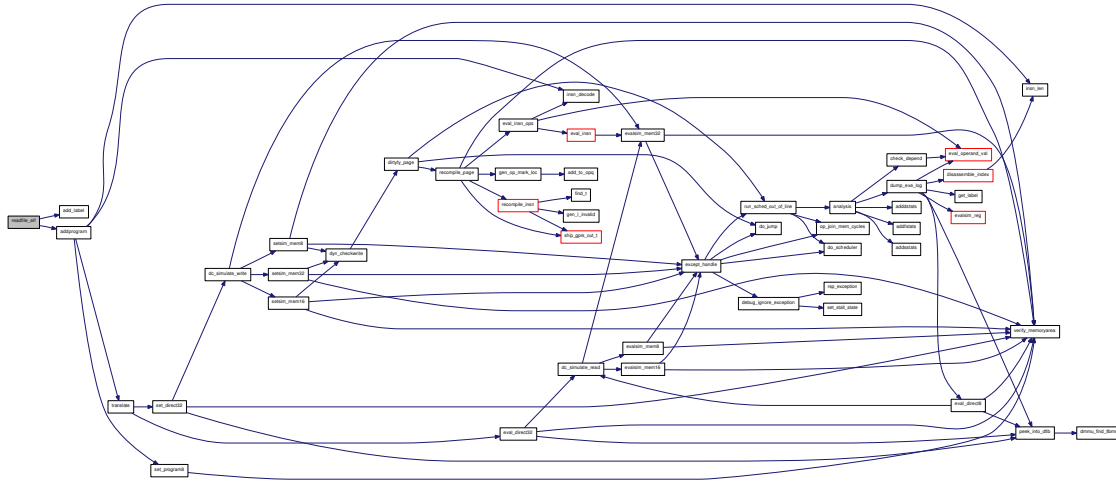
- ← *filename* File to load
- ← *sections* Number of sections in file

Here is the call graph for this function:



6.16.2.5 static void readfile_elf (char *filename) [static]

Here is the call graph for this function:



6.16.2.6 static void readsyms_coff (char *filename, uint32_t symptr, uint32_t syms) [static]

Here is the call graph for this function:



6.16.2.7 char* strstrip (char *dst, const char *src, int n)

Copy a string with null termination

This function is very similar to strncpy, except it null terminates the string. A global function also used by the CUC.

Parameters:

- ← *dst* The destination string
- ← *src* The source string
- ← *n* Number of chars to copy EXCLUDING the null terminator (i.e. dst had better have room for n+1 chars)

Returns:

A pointer to dst

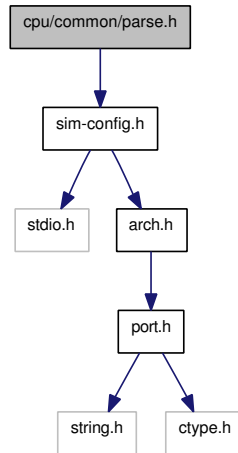
6.16.2.8 static oraddr_t translate (oraddr_t laddr, int *breakpoint) [static]

Translate logical to physical addresses for the loader

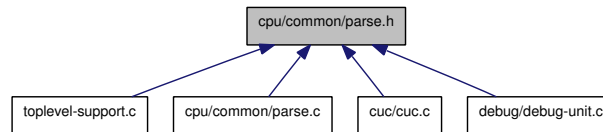
6.17 cpu/common/parse.h File Reference

```
#include "sim-config.h"
```

Include dependency graph for parse.h:



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



Functions

- `char * rstrip` (`char *dst`, `const char *src`, `int n`)
- `uint32_t loadcode` (`char *filename`, `oraddr_t startaddr`, `oraddr_t virtphy_transl`)

6.17.1 Function Documentation

6.17.1.1 `uint32_t loadcode` (`char *filename`, `oraddr_t startaddr`, `oraddr_t virtphy_transl`)

Load file to memory

Loads file to memory starting at address `startaddr` and returns `freemem`.

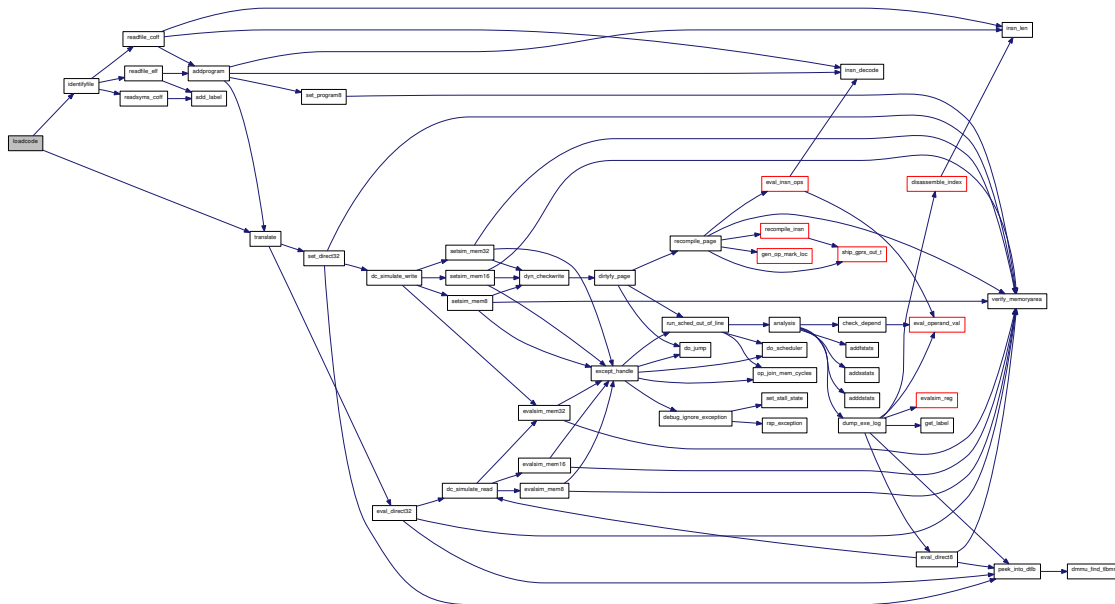
Parameters:

- ← *filename* File to load
- ← *startaddr* Start address at which to load
- ← *virtphy_transl* Virtual to physical translation table if required. Only used for microkernel simulation, and not used in Ork1sim at present (set to NULL)

Returns:

zero on success, negative on failure.

Here is the call graph for this function:



6.17.1.2 char* strstrip(char *dst, const char *src, int n)

Copy a string with null termination

This function is very similar to strncpy, except it null terminates the string. A global function also used by the CUC.

Parameters:

- ← *dst* The destination string
- ← *src* The source string
- ← *n* Number of chars to copy EXCLUDING the null terminator (i.e. dst had better have room for n+1 chars)

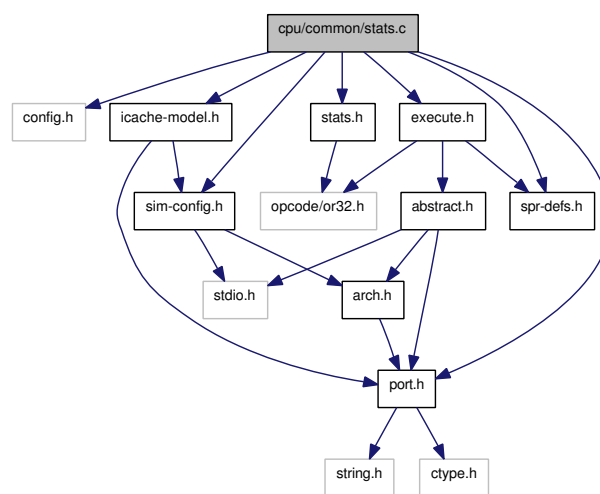
Returns:

A pointer to dst

6.18 cpu/common/stats.c File Reference

```
#include "config.h"
#include "port.h"
#include "stats.h"
#include "sim-config.h"
#include "icache-model.h"
#include "spr-defs.h"
#include "execute.h"
```

Include dependency graph for stats.c:



Data Structures

- struct [branchstat](#)
- struct [dstats_entry](#)
- struct [sstats_entry](#)
- struct [fstats_entry](#)

Defines

- #define [DSTATS_LEN](#) 3000
- #define [SSTATS_LEN](#) 300
- #define [FSTATS_LEN](#) 200
- #define [SD\(X\)](#) (X != 0 ? X : 1)

Functions

- void [addsstats](#) (int item, int cnt_dynamic)
- void [adddstats](#) (int item1, int item2, int cnt_dynamic, int depend)
- void [addfstats](#) (enum insn_type item1, enum insn_type item2, int cnt_dynamic, int depend)

- void `initstats` ()
- static void `printotherstats` (int *which*)
- void `printstats` (int *which*)

Variables

- static const char `func_unit_str` [30][30]
- struct `mstats_entry` `or1k_mstats` = { 0 }
- struct `cachestats_entry` `ic_stats` = { 0 }
- struct `cachestats_entry` `dc_stats` = { 0 }
- struct `immustats_entry` `immu_stats` = { 0 }
- struct `dmmustats_entry` `dmmu_stats` = { 0 }
- struct `raw_stats` `raw_stats`
- static struct `dstats_entry` `dstats` [DSTATS_LEN]
- static struct `sstats_entry` `sstats` [SSTATS_LEN]
- static struct `fstats_entry` `fstats` [FSTATS_LEN]

6.18.1 Define Documentation

6.18.1.1 `#define DSTATS_LEN 3000`

6.18.1.2 `#define FSTATS_LEN 200`

6.18.1.3 `#define SD(X) (X != 0 ? X : 1)`

6.18.1.4 `#define SSTATS_LEN 300`

6.18.2 Function Documentation

6.18.2.1 void `adddstats` (int *item1*, int *item2*, int *cnt_dynamic*, int *depend*)

6.18.2.2 void `addfstats` (enum *insn_type item1*, enum *insn_type item2*, int *cnt_dynamic*, int *depend*)

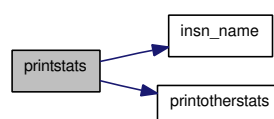
6.18.2.3 void `addsstats` (int *item*, int *cnt_dynamic*)

6.18.2.4 void `initstats` ()

6.18.2.5 static void `printotherstats` (int *which*) [`static`]

6.18.2.6 void `printstats` (int *which*)

Here is the call graph for this function:



6.18.3 Variable Documentation

6.18.3.1 struct cachestats_entry dc_stats = { 0 }

data cache stats

6.18.3.2 struct dmmustats_entry dmmu_stats = { 0 }

data MMU stats

6.18.3.3 struct dstats_entry dstats[DSTATS_LEN] [static]

dependency stats

6.18.3.4 struct fstats_entry fstats[FSTATS_LEN] [static]

func units stats

6.18.3.5 const char func_unit_str[30][30] [static]

Initial value:

```
{
  "unknown",
  "exception",
  "arith",
  "shift",
  "compare",
  "branch",
  "jump",
  "load",
  "store",
  "movimm",
  "move",
  "extend",
  "nop",
  "mac"
}
```

See also:

also enum `insn_type` in [abstract.h](#)

6.18.3.6 struct cachestats_entry ic_stats = { 0 }

instruction cache stats

6.18.3.7 struct immustats_entry immu_stats = { 0 }

insn MMU stats

6.18.3.8 struct mstats_entry or1k_mstats = { 0 }

misc units stats

6.18.3.9 struct raw_stats raw_stats

RAW hazard stats

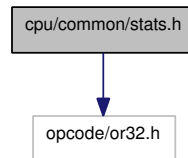
6.18.3.10 struct sstats_entry sstats[SSTATS_LEN] [static]

single stats

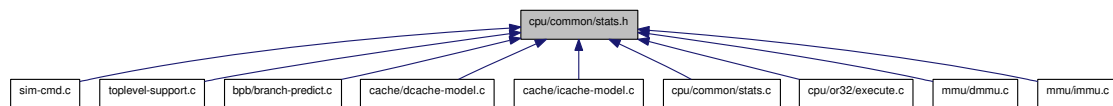
6.19 cpu/common/stats.h File Reference

```
#include "opcode/or32.h"
```

Include dependency graph for stats.h:



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



Data Structures

- struct [bpbstat](#)
- struct [bticstat](#)
- struct [mstats_entry](#)
- struct [cachestats_entry](#)
- struct [immustats_entry](#)
- struct [dmmustats_entry](#)
- struct [raw_stats](#)

Defines

- #define [RAW_RANGE](#) 1000

Functions

- void [addsstats](#) (int item, int cnt_dynamic)
- void [adddstats](#) (int item1, int item2, int cnt_dynamic, int depend)
- void [addfstats](#) (enum insn_type item1, enum insn_type item2, int cnt_dynamic, int depend)
- void [initstats](#) ()
- void [printstats](#) ()

Variables

- struct [mstats_entry](#) [or1k_mstats](#)
- struct [cachestats_entry](#) [ic_stats](#)
- struct [cachestats_entry](#) [dc_stats](#)
- struct [immustats_entry](#) [immu_stats](#)
- struct [dmmustats_entry](#) [dmmu_stats](#)
- struct [raw_stats](#) [raw_stats](#)

6.19.1 Define Documentation

6.19.1.1 `#define RAW_RANGE 1000`

6.19.2 Function Documentation

6.19.2.1 `void adddstats (int item1, int item2, int cnt_dynamic, int depend)`

6.19.2.2 `void addfstats (enum insn_type item1, enum insn_type item2, int cnt_dynamic, int depend)`

6.19.2.3 `void addsstats (int item, int cnt_dynamic)`

6.19.2.4 `void initstats ()`

6.19.2.5 `void printstats ()`

6.19.3 Variable Documentation

6.19.3.1 `struct cachestats_entry dc_stats`

data cache stats

6.19.3.2 `struct dmmustats_entry dmmu_stats`

data MMU stats

6.19.3.3 `struct cachestats_entry ic_stats`

instruction cache stats

6.19.3.4 `struct immustats_entry immu_stats`

insn MMU stats

6.19.3.5 `struct mstats_entry or1k_mstats`

misc units stats

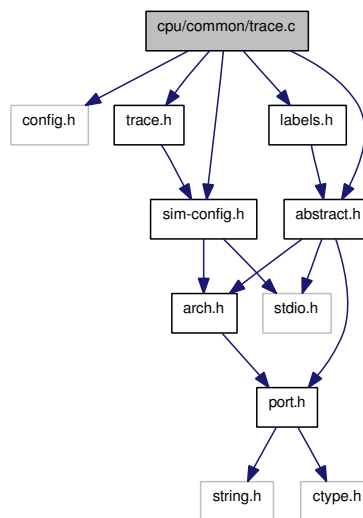
6.19.3.6 `struct raw_stats raw_stats`

RAW hazard stats

6.20 cpu/common/trace.c File Reference

```
#include "config.h"
#include "trace.h"
#include "sim-config.h"
#include "abstract.h"
#include "labels.h"
```

Include dependency graph for trace.c:



Functions

- void `set_insnbrkpoint` (`oraddr_t` *addr*)

6.20.1 Function Documentation

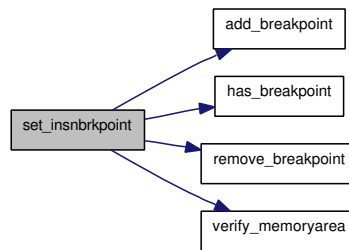
6.20.1.1 void `set_insnbrkpoint` (`oraddr_t` *addr*)

Set instruction execution breakpoint

Parameters:

- ← *addr* The address for the breakpoint

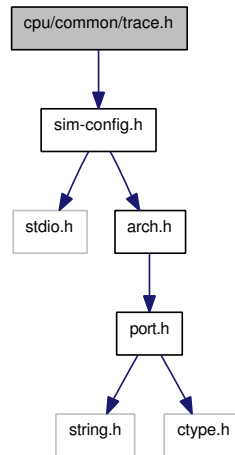
Here is the call graph for this function:



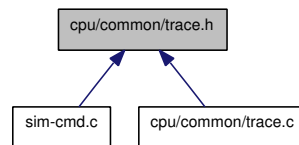
6.21 cpu/common/trace.h File Reference

```
#include "sim-config.h"
```

Include dependency graph for trace.h:



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



Functions

- void [set_insnbrkpoint](#) (`oraddr_t` *addr*)

6.21.1 Function Documentation

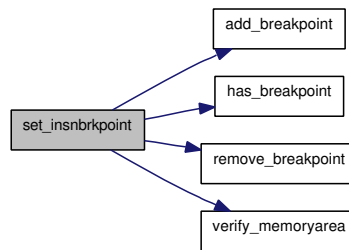
6.21.1.1 void set_insnbrkpoint (oraddr_t *addr*)

Set instruction execution breakpoint

Parameters:

- ← *addr* The address for the breakpoint

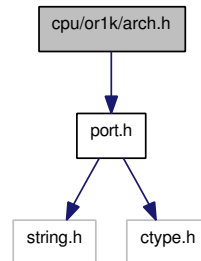
Here is the call graph for this function:



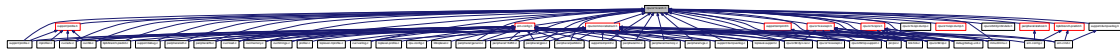
6.22 cpu/or1k/arch.h File Reference

```
#include "port.h"
```

Include dependency graph for arch.h:



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



Defines

- #define [LINK_REGNO](#) 9
- #define [ADDR_C\(c\)](#) [UINT32_C\(c\)](#)
- #define [REG_C\(c\)](#) [UINT32_C\(c\)](#)
- #define [PRIxADDR](#) "08" [PRIx32](#)
- #define [PRIxREG](#) "08" [PRIx32](#)
- #define [PRIdREG](#) [PRId32](#)

Typedefs

- typedef [uint32_t](#) [oraddr_t](#)
- typedef [uint32_t](#) [uorreg_t](#)
- typedef [int32_t](#) [orreg_t](#)

6.22.1 Define Documentation

6.22.1.1 #define ADDR_C(c) UINT32_C(c)

6.22.1.2 #define LINK_REGNO 9

Index of the link register

6.22.1.3 #define PRIdREG PRId32

print an openrisc register in decimals

6.22.1.4 #define PRIxADDR "08" PRIx32

print an openrisc address in hex

6.22.1.5 #define PRIxREG "08" PRIx32

print an openrisc register in hex

6.22.1.6 #define REG_C(c) UINT32_C(c)**6.22.2 Typedef Documentation****6.22.2.1 typedef uint32_t oraddr_t**

Address as addressed by openrisc

6.22.2.2 typedef int32_t orreg_t

A signed register of openrisc

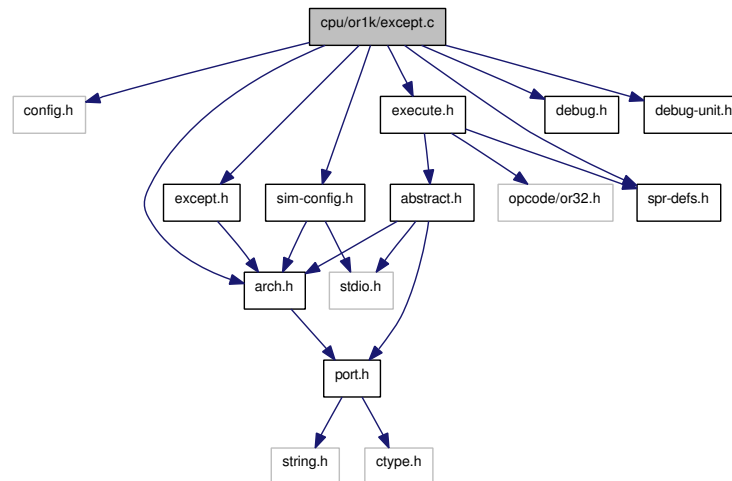
6.22.2.3 typedef uint32_t uorreg_t

An unsigned register of openrisc

6.23 cpu/or1k/except.c File Reference

```
#include "config.h"
#include "except.h"
#include "sim-config.h"
#include "arch.h"
#include "debug.h"
#include "spr-defs.h"
#include "execute.h"
#include "debug-unit.h"
```

Include dependency graph for except.c:



Functions

- void `op_join_mem_cycles` (void)
- void `except_handle` (`oraddr_t` except, `oraddr_t` ea)

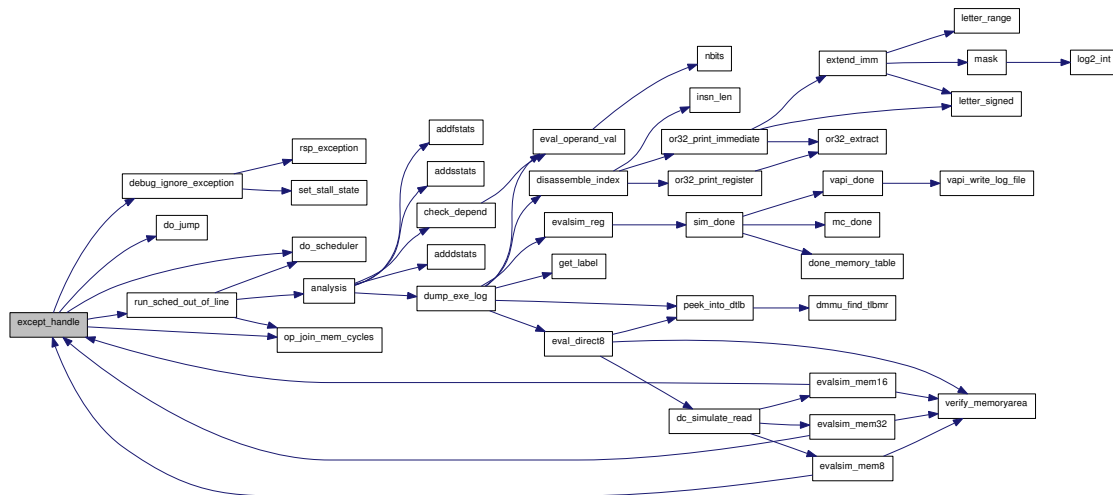
Variables

- int `except_pending` = 0

6.23.1 Function Documentation

6.23.1.1 void except_handle (oraddr_t except, oraddr_t ea)

Here is the call graph for this function:



6.23.1.2 void op_join_mem_cycles (void)

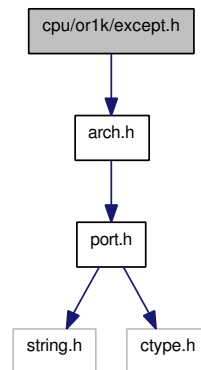
6.23.2 Variable Documentation

6.23.2.1 int except_pending = 0

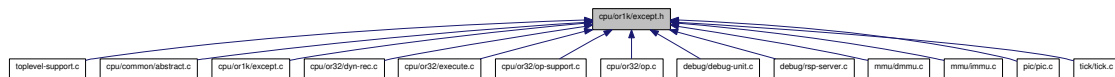
6.24 cpu/or1k/except.h File Reference

```
#include "arch.h"
```

Include dependency graph for except.h:



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



Defines

- #define [EXCEPT_NONE](#) 0x0000
- #define [EXCEPT_RESET](#) 0x0100
- #define [EXCEPT_BUSERR](#) 0x0200
- #define [EXCEPT_DPF](#) 0x0300
- #define [EXCEPT_IPF](#) 0x0400
- #define [EXCEPT_TICK](#) 0x0500
- #define [EXCEPT_ALIGN](#) 0x0600
- #define [EXCEPT_ILLEGAL](#) 0x0700
- #define [EXCEPT_INT](#) 0x0800
- #define [EXCEPT_DTLBMISS](#) 0x0900
- #define [EXCEPT_ITLBMISS](#) 0x0a00
- #define [EXCEPT_RANGE](#) 0x0b00
- #define [EXCEPT_SYSCALL](#) 0x0c00
- #define [EXCEPT_FPE](#) 0x0d00
- #define [EXCEPT_TRAP](#) 0x0e00

Functions

- void [except_handle](#) ([oraddr_t](#) except, [oraddr_t](#) ea)

Variables

- int [except_pending](#)

6.24.1 Define Documentation

6.24.1.1 `#define EXCEPT_ALIGN 0x0600`

6.24.1.2 `#define EXCEPT_BUSERR 0x0200`

6.24.1.3 `#define EXCEPT_DPF 0x0300`

6.24.1.4 `#define EXCEPT_DTLBMISS 0x0900`

6.24.1.5 `#define EXCEPT_FPE 0x0d00`

6.24.1.6 `#define EXCEPT_ILLEGAL 0x0700`

6.24.1.7 `#define EXCEPT_INT 0x0800`

6.24.1.8 `#define EXCEPT_IPF 0x0400`

6.24.1.9 `#define EXCEPT_ITLBMISS 0x0a00`

6.24.1.10 `#define EXCEPT_NONE 0x0000`

6.24.1.11 `#define EXCEPT_RANGE 0x0b00`

6.24.1.12 `#define EXCEPT_RESET 0x0100`

6.24.1.13 `#define EXCEPT_SYSCALL 0x0c00`

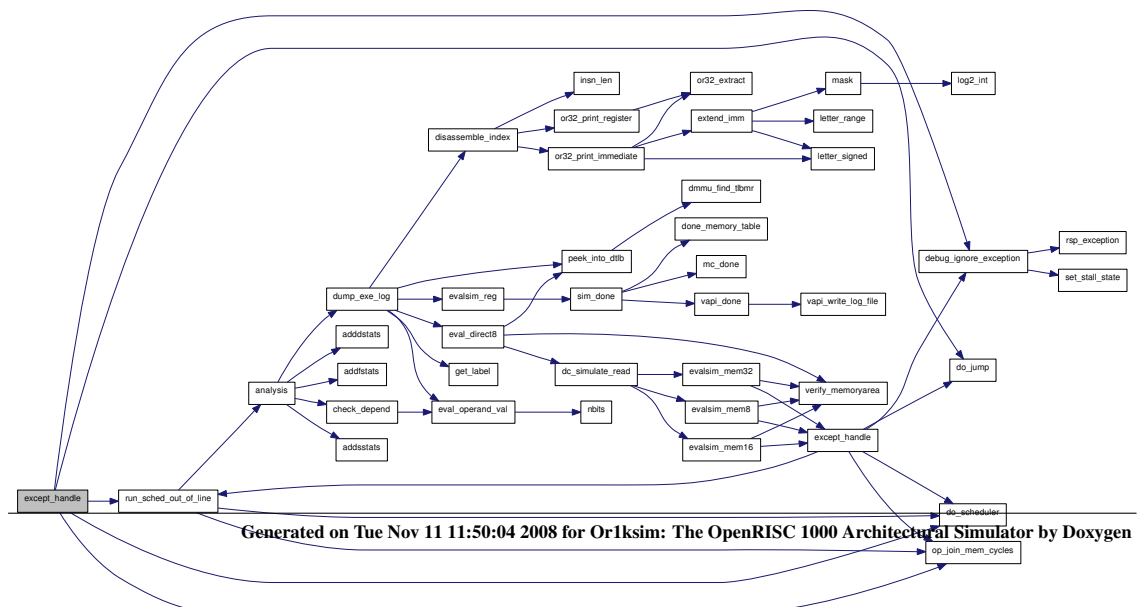
6.24.1.14 `#define EXCEPT_TICK 0x0500`

6.24.1.15 `#define EXCEPT_TRAP 0x0e00`

6.24.2 Function Documentation

6.24.2.1 `void except_handle (oraddr_t except, oraddr_t ea)`

Here is the call graph for this function:



6.24.3 Variable Documentation

6.24.3.1 int except_pending

- #define [SPR_ITLBMR_BASE](#)(WAY) (SPRGROUP_IMMU + 0x200 + (WAY) * 0x100)
- #define [SPR_ITLBMR_LAST](#)(WAY) (SPRGROUP_IMMU + 0x27f + (WAY) * 0x100)
- #define [SPR_ITLBTR_BASE](#)(WAY) (SPRGROUP_IMMU + 0x280 + (WAY) * 0x100)
- #define [SPR_ITLBTR_LAST](#)(WAY) (SPRGROUP_IMMU + 0x2ff + (WAY) * 0x100)
- #define [SPR_DCCR](#) (SPRGROUP_DC + 0)
- #define [SPR_DCBPR](#) (SPRGROUP_DC + 1)
- #define [SPR_DCBFR](#) (SPRGROUP_DC + 2)
- #define [SPR_DCBIR](#) (SPRGROUP_DC + 3)
- #define [SPR_DCBWR](#) (SPRGROUP_DC + 4)
- #define [SPR_DCBLR](#) (SPRGROUP_DC + 5)
- #define [SPR_DCR_BASE](#)(WAY) (SPRGROUP_DC + 0x200 + (WAY) * 0x200)
- #define [SPR_DCR_LAST](#)(WAY) (SPRGROUP_DC + 0x3ff + (WAY) * 0x200)
- #define [SPR_ICCR](#) (SPRGROUP_IC + 0)
- #define [SPR_ICBPR](#) (SPRGROUP_IC + 1)
- #define [SPR_ICBIR](#) (SPRGROUP_IC + 2)
- #define [SPR_ICBLR](#) (SPRGROUP_IC + 3)
- #define [SPR_ICR_BASE](#)(WAY) (SPRGROUP_IC + 0x200 + (WAY) * 0x200)
- #define [SPR_ICR_LAST](#)(WAY) (SPRGROUP_IC + 0x3ff + (WAY) * 0x200)
- #define [SPR_MACLO](#) (SPRGROUP_MAC + 1)
- #define [SPR_MACHI](#) (SPRGROUP_MAC + 2)
- #define [SPR_DVR](#)(N) (SPRGROUP_D + (N))
- #define [SPR_DCR](#)(N) (SPRGROUP_D + 8 + (N))
- #define [SPR_DMR1](#) (SPRGROUP_D + 16)
- #define [SPR_DMR2](#) (SPRGROUP_D + 17)
- #define [SPR_DWCR0](#) (SPRGROUP_D + 18)
- #define [SPR_DWCR1](#) (SPRGROUP_D + 19)
- #define [SPR_DSR](#) (SPRGROUP_D + 20)
- #define [SPR_DRR](#) (SPRGROUP_D + 21)
- #define [SPR_PCCR](#)(N) (SPRGROUP_PC + (N))
- #define [SPR_PCMR](#)(N) (SPRGROUP_PC + 8 + (N))
- #define [SPR_PMR](#) (SPRGROUP_PM + 0)
- #define [SPR_PICMR](#) (SPRGROUP_PIC + 0)
- #define [SPR_PICPR](#) (SPRGROUP_PIC + 1)
- #define [SPR_PICSR](#) (SPRGROUP_PIC + 2)
- #define [SPR_TTMR](#) (SPRGROUP_TT + 0)
- #define [SPR_TTCR](#) (SPRGROUP_TT + 1)
- #define [SPR_VR_VER](#) 0xff000000
- #define [SPR_VR_CFG](#) 0x00ff0000
- #define [SPR_VR_RES](#) 0x00ff0000
- #define [SPR_VR_REV](#) 0x0000003f
- #define [SPR_VR_VER_OFF](#) 24
- #define [SPR_VR_CFG_OFF](#) 16
- #define [SPR_VR_REV_OFF](#) 0
- #define [SPR_UPR_UP](#) 0x00000001
- #define [SPR_UPR_DCP](#) 0x00000002
- #define [SPR_UPR_ICP](#) 0x00000004
- #define [SPR_UPR_DMP](#) 0x00000008
- #define [SPR_UPR_IMP](#) 0x00000010
- #define [SPR_UPR_MP](#) 0x00000020
- #define [SPR_UPR_DUP](#) 0x00000040

- #define `SPR_UPR_PCUP` 0x00000080
- #define `SPR_UPR_PMP` 0x00000100
- #define `SPR_UPR_PICP` 0x00000200
- #define `SPR_UPR_TTP` 0x00000400
- #define `SPR_UPR_RES` 0x00fe0000
- #define `SPR_UPR_CUP` 0xff000000
- #define `SPR_CPUCFGR_NSGF` 0x0000000f
- #define `SPR_CPUCFGR_CGF` 0x00000010
- #define `SPR_CPUCFGR_OB32S` 0x00000020
- #define `SPR_CPUCFGR_OB64S` 0x00000040
- #define `SPR_CPUCFGR_OF32S` 0x00000080
- #define `SPR_CPUCFGR_OF64S` 0x00000100
- #define `SPR_CPUCFGR_OV64S` 0x00000200
- #define `SPR_CPUCFGR_RES` 0xffffc00
- #define `SPR_DCFGR_NDP` 0x00000007
- #define `SPR_DCFGR_NDP1` 0x00000000
- #define `SPR_DCFGR_NDP2` 0x00000001
- #define `SPR_DCFGR_NDP3` 0x00000002
- #define `SPR_DCFGR_NDP4` 0x00000003
- #define `SPR_DCFGR_NDP5` 0x00000004
- #define `SPR_DCFGR_NDP6` 0x00000005
- #define `SPR_DCFGR_NDP7` 0x00000006
- #define `SPR_DCFGR_NDP8` 0x00000007
- #define `SPR_DCFGR_WPCI` 0x00000008
- #define `MATCHPOINTS_TO_NDP`(n)
- #define `MAX_MATCHPOINTS` 8
- #define `MAX_WATCHPOINTS` (MAX_MATCHPOINTS + 2)
- #define `SPR_SR_SM` 0x00000001
- #define `SPR_SR_TEE` 0x00000002
- #define `SPR_SR_IEE` 0x00000004
- #define `SPR_SR_DCE` 0x00000008
- #define `SPR_SR_ICE` 0x00000010
- #define `SPR_SR_DME` 0x00000020
- #define `SPR_SR_IME` 0x00000040
- #define `SPR_SR_LEE` 0x00000080
- #define `SPR_SR_CE` 0x00000100
- #define `SPR_SR_F` 0x00000200
- #define `SPR_SR_CY` 0x00000400
- #define `SPR_SR_OV` 0x00000800
- #define `SPR_SR_OVE` 0x00001000
- #define `SPR_SR_DSX` 0x00002000
- #define `SPR_SR_EPH` 0x00004000
- #define `SPR_SR_FO` 0x00008000
- #define `SPR_SR_SUMRA` 0x00010000
- #define `SPR_SR_RES` 0x0ffe0000
- #define `SPR_SR_CID` 0xf0000000
- #define `SPR_DMMUCR_P2S` 0x0000003e
- #define `SPR_DMMUCR_P1S` 0x000007c0
- #define `SPR_DMMUCR_VADDR_WIDTH` 0x0000f800
- #define `SPR_DMMUCR_PADDR_WIDTH` 0x000f0000

- #define `SPR_IMMUCR_P2S` 0x0000003e
- #define `SPR_IMMUCR_P1S` 0x000007c0
- #define `SPR_IMMUCR_VADDR_WIDTH` 0x0000f800
- #define `SPR_IMMUCR_PADDR_WIDTH` 0x000f0000
- #define `SPR_DTLBMR_V` 0x00000001
- #define `SPR_DTLBMR_PL1` 0x00000002
- #define `SPR_DTLBMR_CID` 0x0000003c
- #define `SPR_DTLBMR_LRU` 0x000000c0
- #define `SPR_DTLBMR_VPN` 0xffff000
- #define `SPR_DTLBTR_CC` 0x00000001
- #define `SPR_DTLBTR_CI` 0x00000002
- #define `SPR_DTLBTR_WBC` 0x00000004
- #define `SPR_DTLBTR_WOM` 0x00000008
- #define `SPR_DTLBTR_A` 0x00000010
- #define `SPR_DTLBTR_D` 0x00000020
- #define `SPR_DTLBTR_URE` 0x00000040
- #define `SPR_DTLBTR_UWE` 0x00000080
- #define `SPR_DTLBTR_SRE` 0x00000100
- #define `SPR_DTLBTR_SWE` 0x00000200
- #define `SPR_DTLBTR_PPN` 0xffff000
- #define `SPR_ITLBMR_V` 0x00000001
- #define `SPR_ITLBMR_PL1` 0x00000002
- #define `SPR_ITLBMR_CID` 0x0000003c
- #define `SPR_ITLBMR_LRU` 0x000000c0
- #define `SPR_ITLBMR_VPN` 0xffff000
- #define `SPR_ITLBTR_CC` 0x00000001
- #define `SPR_ITLBTR_CI` 0x00000002
- #define `SPR_ITLBTR_WBC` 0x00000004
- #define `SPR_ITLBTR_WOM` 0x00000008
- #define `SPR_ITLBTR_A` 0x00000010
- #define `SPR_ITLBTR_D` 0x00000020
- #define `SPR_ITLBTR_SXE` 0x00000040
- #define `SPR_ITLBTR_UXE` 0x00000080
- #define `SPR_ITLBTR_PPN` 0xffff000
- #define `SPR_DCCR_EW` 0x000000ff
- #define `SPR_ICCR_EW` 0x000000ff
- #define `SPR_DCCFGR_NCW` 0x00000007
- #define `SPR_DCCFGR_NCS` 0x00000078
- #define `SPR_DCCFGR_CBS` 0x00000080
- #define `SPR_DCCFGR_CWS` 0x00000100
- #define `SPR_DCCFGR_CCRI` 0x00000200
- #define `SPR_DCCFGR_CBIRI` 0x00000400
- #define `SPR_DCCFGR_CBPRI` 0x00000800
- #define `SPR_DCCFGR_CBLRI` 0x00001000
- #define `SPR_DCCFGR_CBFRI` 0x00002000
- #define `SPR_DCCFGR_CBWBRI` 0x00004000
- #define `SPR_DCCFGR_NCW_OFF` 0
- #define `SPR_DCCFGR_NCS_OFF` 3
- #define `SPR_DCCFGR_CBS_OFF` 7
- #define `SPR_ICCFGR_NCW` 0x00000007

- #define `SPR_ICCFGR_NCS` 0x00000078
- #define `SPR_ICCFGR_CBS` 0x00000080
- #define `SPR_ICCFGR_CCRI` 0x00000200
- #define `SPR_ICCFGR_CBIRI` 0x00000400
- #define `SPR_ICCFGR_CBPRI` 0x00000800
- #define `SPR_ICCFGR_CBLRI` 0x00001000
- #define `SPR_ICCFGR_NCW_OFF` 0
- #define `SPR_ICCFGR_NCS_OFF` 3
- #define `SPR_ICCFGR_CBS_OFF` 7
- #define `SPR_DMMUCFGR_NTW` 0x00000003
- #define `SPR_DMMUCFGR_NTS` 0x0000001C
- #define `SPR_DMMUCFGR_NAE` 0x000000E0
- #define `SPR_DMMUCFGR_CRI` 0x00000100
- #define `SPR_DMMUCFGR_PRI` 0x00000200
- #define `SPR_DMMUCFGR_TEIRI` 0x00000400
- #define `SPR_DMMUCFGR_HTR` 0x00000800
- #define `SPR_DMMUCFGR_NTW_OFF` 0
- #define `SPR_DMMUCFGR_NTS_OFF` 2
- #define `SPR_IMMUCFGR_NTW` 0x00000003
- #define `SPR_IMMUCFGR_NTS` 0x0000001C
- #define `SPR_IMMUCFGR_NAE` 0x000000E0
- #define `SPR_IMMUCFGR_CRI` 0x00000100
- #define `SPR_IMMUCFGR_PRI` 0x00000200
- #define `SPR_IMMUCFGR_TEIRI` 0x00000400
- #define `SPR_IMMUCFGR_HTR` 0x00000800
- #define `SPR_IMMUCFGR_NTW_OFF` 0
- #define `SPR_IMMUCFGR_NTS_OFF` 2
- #define `SPR_DCR_DP` 0x00000001
- #define `SPR_DCR_CC` 0x0000000e
- #define `SPR_DCR_SC` 0x00000010
- #define `SPR_DCR_CT` 0x000000e0
- #define `SPR_DCR_CC_MASKED` 0x00000000
- #define `SPR_DCR_CC_EQUAL` 0x00000002
- #define `SPR_DCR_CC_LESS` 0x00000004
- #define `SPR_DCR_CC_LESSE` 0x00000006
- #define `SPR_DCR_CC_GREAT` 0x00000008
- #define `SPR_DCR_CC_GREATE` 0x0000000a
- #define `SPR_DCR_CC_NEQUAL` 0x0000000c
- #define `SPR_DCR_CT_DISABLED` 0x00000000
- #define `SPR_DCR_CT_IFEA` 0x00000020
- #define `SPR_DCR_CT_LEA` 0x00000040
- #define `SPR_DCR_CT_SEA` 0x00000060
- #define `SPR_DCR_CT_LD` 0x00000080
- #define `SPR_DCR_CT_SD` 0x000000a0
- #define `SPR_DCR_CT_LSEA` 0x000000c0
- #define `SPR_DCR_CT_LSD` 0x000000e0
- #define `SPR_DMR1_CW` 0x000fffff
- #define `SPR_DMR1_CW0_AND` 0x00000001
- #define `SPR_DMR1_CW0_OR` 0x00000002
- #define `SPR_DMR1_CW0` (`SPR_DMR1_CW0_AND` | `SPR_DMR1_CW0_OR`)

- #define `SPR_DMR1_CW1_AND` 0x00000004
- #define `SPR_DMR1_CW1_OR` 0x00000008
- #define `SPR_DMR1_CW1` (SPR_DMR1_CW1_AND | SPR_DMR1_CW1_OR)
- #define `SPR_DMR1_CW2_AND` 0x00000010
- #define `SPR_DMR1_CW2_OR` 0x00000020
- #define `SPR_DMR1_CW2` (SPR_DMR1_CW2_AND | SPR_DMR1_CW2_OR)
- #define `SPR_DMR1_CW3_AND` 0x00000040
- #define `SPR_DMR1_CW3_OR` 0x00000080
- #define `SPR_DMR1_CW3` (SPR_DMR1_CW3_AND | SPR_DMR1_CW3_OR)
- #define `SPR_DMR1_CW4_AND` 0x00000100
- #define `SPR_DMR1_CW4_OR` 0x00000200
- #define `SPR_DMR1_CW4` (SPR_DMR1_CW4_AND | SPR_DMR1_CW4_OR)
- #define `SPR_DMR1_CW5_AND` 0x00000400
- #define `SPR_DMR1_CW5_OR` 0x00000800
- #define `SPR_DMR1_CW5` (SPR_DMR1_CW5_AND | SPR_DMR1_CW5_OR)
- #define `SPR_DMR1_CW6_AND` 0x00001000
- #define `SPR_DMR1_CW6_OR` 0x00002000
- #define `SPR_DMR1_CW6` (SPR_DMR1_CW6_AND | SPR_DMR1_CW6_OR)
- #define `SPR_DMR1_CW7_AND` 0x00004000
- #define `SPR_DMR1_CW7_OR` 0x00008000
- #define `SPR_DMR1_CW7` (SPR_DMR1_CW7_AND | SPR_DMR1_CW7_OR)
- #define `SPR_DMR1_CW8_AND` 0x00010000
- #define `SPR_DMR1_CW8_OR` 0x00020000
- #define `SPR_DMR1_CW8` (SPR_DMR1_CW8_AND | SPR_DMR1_CW8_OR)
- #define `SPR_DMR1_CW9_AND` 0x00040000
- #define `SPR_DMR1_CW9_OR` 0x00080000
- #define `SPR_DMR1_CW9` (SPR_DMR1_CW9_AND | SPR_DMR1_CW9_OR)
- #define `SPR_DMR1_RES1` 0x00300000
- #define `SPR_DMR1_ST` 0x00400000
- #define `SPR_DMR1_BT` 0x00800000
- #define `SPR_DMR1_RES2` 0xff000000
- #define `SPR_DMR2_WCE0` 0x00000001
- #define `SPR_DMR2_WCE1` 0x00000002
- #define `SPR_DMR2_AWTC` 0x0000ffc
- #define `SPR_DMR2_AWTC_OFF` 2
- #define `SPR_DMR2_WGB` 0x003ff000
- #define `SPR_DMR2_WGB_OFF` 12
- #define `SPR_DMR2_WBS` 0xffc00000
- #define `SPR_DMR2_WBS_OFF` 22
- #define `SPR_DWCR_COUNT` 0x0000ffff
- #define `SPR_DWCR_MATCH` 0xffff0000
- #define `SPR_DWCR_MATCH_OFF` 16
- #define `SPR_DSR_RSTE` 0x00000001
- #define `SPR_DSR_BUSEE` 0x00000002
- #define `SPR_DSR_DPFE` 0x00000004
- #define `SPR_DSR_IPFE` 0x00000008
- #define `SPR_DSR_TTE` 0x00000010
- #define `SPR_DSR_AE` 0x00000020
- #define `SPR_DSR_IIE` 0x00000040
- #define `SPR_DSR_IE` 0x00000080

- #define `SPR_DSR_DME` 0x00000100
- #define `SPR_DSR_IME` 0x00000200
- #define `SPR_DSR_RE` 0x00000400
- #define `SPR_DSR_SCE` 0x00000800
- #define `SPR_DSR_SSE` 0x00001000
- #define `SPR_DSR_TE` 0x00002000
- #define `SPR_DRR_RSTE` 0x00000001
- #define `SPR_DRR_BUSEE` 0x00000002
- #define `SPR_DRR_DPFE` 0x00000004
- #define `SPR_DRR_IPFE` 0x00000008
- #define `SPR_DRR_TTE` 0x00000010
- #define `SPR_DRR_AE` 0x00000020
- #define `SPR_DRR_IIE` 0x00000040
- #define `SPR_DRR_IE` 0x00000080
- #define `SPR_DRR_DME` 0x00000100
- #define `SPR_DRR_IME` 0x00000200
- #define `SPR_DRR_RE` 0x00000400
- #define `SPR_DRR_SCE` 0x00000800
- #define `SPR_DRR_TE` 0x00001000
- #define `SPR_PCMR_CP` 0x00000001
- #define `SPR_PCMR_UMRA` 0x00000002
- #define `SPR_PCMR_CISM` 0x00000004
- #define `SPR_PCMR_CIUM` 0x00000008
- #define `SPR_PCMR_LA` 0x00000010
- #define `SPR_PCMR_SA` 0x00000020
- #define `SPR_PCMR_IF` 0x00000040
- #define `SPR_PCMR_DCM` 0x00000080
- #define `SPR_PCMR_ICM` 0x00000100
- #define `SPR_PCMR_IFS` 0x00000200
- #define `SPR_PCMR_L SUS` 0x00000400
- #define `SPR_PCMR_BS` 0x00000800
- #define `SPR_PCMR_DTLBM` 0x00001000
- #define `SPR_PCMR_ITLBM` 0x00002000
- #define `SPR_PCMR_DDS` 0x00004000
- #define `SPR_PCMR_WPE` 0x03ff8000
- #define `SPR_PMR_SDF` 0x0000000f
- #define `SPR_PMR_DME` 0x00000010
- #define `SPR_PMR_SME` 0x00000020
- #define `SPR_PMR_DCGE` 0x00000040
- #define `SPR_PMR_SUME` 0x00000080
- #define `SPR_PICMR_IUM` 0xffffffc
- #define `SPR_PICPR_IPRIO` 0xffffffc
- #define `SPR_PICSR_IS` 0xfffffff
- #define `SPR_TTCR_PERIOD` 0x0fffffff
- #define `SPR_TTMR_PERIOD` `SPR_TTCR_PERIOD`
- #define `SPR_TTMR_IP` 0x10000000
- #define `SPR_TTMR_IE` 0x20000000
- #define `SPR_TTMR_RT` 0x40000000
- #define `SPR_TTMR_SR` 0x80000000
- #define `SPR_TTMR_CR` 0xc0000000

- #define `SPR_TTMR_M` 0xc0000000
- #define `NOP_NOP` 0x0000
- #define `NOP_EXIT` 0x0001
- #define `NOP_REPORT` 0x0002
- #define `NOP_PRINTF` 0x0003
- #define `NOP_PUTC` 0x0004
- #define `NOP_CNT_RESET` 0x0005
- #define `NOP_REPORT_FIRST` 0x0400
- #define `NOP_REPORT_LAST` 0x03ff

6.25.1 Define Documentation

6.25.1.1 #define MATCHPOINTS_TO_NDP(n)

Value:

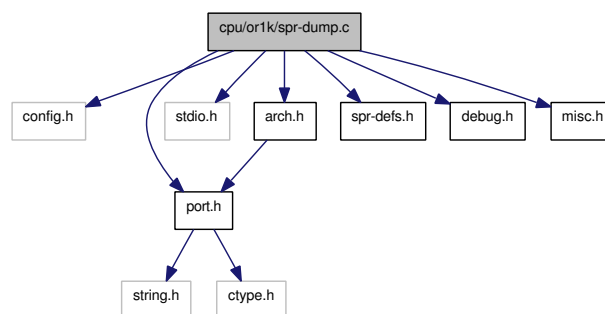
```
(1 == n ? SPR_DCFGR_NDP1 : \  
2 == n ? SPR_DCFGR_NDP2 : \  
3 == n ? SPR_DCFGR_NDP3 : \  
4 == n ? SPR_DCFGR_NDP4 : \  
5 == n ? SPR_DCFGR_NDP5 : \  
6 == n ? SPR_DCFGR_NDP6 : \  
7 == n ? SPR_DCFGR_NDP7 : SPR_DCFGR_NDP8)
```


- 6.25.1.2 `#define MAX_GRPS (32)`
- 6.25.1.3 `#define MAX_MATCHPOINTS 8`
- 6.25.1.4 `#define MAX_SPRS (0x10000)`
- 6.25.1.5 `#define MAX_SPRS_PER_GRP (1 << MAX_SPRS_PER_GRP_BITS)`
- 6.25.1.6 `#define MAX_SPRS_PER_GRP_BITS (11)`
- 6.25.1.7 `#define MAX_WATCHPOINTS (MAX_MATCHPOINTS + 2)`
- 6.25.1.8 `#define NOP_CNT_RESET 0x0005`
- 6.25.1.9 `#define NOP_EXIT 0x0001`
- 6.25.1.10 `#define NOP_NOP 0x0000`
- 6.25.1.11 `#define NOP_PRINTF 0x0003`
- 6.25.1.12 `#define NOP_PUTC 0x0004`
- 6.25.1.13 `#define NOP_REPORT 0x0002`
- 6.25.1.14 `#define NOP_REPORT_FIRST 0x0400`
- 6.25.1.15 `#define NOP_REPORT_LAST 0x03ff`
- 6.25.1.16 `#define SPR_CPUCFGR (SPRGROUP_SYS + 2)`
- 6.25.1.17 `#define SPR_CPUCFGR_CGF 0x00000010`
- 6.25.1.18 `#define SPR_CPUCFGR_NSGF 0x0000000f`
- 6.25.1.19 `#define SPR_CPUCFGR_OB32S 0x00000020`
- 6.25.1.20 `#define SPR_CPUCFGR_OB64S 0x00000040`
- 6.25.1.21 `#define SPR_CPUCFGR_OF32S 0x00000080`
- 6.25.1.22 `#define SPR_CPUCFGR_OF64S 0x00000100`
- 6.25.1.23 `#define SPR_CPUCFGR_OV64S 0x00000200`
- 6.25.1.24 `#define SPR_CPUCFGR_RES 0xffffc00`
- 6.25.1.25 `#define SPR_DCBFR (SPRGROUP_DC + 2)`
- 6.25.1.26 `#define SPR_DCBIR (SPRGROUP_DC + 3)`
- 6.25.1.27 `#define SPR_DCBLR (SPRGROUP_DC + 5)`
- 6.25.1.28 `#define SPR_DCBPR (SPRGROUP_DC + 1)`
- 6.25.1.29 `#define SPR_DCBWR (SPRGROUP_DC + 4)`
- 6.25.1.30 `#define SPR_DCCFGR (SPRGROUP_SYS + 5)`
- 6.25.1.31 `#define SPR_DCCFGR_CBFRI 0x00002000`
- 6.25.1.32 `#define SPR_DCCFGR_CBIRI 0x00000400`

6.26 cpu/or1k/spr-dump.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdio.h>
#include "arch.h"
#include "spr-defs.h"
#include "debug.h"
#include "misc.h"
```

Include dependency graph for spr-dump.c:



Data Structures

- struct [spr_bit_def](#)
- struct [spr_def](#)

Functions

- char * [dump_spr](#) (uint16_t spr, uorreg_t spr_val)

Variables

- static struct [spr_bit_def](#) spr_one_val []
- static struct [spr_bit_def](#) spr_sr []
- static struct [spr_bit_def](#) spr_vr []
- static struct [spr_bit_def](#) spr_upr []
- static struct [spr_bit_def](#) spr_cpucfgr []
- static struct [spr_bit_def](#) spr_dmmucfgr []
- static struct [spr_bit_def](#) spr_immucfgr []
- static struct [spr_bit_def](#) spr_dccfgr []
- static struct [spr_bit_def](#) spr_iccfgr []
- static struct [spr_bit_def](#) spr_defgr []
- static struct [spr_def](#) spr_sys_group []
- static struct [spr_bit_def](#) spr_dmmucr []
- static struct [spr_bit_def](#) spr_dtlbmr []

- static struct `spr_bit_def spr_dtlbtr []`
- static struct `spr_def spr_dmmu_group []`
- static struct `spr_bit_def spr_immucr []`
- static struct `spr_bit_def spr_itlbmr []`
- static struct `spr_bit_def spr_itlbtr []`
- static struct `spr_def spr_immu_group []`
- static struct `spr_bit_def spr_dccr []`
- static struct `spr_def spr_dc_group []`
- static struct `spr_bit_def spr_iccr []`
- static struct `spr_def spr_ic_group []`
- static struct `spr_def spr_mac_group []`
- static struct `spr_bit_def spr_dmr1 []`
- static struct `spr_bit_def spr_dmr2 []`
- static struct `spr_bit_def spr_dwcr []`
- static struct `spr_bit_def spr_dsr []`
- static struct `spr_bit_def spr_drr []`
- static struct `spr_def spr_d_group []`
- static struct `spr_bit_def spr_pcmr []`
- static struct `spr_def spr_pc_group []`
- static struct `spr_bit_def spr_pmr []`
- static struct `spr_def spr_pm_group []`
- static struct `spr_bit_def spr_picmr []`
- static struct `spr_def spr_pic_group []`
- static struct `spr_bit_def spr_tmr []`
- static struct `spr_def spr_tt_group []`
- static struct `spr_def * spr_groups []`
- static char `ret_spr [1000]`

6.26.1 Function Documentation

6.26.1.1 `char* dump_spr (uint16_t spr, uorreg_t spr_val)`

Here is the call graph for this function:



6.26.2 Variable Documentation

6.26.2.1 `char ret_spr[1000]` [static]

6.26.2.2 `struct spr_bit_def spr_cpucfgr[]` [static]

Initial value:

```

{
{ "CPUCFGR_OB32S", SPR_CPUCFGR_OB32S },
{ "CPUCFGR_OB64S", SPR_CPUCFGR_OB64S },
{ "CPUCFGR_OF32S", SPR_CPUCFGR_OF32S },
{ "CPUCFGR_OF64S", SPR_CPUCFGR_OF64S },
{ "CPUCFGR_OV64S", SPR_CPUCFGR_OV64S },
{ NULL, 0 } }
  
```

6.26.2.3 struct spr_def spr_d_group[] [static]**Initial value:**

```
{
    { 0x10, 0x10, "SPR_DMR1", spr_dmr1 },
    { 0x11, 0x11, "SPR_DMR2", spr_dmr2 },
    { 0x12, 0x12, "SPR_DWCR0", spr_dwcr },
    { 0x13, 0x13, "SPR_DWCR1", spr_dwcr },
    { 0x14, 0x14, "SPR_DSR", spr_dsr },
    { 0x15, 0x15, "SPR_DRR", spr_drr },
    { -1, -1, NULL, NULL } }
```

6.26.2.4 struct spr_def spr_dc_group[] [static]**Initial value:**

```
{
    { 0x000, 0x000, "SPR_DCCR", spr_dccr },
    { 0x001, 0x001, "SPR_DCBPR", spr_one_val },
    { 0x002, 0x002, "SPR_DCBFR", spr_one_val },
    { 0x003, 0x003, "SPR_DCBIR", spr_one_val },
    { 0x004, 0x004, "SPR_DCBWR", spr_one_val },
    { 0x005, 0x005, "SPR_DCBLR", spr_one_val },
    { 0x200, 0x3ff, "SPR_DCR way 0 set %i", spr_one_val },
    { 0x400, 0x5ff, "SPR_DCR way 1 set %i", spr_one_val },
    { 0x600, 0x7ff, "SPR_DCR way 2 set %i", spr_one_val },
    { 0x800, 0x9ff, "SPR_DCR way 3 set %i", spr_one_val },
    { -1, -1, NULL, NULL } }
```

6.26.2.5 struct spr_bit_def spr_dccfgr[] [static]**Initial value:**

```
{
    { "DCCFGR_NCW", SPR_DCCFGR_NCW },
    { "DCCFGR_NCS", SPR_DCCFGR_NCS },
    { "DCCFGR_CBS", SPR_DCCFGR_CBS },
    { "DCCFGR_CWS", SPR_DCCFGR_CWS },
    { "DCCFGR_CCRI", SPR_DCCFGR_CCRI },
    { "DCCFGR_CBIRI", SPR_DCCFGR_CBIRI },
    { "DCCFGR_CBPRI", SPR_DCCFGR_CBPRI },
    { "DCCFGR_CBLRI", SPR_DCCFGR_CBLRI },
    { "DCCFGR_CBFRI", SPR_DCCFGR_CBFRI },
    { "DCCFGR_CBWBRI", SPR_DCCFGR_CBWBRI },
    { NULL, 0 } }
```

6.26.2.6 struct spr_bit_def spr_dccr[] [static]**Initial value:**

```
{
    { "DCCR_EW", SPR_DCCR_EW },
    { NULL, 0 } }
```

6.26.2.7 struct spr_bit_def spr_dcfgr[] [static]**Initial value:**

```
{
{ "DCFGR_NDP", SPR_DCFGR_NDP },
{ "DCFGR_WPCI", SPR_DCFGR_WPCI },
{ NULL, 0 } }
```

6.26.2.8 struct spr_def spr_dmmu_group[] [static]**Initial value:**

```
{
{ 0, 0, "SPR_DMMUCR", spr_dmmucr },
{ 0x200, 0x27f, "SPR_DTLBMR way 0 set %i", spr_dtlbmr },
{ 0x280, 0x2ff, "SPR_DTLBTR way 0 set %i", spr_dtlbtr },
{ 0x300, 0x37f, "SPR_DTLBMR way 1 set %i", spr_dtlbmr },
{ 0x380, 0x3ff, "SPR_DTLBTR way 1 set %i", spr_dtlbtr },
{ 0x400, 0x47f, "SPR_DTLBMR way 2 set %i", spr_dtlbmr },
{ 0x480, 0x4ff, "SPR_DTLBTR way 2 set %i", spr_dtlbtr },
{ 0x500, 0x57f, "SPR_DTLBMR way 3 set %i", spr_dtlbmr },
{ 0x580, 0x5ff, "SPR_DTLBTR way 3 set %i", spr_dtlbtr },
{ -1, -1, NULL, NULL } }
```

6.26.2.9 struct spr_bit_def spr_dmmucfgr[] [static]**Initial value:**

```
{
{ "DMMUCFGR_NTW", SPR_DMMUCFGR_NTW },
{ "DMMUCFGR_NTS", SPR_DMMUCFGR_NTS },
{ "DMMUCFGR_NAE", SPR_DMMUCFGR_NAE },
{ "DMMUCFGR_CRI", SPR_DMMUCFGR_CRI },
{ "DMMUCFGR_PRI", SPR_DMMUCFGR_PRI },
{ "DMMUCFGR_TEIRI", SPR_DMMUCFGR_TEIRI },
{ "DMMUCFGR_HTR", SPR_DMMUCFGR_HTR },
{ NULL, 0 } }
```

6.26.2.10 struct spr_bit_def spr_dmmucr[] [static]**Initial value:**

```
{
{ "DMMUCR_P2S", SPR_DMMUCR_P2S },
{ "DMMUCR_P1S", SPR_DMMUCR_P1S },
{ "DMMUCR_VADDR_WIDTH", SPR_DMMUCR_VADDR_WIDTH },
{ "DMMUCR_PADDR_WIDTH", SPR_DMMUCR_PADDR_WIDTH },
{ NULL, 0 } }
```

6.26.2.11 struct spr_bit_def spr_dmr1[] [static]**Initial value:**

```

{
{ "DMR1_CW0",  SPR_DMR1_CW0  },
{ "DMR1_CW1",  SPR_DMR1_CW1  },
{ "DMR1_CW2",  SPR_DMR1_CW2  },
{ "DMR1_CW3",  SPR_DMR1_CW3  },
{ "DMR1_CW4",  SPR_DMR1_CW4  },
{ "DMR1_CW5",  SPR_DMR1_CW5  },
{ "DMR1_CW6",  SPR_DMR1_CW6  },
{ "DMR1_CW7",  SPR_DMR1_CW7  },
{ "DMR1_CW8",  SPR_DMR1_CW8  },
{ "DMR1_CW9",  SPR_DMR1_CW9  },
{ "DMR1_RES1", SPR_DMR1_RES1 },
{ "DMR1_ST",   SPR_DMR1_ST   },
{ "DMR1_BT",   SPR_DMR1_BT   },
{ "DMR1_RES2", SPR_DMR1_RES2 },
{ NULL,       0              } }

```

6.26.2.12 struct spr_bit_def spr_dmr2[] [static]**Initial value:**

```

{
{ "DMR2_WCE0", SPR_DMR2_WCE0 },
{ "DMR2_WCE1", SPR_DMR2_WCE1 },
{ "DMR2_AWTC", SPR_DMR2_AWTC },
{ "DMR2_WGB",  SPR_DMR2_WGB  },
{ "DMR2_WBS",  SPR_DMR2_WBS  },
{ NULL,       0              } }

```

6.26.2.13 struct spr_bit_def spr_drr[] [static]**Initial value:**

```

{
{ "DRR_RSTE", SPR_DRR_RSTE },
{ "DRR_BUSE", SPR_DRR_BUSEE },
{ "DRR_DPFE", SPR_DRR_DPFE },
{ "DRR_IPFE", SPR_DRR_IPFE },
{ "DRR_TTE",  SPR_DRR_TTE  },
{ "DRR_AE",   SPR_DRR_AE   },
{ "DRR_IIE",  SPR_DRR_IIE  },
{ "DRR_IE",   SPR_DRR_IE   },
{ "DRR_DME",  SPR_DRR_DME  },
{ "DRR_IME",  SPR_DRR_IME  },
{ "DRR_RE",   SPR_DRR_RE   },
{ "DRR_SCE",  SPR_DRR_SCE  },
{ "DRR_TE",   SPR_DRR_TE   },
{ NULL, 0 } }

```

6.26.2.14 struct spr_bit_def spr_dsr[] [static]**Initial value:**


```

{
{ "DSR_RSTE", SPR_DSR_RSTE },
{ "DSR_BUSE", SPR_DSR_BUSEE },
{ "DSR_DPFE", SPR_DSR_DPFE },
{ "DSR_IPFE", SPR_DSR_IPFE },
{ "DSR_TTE", SPR_DSR_TTE },
{ "DSR_AE", SPR_DSR_AE },
{ "DSR_IIE", SPR_DSR_IIE },
{ "DSR_IE", SPR_DSR_IE },
{ "DSR_DME", SPR_DSR_DME },
{ "DSR_IME", SPR_DSR_IME },
{ "DSR_RE", SPR_DSR_RE },
{ "DSR_SCE", SPR_DSR_SCE },
{ "DSR_SSE", SPR_DSR_SSE },
{ "DSR_TE", SPR_DSR_TE },
{ NULL, 0 } }

```

6.26.2.15 struct spr_bit_def spr_dtlbmr[] [static]

Initial value:

```

{
{ "DTLBMR_V", SPR_DTLBMR_V },
{ "DTLBMR_PL1", SPR_DTLBMR_PL1 },
{ "DTLBMR_CID", SPR_DTLBMR_CID },
{ "DTLBMR_LRU", SPR_DTLBMR_LRU },
{ "DTLBMR_VPN", SPR_DTLBMR_VPN },
{ NULL, 0 } }

```

6.26.2.16 struct spr_bit_def spr_dtlbtr[] [static]

Initial value:

```

{
{ "DTLBTR_CC", SPR_DTLBTR_CC },
{ "DTLBTR_CI", SPR_DTLBTR_CI },
{ "DTLBTR_WBC", SPR_DTLBTR_WBC },
{ "DTLBTR_WOM", SPR_DTLBTR_WOM },
{ "DTLBTR_A", SPR_DTLBTR_A },
{ "DTLBTR_D", SPR_DTLBTR_D },
{ "DTLBTR_URE", SPR_DTLBTR_URE },
{ "DTLBTR_UWE", SPR_DTLBTR_UWE },
{ "DTLBTR_SRE", SPR_DTLBTR_SRE },
{ "DTLBTR_SWE", SPR_DTLBTR_SWE },
{ "DTLBTR_PPN", SPR_DTLBTR_PPN },
{ NULL, 0 } }

```

6.26.2.17 struct spr_bit_def spr_dwcr[] [static]

Initial value:

```

{
{ "DWCR_COUNT", SPR_DWCR_COUNT },
{ "DWCR_MATCH", SPR_DWCR_MATCH },
{ NULL, 0 } }

```

6.26.2.18 struct spr_def* spr_groups[] [static]**Initial value:**

```
{
  spr_sys_group,
  spr_dmmu_group,
  spr_immu_group,
  spr_dc_group,
  spr_ic_group,
  spr_mac_group,
  spr_d_group,
  spr_pc_group,
  spr_pm_group,
  spr_pic_group,
  spr_tt_group }
```

6.26.2.19 struct spr_def spr_ic_group[] [static]**Initial value:**

```
{
  { 0x000, 0x000, "SPR_ICCR", spr_iccr },
  { 0x001, 0x001, "SPR_ICBPR", spr_one_val },
  { 0x002, 0x002, "SPR_ICBFR", spr_one_val },
  { 0x003, 0x003, "SPR_ICBIR", spr_one_val },
  { 0x200, 0x3ff, "SPR_ICR way 0 set %i", spr_one_val },
  { 0x400, 0x5ff, "SPR_ICR way 1 set %i", spr_one_val },
  { 0x600, 0x7ff, "SPR_ICR way 2 set %i", spr_one_val },
  { 0x800, 0x9ff, "SPR_ICR way 3 set %i", spr_one_val },
  { -1, -1, NULL, NULL } }
```

6.26.2.20 struct spr_bit_def spr_iccfgr[] [static]**Initial value:**

```
{
  { "ICCFGR_NCW", SPR_ICCFGR_NCW },
  { "ICCFGR_NCS", SPR_ICCFGR_NCS },
  { "ICCFGR_CBS", SPR_ICCFGR_CBS },
  { "ICCFGR_CCRI", SPR_ICCFGR_CCRI },
  { "ICCFGR_CBIRI", SPR_ICCFGR_CBIRI },
  { "ICCFGR_CBPRI", SPR_ICCFGR_CBPRI },
  { "ICCFGR_CBLRI", SPR_ICCFGR_CBLRI },
  { NULL, 0 } }
```

6.26.2.21 struct spr_bit_def spr_iccr[] [static]**Initial value:**

```
{
  { "ICCR_EW", SPR_ICCR_EW },
  { NULL, 0 } }
```

6.26.2.22 struct spr_def spr_immu_group[] [static]**Initial value:**

```
{
  { 0, 0, "SPR_IMMUCR", spr_immucr },
  { 0x200, 0x27f, "SPR_ITLBMR way 0 set %i", spr_itlbmr },
  { 0x280, 0x2ff, "SPR_ITLBTR way 0 set %i", spr_itlbtr },
  { 0x300, 0x37f, "SPR_ITLBMR way 1 set %i", spr_itlbmr },
  { 0x380, 0x3ff, "SPR_ITLBTR way 1 set %i", spr_itlbtr },
  { 0x400, 0x47f, "SPR_ITLBMR way 2 set %i", spr_itlbmr },
  { 0x480, 0x4ff, "SPR_ITLBTR way 2 set %i", spr_itlbtr },
  { 0x500, 0x57f, "SPR_ITLBMR way 3 set %i", spr_itlbmr },
  { 0x580, 0x5ff, "SPR_ITLBTR way 3 set %i", spr_itlbtr },
  { -1, -1, NULL, NULL } }
```

6.26.2.23 struct spr_bit_def spr_immucfgr[] [static]**Initial value:**

```
{
  { "IMMUCFGR_NTW", SPR_IMMUCFGR_NTW },
  { "IMMUCFGR_NTS", SPR_IMMUCFGR_NTS },
  { "IMMUCFGR_NAE", SPR_IMMUCFGR_NAE },
  { "IMMUCFGR_CRI", SPR_IMMUCFGR_CRI },
  { "IMMUCFGR_PRI", SPR_IMMUCFGR_PRI },
  { "IMMUCFGR_TEIRI", SPR_IMMUCFGR_TEIRI },
  { "IMMUCFGR_HTR", SPR_IMMUCFGR_HTR },
  { NULL, 0 } }
```

6.26.2.24 struct spr_bit_def spr_immucr[] [static]**Initial value:**

```
{
  { "IMMUCR_P2S", SPR_IMMUCR_P2S },
  { "IMMUCR_P1S", SPR_IMMUCR_P1S },
  { "IMMUCR_VADDR_WIDTH", SPR_IMMUCR_VADDR_WIDTH },
  { "IMMUCR_PADDR_WIDTH", SPR_IMMUCR_PADDR_WIDTH },
  { NULL, 0 } }
```

6.26.2.25 struct spr_bit_def spr_itlbmr[] [static]**Initial value:**

```
{
  { "ITLBMR_V", SPR_ITLBMR_V },
  { "ITLBMR_PL1", SPR_ITLBMR_PL1 },
  { "ITLBMR_CID", SPR_ITLBMR_CID },
  { "ITLBMR_LRU", SPR_ITLBMR_LRU },
  { "ITLBMR_VPN", SPR_ITLBMR_VPN },
  { NULL, 0 } }
```

6.26.2.26 struct spr_bit_def spr_itlbtr[] [static]**Initial value:**

```
{
{ "ITLBTR_CC", SPR_ITLBTR_CC },
{ "ITLBTR_CI", SPR_ITLBTR_CI },
{ "ITLBTR_WBC", SPR_ITLBTR_WBC },
{ "ITLBTR_WOM", SPR_ITLBTR_WOM },
{ "ITLBTR_A", SPR_ITLBTR_A },
{ "ITLBTR_D", SPR_ITLBTR_D },
{ "ITLBTR_URE", SPR_ITLBTR_SXE },
{ "ITLBTR_UWE", SPR_ITLBTR_UXE },
{ "ITLBTR_PPN", SPR_ITLBTR_PPN },
{ NULL, 0 } }
```

6.26.2.27 struct spr_def spr_mac_group[] [static]**Initial value:**

```
{
{ 0x1, 0x1, "SPR_MACLO", spr_one_val },
{ 0x2, 0x2, "SPR_MACHI", spr_one_val },
{ -1, -1, NULL, NULL } }
```

6.26.2.28 struct spr_bit_def spr_one_val[] [static]**Initial value:**

```
{
{ "", 0xffffffff },
{ NULL, 0 } }
```

6.26.2.29 struct spr_def spr_pc_group[] [static]**Initial value:**

```
{
{ 0x00, 0x07, "PCCR", spr_one_val },
{ 0x08, 0x0f, "PCMR", spr_pcmr },
{ -1, -1, NULL, NULL } }
```

6.26.2.30 struct spr_bit_def spr_pcmr[] [static]**Initial value:**

```
{
{ "PCMR_CP", SPR_PCMR_CP },
{ "PCMR_UMRA", SPR_PCMR_UMRA },
{ "PCMR_CISM", SPR_PCMR_CISM },
{ "PCMR_CIUUM", SPR_PCMR_CIUUM },
{ "PCMR_LA", SPR_PCMR_LA },
```

```

{ "PCMR_SA", SPR_PCMR_SA },
{ "PCMR_IF", SPR_PCMR_IF },
{ "PCMR_DCM", SPR_PCMR_DCM },
{ "PCMR_ICM", SPR_PCMR_ICM },
{ "PCMR_IFS", SPR_PCMR_IFS },
{ "PCMR_LSUS", SPR_PCMR_LSUS },
{ "PCMR_BS", SPR_PCMR_BS },
{ "PCMR_DTLBM", SPR_PCMR_DTLBM },
{ "PCMR_ITLBM", SPR_PCMR_ITLBM },
{ "PCMR_DDS", SPR_PCMR_DDS },
{ "PCMR_WPE", SPR_PCMR_WPE },
{ NULL, 0 } }

```

6.26.2.31 struct spr_def spr_pic_group[] [static]

Initial value:

```

{
{ 0, 0, "PICMR", spr_picmr },
{ 2, 2, "PICSR", spr_one_val },
{ -1, -1, NULL, NULL } }

```

6.26.2.32 struct spr_bit_def spr_picmr[] [static]

Initial value:

```

{
{ "PICMR_IUM", SPR_PICMR_IUM },
{ NULL, 0 } }

```

6.26.2.33 struct spr_def spr_pm_group[] [static]

Initial value:

```

{
{ 0x0, 0x0, "SPR_PMR", spr_pmr },
{ -1, -1, NULL, NULL } }

```

6.26.2.34 struct spr_bit_def spr_pmr[] [static]

Initial value:

```

{
{ "PMR_SDF", SPR_PMR_SDF },
{ "PMR_DME", SPR_PMR_DME },
{ "PMR_SME", SPR_PMR_SME },
{ "PMR_DCGE", SPR_PMR_DCGE },
{ "PMR_SUME", SPR_PMR_SUME },
{ NULL, 0 } }

```

6.26.2.35 struct spr_bit_def spr_sr[] [static]**Initial value:**

```

{
{ "SR_SM",      SPR_SR_SM      },
{ "SR_TEE",    SPR_SR_TEE     },
{ "SR_IEE",    SPR_SR_IEE     },
{ "SR_DCE",    SPR_SR_DCE     },
{ "SR_ICE",    SPR_SR_ICE     },
{ "SR_DME",    SPR_SR_DME     },
{ "SR_IME",    SPR_SR_IME     },
{ "SR_LEE",    SPR_SR_IME     },
{ "SR_CE",     SPR_SR_CE      },
{ "SR_F",      SPR_SR_F       },
{ "SR_CY",     SPR_SR_CY      },
{ "SR_OV",     SPR_SR_OV      },
{ "SR_OVE",    SPR_SR_OVE     },
{ "SR_DSX",    SPR_SR_DSX     },
{ "SR_EPH",    SPR_SR_EPH     },
{ "SR_FO",     SPR_SR_FO      },
{ "SR_SUMRA",  SPR_SR_SUMRA   },
{ "SR_RES",    SPR_SR_RES     },
{ "SR_CID",    SPR_SR_CID     },
{ NULL,        0              } }

```

6.26.2.36 struct spr_def spr_sys_group[] [static]**Initial value:**

```

{
{ 0x000, 0x000, "SPR_VR",      spr_vr      },
{ 0x001, 0x001, "SPR_UPR",    spr_upr    },
{ 0x002, 0x002, "SPR_CPUCFGR", spr_cpucfgr },
{ 0x003, 0x003, "SPR_DMMUCFGR", spr_dmmucfgr },
{ 0x004, 0x004, "SPR_IMMUCFGR", spr_immucfgr },
{ 0x005, 0x005, "SPR_DCCFGR",  spr_dccfgr },
{ 0x006, 0x006, "SPR_ICCFGR",  spr_iccfgr },
{ 0x007, 0x007, "SPR_DCFGR",   spr_dcfgr },
{ 0x010, 0x010, "SPR_NPC",     spr_one_val },
{ 0x011, 0x011, "SPR_SR",     spr_sr     },
{ 0x012, 0x012, "SPR_PPC",    spr_one_val },
{ 0x020, 0x02f, "SPR_EPCR(%i)", spr_one_val },
{ 0x030, 0x03f, "SPR_EEAR(%i)", spr_one_val },
{ 0x040, 0x04f, "SPR_ESR(%i)", spr_sr     },
{ 0x400, 0x41f, "GPR(%i)",    spr_one_val },
{ -1,    -1,    NULL,         NULL        } }

```

6.26.2.37 struct spr_def spr_tt_group[] [static]**Initial value:**

```

{
{ 0, 0, "TTMR", spr_ttmr },
{ 0, 0, "TTCR", spr_one_val },
{ -1, -1, NULL, NULL } }

```

6.26.2.38 struct spr_bit_def spr_ttmr[] [static]**Initial value:**

```
{
{ "TTMR_PERIOD", SPR_TTMR_PERIOD },
{ "TTMR_IP", SPR_TTMR_IP },
{ "TTMR_IE", SPR_TTMR_IE },
{ "TTMR_M", SPR_TTMR_M },
{ NULL, 0 } }
```

6.26.2.39 struct spr_bit_def spr_upr[] [static]**Initial value:**

```
{
{ "UPR_UP", SPR_UPR_UP },
{ "UPR_DCP", SPR_UPR_DCP },
{ "UPR_ICP", SPR_UPR_ICP },
{ "UPR_DMP", SPR_UPR_DMP },
{ "UPR_IMP", SPR_UPR_IMP },
{ "UPR_MP", SPR_UPR_MP },
{ "UPR_DUP", SPR_UPR_DUP },
{ "UPR_PCUP", SPR_UPR_PCUP },
{ "UPR_PMP", SPR_UPR_PMP },
{ "UPR_PICP", SPR_UPR_PICP },
{ "UPR_TTP", SPR_UPR_TTP },
{ "UPR_RES", SPR_UPR_RES },
{ "UPR_CUP", SPR_UPR_CUP },
{ NULL, 0 } }
```

6.26.2.40 struct spr_bit_def spr_vr[] [static]**Initial value:**

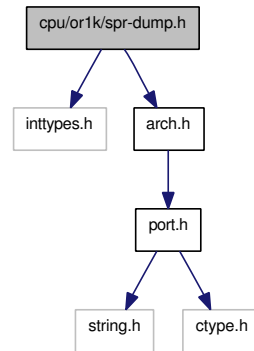
```
{
{ "VR_VER", SPR_VR_VER },
{ "VR_REV", SPR_VR_REV },
{ NULL, 0 } }
```

6.27 cpu/or1k/spr-dump.h File Reference

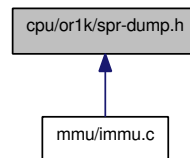
```
#include <inttypes.h>
```

```
#include "arch.h"
```

Include dependency graph for spr-dump.h:



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



Functions

- `char * dump_spr (uint16_t spr, uorreg_t spr_val)`

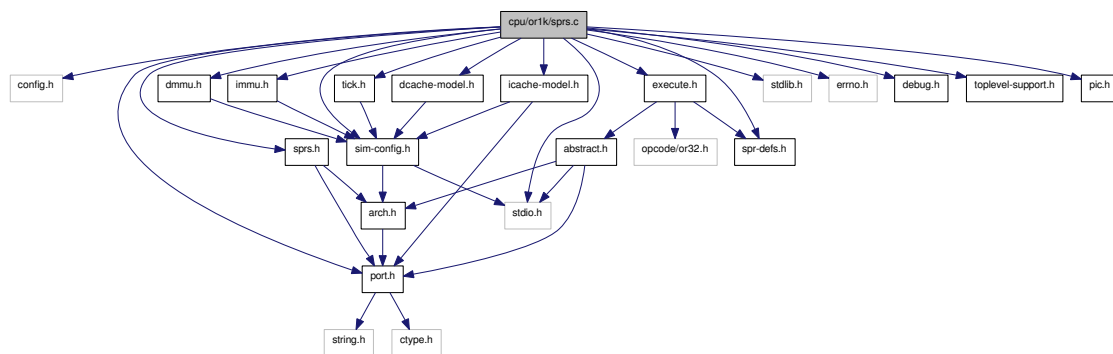
6.27.1 Function Documentation

6.27.1.1 `char* dump_spr (uint16_t spr, uorreg_t spr_val)`

6.28 cpu/or1k/sprs.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include "sprs.h"
#include "sim-config.h"
#include "debug.h"
#include "execute.h"
#include "spr-defs.h"
#include "tick.h"
#include "dcache-model.h"
#include "icache-model.h"
#include "dmmu.h"
#include "immu.h"
#include "toplevel-support.h"
#include "pic.h"
```

Include dependency graph for sprs.c:



Functions

- `DECLARE_DEBUG_CHANNEL` (`immu`)
- `void mtspr` (`uint16_t regno`, `const uorreg_t value`)
- `uorreg_t mfspr` (`const uint16_t regno`)
- `void sprs_status` (`void`)

Variables

- `static int audio_cnt = 0`

- static FILE * fo = 0

6.28.1 Function Documentation

6.28.1.1 DECLARE_DEBUG_CHANNEL (immu)

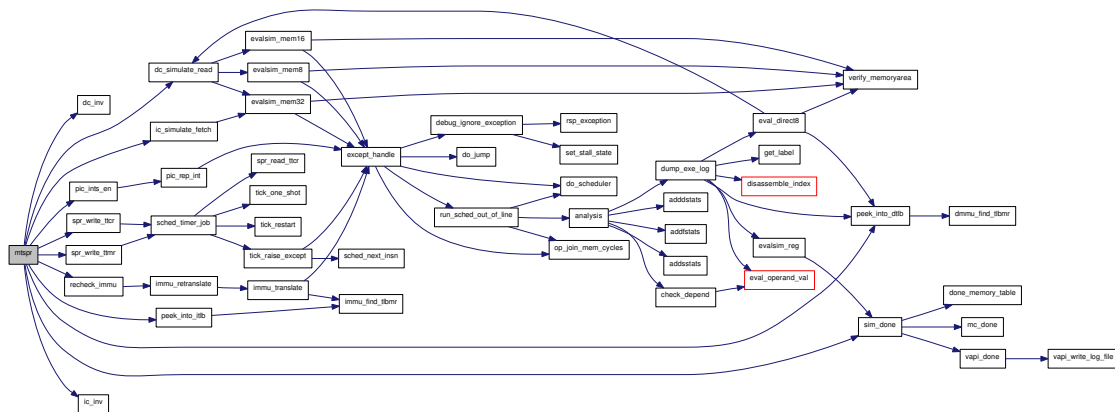
6.28.1.2 uorreg_t mfspr (const uint16_t regno)

Here is the call graph for this function:



6.28.1.3 void mtspr (uint16_t regno, const uorreg_t value)

Here is the call graph for this function:



6.28.1.4 void sprs_status (void)

6.28.2 Variable Documentation

6.28.2.1 int audio_cnt = 0 [static]

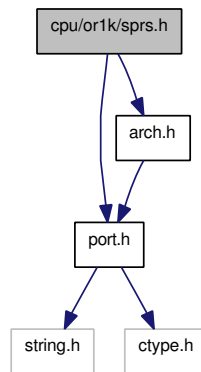
6.28.2.2 FILE* fo = 0 [static]

6.29 cpu/or1k/sprs.h File Reference

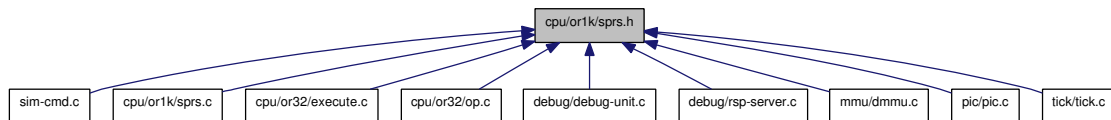
```
#include "port.h"
```

```
#include "arch.h"
```

Include dependency graph for sprs.h:



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



Functions

- void `mtspr` (uint16_t regno, const `uorreg_t` value)
- `uorreg_t` `mfspr` (const uint16_t regno)
- void `sprs_status` ()
- char * `dump_spr` (uint16_t spr, `uorreg_t` spr_val)

6.29.1 Function Documentation

6.29.1.1 char* `dump_spr` (uint16_t *spr*, `uorreg_t` *spr_val*)

Here is the call graph for this function:



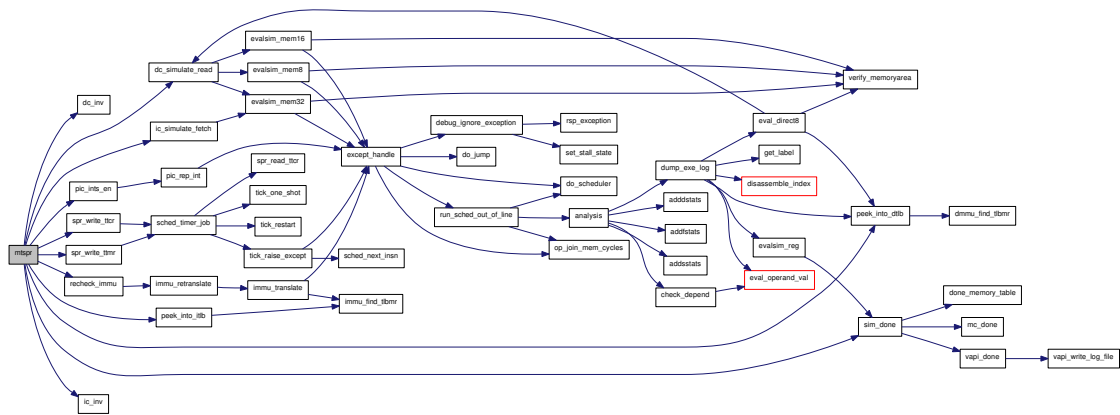
6.29.1.2 uorreg_t mfspr (const uint16_t regno)

Here is the call graph for this function:



6.29.1.3 void mtspr (uint16_t regno, const uorreg_t value)

Here is the call graph for this function:



6.29.1.4 void sprs_status ()

6.30 cpu/or32/common-i386.h File Reference

Functions

- static void `set_pc` (`oraddr_t pc`)
- static `oraddr_t get_pc` (`void`)
- static void `upd_sim_cycles` (`void`)

Variables

- union {
 struct {
 uint32_t `low32`
 uint32_t `high32`
 } `val3232`
 uint64_t `val64`
} `useless_x86`

6.30.1 Function Documentation

6.30.1.1 `static oraddr_t get_pc (void)` [static]

6.30.1.2 `static void set_pc (oraddr_t pc)` [static]

6.30.1.3 `static void upd_sim_cycles (void)` [static]

6.30.2 Variable Documentation

6.30.2.1 `uint32_t high32`

6.30.2.2 `uint32_t low32`

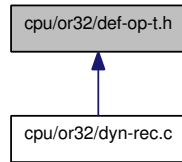
6.30.2.3 `union { ... } useless_x86` [static]

6.30.2.4 `struct { ... } val3232`

6.30.2.5 `uint64_t val64`

6.31 cpu/or32/def-op-t.h File Reference

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



Defines

- #define [GPR_T](#)(op_name, temp)
- #define [DEF_1T_OP](#)(type, name, op_name)
- #define [DEF_2T_OP](#)(type, name, op_name)
- #define [DEF_3T_OP](#)(type, name, op_name)
- #define [DEF_2T_OP_NEQ](#)(type, name, op_name)
- #define [DEF_3T_OP_NEQ](#)(type, name, op_name)
- #define [DEF_GPR_OP](#)(type, name, op_name)

6.31.1 Define Documentation

6.31.1.1 #define DEF_1T_OP(type, name, op_name)

Value:

```
static const type name[NUM_T_REGS] = \
    OP_ROW(op_name)
```

6.31.1.2 #define DEF_2T_OP(type, name, op_name)

Value:

```
static const type name[NUM_T_REGS][NUM_T_REGS] = \
    OP_ROW_COL(op_name)
```

6.31.1.3 #define DEF_2T_OP_NEQ(type, name, op_name)

Value:

```
static const type name[NUM_T_REGS][NUM_T_REGS] = \
    OP_ROW_COL_NEQ(op_name)
```

6.31.1.4 #define DEF_3T_OP(type, name, op_name)**Value:**

```
static const type name[NUM_T_REGS][NUM_T_REGS][NUM_T_REGS] = \
    OP_ROW_COL_3D(op_name)
```

6.31.1.5 #define DEF_3T_OP_NEQ(type, name, op_name)**Value:**

```
static const type name[NUM_T_REGS][NUM_T_REGS][NUM_T_REGS] = \
    OP_ROW_COL_3D_NEQ(op_name)
```

6.31.1.6 #define DEF_GPR_OP(type, name, op_name)**Value:**

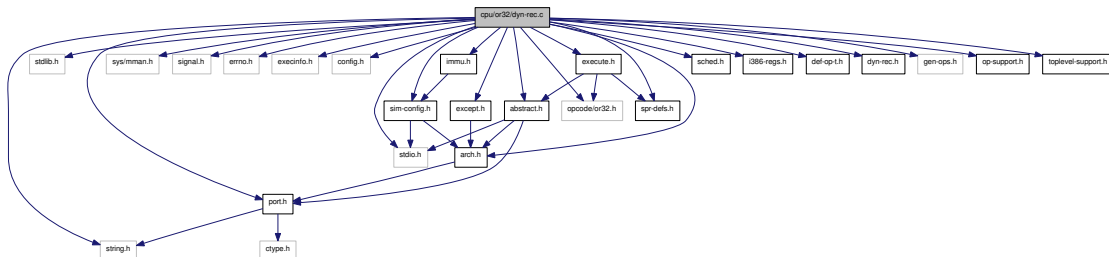
```
static const generic_gen_op name[NUM_T_REGS][32] = \
    GPR_ROW_COL(op_name)
```

6.31.1.7 #define GPR_T(op_name, temp)

6.32 cpu/or32/dyn-rec.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/mman.h>
#include <signal.h>
#include <errno.h>
#include <execinfo.h>
#include "config.h"
#include "port.h"
#include "arch.h"
#include "immu.h"
#include "abstract.h"
#include "opcode/or32.h"
#include "spr-defs.h"
#include "execute.h"
#include "except.h"
#include "sim-config.h"
#include "sched.h"
#include "i386-regs.h"
#include "def-op-t.h"
#include "dyn-rec.h"
#include "gen-ops.h"
#include "op-support.h"
#include "toplevel-support.h"
```

Include dependency graph for dyn-rec.c:



Defines

- #define [RECED_PAGE_ENLARGE_BY](#) 51200

- #define `OPS_ENLARGE_BY` 5
- #define `T_NONE` (-1)
- #define `TFLAG_SRC` 1
- #define `TFLAG_DST` 2
- #define `TFLAG_SAVED` 4
- #define `TFLAG_SOURCED` 8

Typedefs

- typedef void(* `generic_gen_op`)(struct `op_queue` *opq, int end)
- typedef void(* `imm_gen_op`)(struct `op_queue` *opq, int end, `uorreg_t` imm)

Functions

- void `gen_l_invalid` (struct `op_queue` *opq, int `param_t`[3], int delay_slot)
- `DEF_GPR_OP` (`generic_gen_op`, `gen_op_move_gpr_t`, `gen_op_ttg_gpr`)
- `DEF_GPR_OP` (`generic_gen_op`, `gen_op_move_t_gpr`, `gen_op_gtt_gpr`)
- `DEF_1T_OP` (`imm_gen_op`, `calc_insn_ea_table`, `gen_op_calc_insn_ea`)
- void `dyn_ret_stack_prot` (void)
- void `dyn_sigsegv_debug` (int u, `siginfo_t` *siginf, void *dat)
- struct `dyn_page` * `new_dp` (`oraddr_t` page)
- void `dyn_main` (void)
- static void `immu_retranslate` (void *dat)
- void `recheck_immu` (int got_en_dis)
- void `run_sched_out_of_line` (void)
- static void `dirtyfy_page` (struct `dyn_page` *dp)
- void `dyn_checkwrite` (`oraddr_t` addr)
- static void `ship_t_out` (struct `op_queue` *opq, unsigned int t)
- static void `ship_gprs_out_t` (struct `op_queue` *opq)
- static int `find_t` (struct `op_queue` *opq, unsigned int reg)
- void * `enough_host_page` (struct `dyn_page` *dp, void *cur, unsigned int *len, unsigned int amount)
- void `add_to_opq` (struct `op_queue` *opq, int end, int op)
- static void `gen_op_mark_loc` (struct `op_queue` *opq, int end)
- void `add_to_op_params` (struct `op_queue` *opq, int end, unsigned long param)
- void `init_dyn_recomp` (void)
- static void `eval_insn_ops` (struct `op_queue` *opq, `oraddr_t` addr)
- static void `recompile_insn` (struct `op_queue` *opq, int delay_insn)
- void `recompile_page` (struct `dyn_page` *dyn)
- static void `recompile_delay_insn` (struct `op_queue` *opq)
- static int `find_jump_loc` (`oraddr_t` j_ea, struct `op_queue` *opq)
- static void `gen_j_imm` (struct `op_queue` *opq, `oraddr_t` off)
- static void `gen_j_reg` (struct `op_queue` *opq, unsigned int gpr)
- `DEF_1T_OP` (`generic_gen_op`, `clear_t`, `gen_op_clear`)
- `DEF_2T_OP_NEQ` (`generic_gen_op`, `move_t_t`, `gen_op_move`)
- `DEF_1T_OP` (`imm_gen_op`, `mov_t_imm`, `gen_op_imm`)
- `DEF_2T_OP` (`imm_gen_op`, `l_add_imm_t_table`, `gen_op_add_imm`)
- `DEF_3T_OP` (`generic_gen_op`, `l_add_t_table`, `gen_op_add`)
- void `gen_l_add` (struct `op_queue` *opq, int `param_t`[3], int delay_slot)
- `DEF_3T_OP` (`generic_gen_op`, `l_addc_t_table`, `gen_op_addc`)

- void `gen_l_addc` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_2T_OP` (`imm_gen_op`, `l_and_imm_t_table`, `gen_op_and_imm`)
- `DEF_3T_OP_NEQ` (`generic_gen_op`, `l_and_t_table`, `gen_op_and`)
- void `gen_l_and` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_bf` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_bnf` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_3T_OP_NEQ` (`generic_gen_op`, `l_cmov_t_table`, `gen_op_cmov`)
- void `gen_l_cmov` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_cust1` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_cust2` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_cust3` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_cust4` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_cust5` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_cust6` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_cust7` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_cust8` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_1T_OP` (`generic_gen_op`, `check_null_except`, `gen_op_check_null_except`)
- `DEF_1T_OP` (`generic_gen_op`, `check_null_except_delay`, `gen_op_check_null_except_delay`)
- `DEF_3T_OP` (`generic_gen_op`, `l_div_t_table`, `gen_op_div`)
- void `gen_l_div` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_3T_OP` (`generic_gen_op`, `l_divu_t_table`, `gen_op_divu`)
- void `gen_l_divu` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_2T_OP` (`generic_gen_op`, `l_extbs_t_table`, `gen_op_extbs`)
- void `gen_l_extbs` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_2T_OP` (`generic_gen_op`, `l_extbz_t_table`, `gen_op_extbz`)
- void `gen_l_extbz` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_2T_OP` (`generic_gen_op`, `l_exths_t_table`, `gen_op_exths`)
- void `gen_l_exths` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_2T_OP` (`generic_gen_op`, `l_exthz_t_table`, `gen_op_exthz`)
- void `gen_l_exthz` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_extws` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_extwz` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_2T_OP` (`generic_gen_op`, `l_ff1_t_table`, `gen_op_ff1`)
- void `gen_l_ff1` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_j` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_jal` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_jr` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- void `gen_l_jalr` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_1T_OP` (`imm_gen_op`, `l_lbs_imm_t_table`, `gen_op_lbs_imm`)
- `DEF_2T_OP` (`imm_gen_op`, `l_lbs_t_table`, `gen_op_lbs`)
- void `gen_l_lbs` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_1T_OP` (`imm_gen_op`, `l_lbz_imm_t_table`, `gen_op_lbz_imm`)
- `DEF_2T_OP` (`imm_gen_op`, `l_lbz_t_table`, `gen_op_lbz`)
- void `gen_l_lbz` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_1T_OP` (`imm_gen_op`, `l_lhs_imm_t_table`, `gen_op_lhs_imm`)
- `DEF_2T_OP` (`imm_gen_op`, `l_lhs_t_table`, `gen_op_lhs`)
- void `gen_l_lhs` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)
- `DEF_1T_OP` (`imm_gen_op`, `l_lhz_imm_t_table`, `gen_op_lhz_imm`)
- `DEF_2T_OP` (`imm_gen_op`, `l_lhz_t_table`, `gen_op_lhz`)
- void `gen_l_lhz` (struct `op_queue` *opq, int `param_t`[3], int `delay_slot`)

- DEF_1T_OP (imm_gen_op, l_lws_imm_t_table, gen_op_lws_imm)
- DEF_2T_OP (imm_gen_op, l_lws_t_table, gen_op_lws)
- void gen_l_lws (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_lwz_imm_t_table, gen_op_lwz_imm)
- DEF_2T_OP (imm_gen_op, l_lwz_t_table, gen_op_lwz)
- void gen_l_lwz (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_mac_imm_t_table, gen_op_mac_imm)
- DEF_2T_OP (generic_gen_op, l_mac_t_table, gen_op_mac)
- void gen_l_mac (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (generic_gen_op, l_macrc_t_table, gen_op_macrc)
- void gen_l_macrc (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_mfspr_imm_t_table, gen_op_mfspr_imm)
- DEF_2T_OP (imm_gen_op, l_mfspr_t_table, gen_op_mfspr)
- void gen_l_mfspr (struct op_queue *opq, int param_t[3], int delay_slot)
- void gen_l_movhi (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_2T_OP (generic_gen_op, l_msb_t_table, gen_op_msb)
- void gen_l_msb (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_mtspr_clear_t_table, gen_op_mtspr_clear)
- DEF_1T_OP (imm_gen_op, l_mtspr_imm_t_table, gen_op_mtspr_imm)
- DEF_2T_OP (imm_gen_op, l_mtspr_t_table, gen_op_mtspr)
- void gen_l_mtspr (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_2T_OP (imm_gen_op, l_mul_imm_t_table, gen_op_mul_imm)
- DEF_3T_OP (generic_gen_op, l_mul_t_table, gen_op_mul)
- void gen_l_mul (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_3T_OP (generic_gen_op, l_mulu_t_table, gen_op_mulu)
- void gen_l_mulu (struct op_queue *opq, int param_t[3], int delay_slot)
- void gen_l_nop (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_2T_OP (imm_gen_op, l_or_imm_t_table, gen_op_or_imm)
- DEF_3T_OP_NEQ (generic_gen_op, l_or_t_table, gen_op_or)
- void gen_l_or (struct op_queue *opq, int param_t[3], int delay_slot)
- void gen_l_rfe (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_sb_clear_table, gen_op_sb_clear)
- DEF_1T_OP (imm_gen_op, l_sb_imm_t_table, gen_op_sb_imm)
- DEF_2T_OP (imm_gen_op, l_sb_t_table, gen_op_sb)
- void gen_l_sb (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_sh_clear_table, gen_op_sh_clear)
- DEF_1T_OP (imm_gen_op, l_sh_imm_t_table, gen_op_sh_imm)
- DEF_2T_OP (imm_gen_op, l_sh_t_table, gen_op_sh)
- void gen_l_sh (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_sw_clear_table, gen_op_sw_clear)
- DEF_1T_OP (imm_gen_op, l_sw_imm_t_table, gen_op_sw_imm)
- DEF_2T_OP (imm_gen_op, l_sw_t_table, gen_op_sw)
- void gen_l_sw (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (generic_gen_op, l_sfeq_null_t_table, gen_op_sfeq_null)
- DEF_1T_OP (imm_gen_op, l_sfeq_imm_t_table, gen_op_sfeq_imm)
- DEF_2T_OP (generic_gen_op, l_sfeq_t_table, gen_op_sfeq)
- void gen_l_sfeq (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (generic_gen_op, l_sfges_null_t_table, gen_op_sfges_null)
- DEF_1T_OP (generic_gen_op, l_sfles_null_t_table, gen_op_sfles_null)
- DEF_1T_OP (imm_gen_op, l_sfges_imm_t_table, gen_op_sfges_imm)

- DEF_2T_OP (generic_gen_op, l_sfges_t_table, gen_op_sfges)
- void gen_l_sfges (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (generic_gen_op, l_sfgeu_null_t_table, gen_op_sfgeu_null)
- DEF_1T_OP (generic_gen_op, l_sfleu_null_t_table, gen_op_sfleu_null)
- DEF_1T_OP (imm_gen_op, l_sfgeu_imm_t_table, gen_op_sfgeu_imm)
- DEF_2T_OP (generic_gen_op, l_sfgeu_t_table, gen_op_sfgeu)
- void gen_l_sfgeu (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (generic_gen_op, l_sfgts_null_t_table, gen_op_sfgts_null)
- DEF_1T_OP (generic_gen_op, l_sflts_null_t_table, gen_op_sflts_null)
- DEF_1T_OP (imm_gen_op, l_sfgts_imm_t_table, gen_op_sfgts_imm)
- DEF_2T_OP (generic_gen_op, l_sfgts_t_table, gen_op_sfgts)
- void gen_l_sfgts (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (generic_gen_op, l_sfgtu_null_t_table, gen_op_sfgtu_null)
- DEF_1T_OP (generic_gen_op, l_sfltu_null_t_table, gen_op_sfltu_null)
- DEF_1T_OP (imm_gen_op, l_sfgtu_imm_t_table, gen_op_sfgtu_imm)
- DEF_2T_OP (generic_gen_op, l_sfgtu_t_table, gen_op_sfgtu)
- void gen_l_sfgtu (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_sfles_imm_t_table, gen_op_sfles_imm)
- DEF_2T_OP (generic_gen_op, l_sfles_t_table, gen_op_sfles)
- void gen_l_sfles (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_sfleu_imm_t_table, gen_op_sfleu_imm)
- DEF_2T_OP (generic_gen_op, l_sfleu_t_table, gen_op_sfleu)
- void gen_l_sfleu (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_sflts_imm_t_table, gen_op_sflts_imm)
- DEF_2T_OP (generic_gen_op, l_sflts_t_table, gen_op_sflts)
- void gen_l_sflts (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (imm_gen_op, l_sfltu_imm_t_table, gen_op_sfltu_imm)
- DEF_2T_OP (generic_gen_op, l_sfltu_t_table, gen_op_sfltu)
- void gen_l_sfltu (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_1T_OP (generic_gen_op, l_sfne_null_t_table, gen_op_sfne_null)
- DEF_1T_OP (imm_gen_op, l_sfne_imm_t_table, gen_op_sfne_imm)
- DEF_2T_OP (generic_gen_op, l_sfne_t_table, gen_op_sfne)
- void gen_l_sfne (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_2T_OP (imm_gen_op, l_sll_imm_t_table, gen_op_sll_imm)
- DEF_3T_OP (generic_gen_op, l_sll_t_table, gen_op_sll)
- void gen_l_sll (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_2T_OP (imm_gen_op, l_sra_imm_t_table, gen_op_sra_imm)
- DEF_3T_OP (generic_gen_op, l_sra_t_table, gen_op_sra)
- void gen_l_sra (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_2T_OP (imm_gen_op, l_srl_imm_t_table, gen_op_srl_imm)
- DEF_3T_OP (generic_gen_op, l_srl_t_table, gen_op_srl)
- void gen_l_srl (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_2T_OP (generic_gen_op, l_neg_t_table, gen_op_neg)
- DEF_3T_OP (generic_gen_op, l_sub_t_table, gen_op_sub)
- void gen_l_sub (struct op_queue *opq, int param_t[3], int delay_slot)
- void gen_l_sys (struct op_queue *opq, int param_t[3], int delay_slot)
- void gen_l_trap (struct op_queue *opq, int param_t[3], int delay_slot)
- DEF_2T_OP (imm_gen_op, l_xor_imm_t_table, gen_op_xor_imm)
- DEF_3T_OP_NEQ (generic_gen_op, l_xor_t_table, gen_op_xor)
- void gen_l_xor (struct op_queue *opq, int param_t[3], int delay_slot)

- void [gen_lf_add_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_div_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_ftoi_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_itof_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_madd_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_mul_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_rem_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_sfeq_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_sfge_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_sfgt_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_sfle_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_sflt_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_sfne_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)
- void [gen_lf_sub_s](#) (struct [op_queue](#) *opq, int [param_t](#)[3], int delay_slot)

Variables

- [uorreg_t __op_param1](#)
- [uorreg_t __op_param2](#)
- [uorreg_t __op_param3](#)
- [int do_stats](#)
- [static int sigsegv_state = 0](#)
- [static void * sigsegv_addr = NULL](#)
- [static const generic_gen_op set_pc_delay_gpr \[32\]](#)

6.32.1 Define Documentation

- 6.32.1.1 `#define OPS_ENLARGE_BY 5`
- 6.32.1.2 `#define RECED_PAGE_ENLARGE_BY 51200`
- 6.32.1.3 `#define T_NONE (-1)`
- 6.32.1.4 `#define TFLAG_DST 2`
- 6.32.1.5 `#define TFLAG_SAVED 4`
- 6.32.1.6 `#define TFLAG_SOURCED 8`
- 6.32.1.7 `#define TFLAG_SRC 1`

6.32.2 Typedef Documentation

- 6.32.2.1 `typedef void(* generic_gen_op)(struct op_queue *opq, int end)`
- 6.32.2.2 `typedef void(* imm_gen_op)(struct op_queue *opq, int end, uorreg_t imm)`

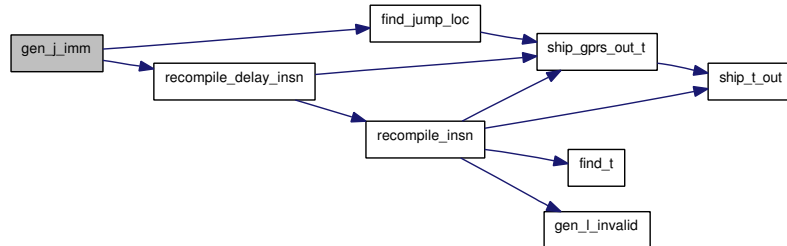
6.32.3 Function Documentation

- 6.32.3.1 `void add_to_op_params (struct op_queue * opq, int end, unsigned long param)`
- 6.32.3.2 `void add_to_opq (struct op_queue * opq, int end, int op)`
- 6.32.3.3 `DEF_1T_OP (imm_gen_op, l_sfne_imm_t_table, gen_op_sfne_imm)`
- 6.32.3.4 `DEF_1T_OP (generic_gen_op, l_sfne_null_t_table, gen_op_sfne_null)`
- 6.32.3.5 `DEF_1T_OP (imm_gen_op, l_sftu_imm_t_table, gen_op_sftu_imm)`
- 6.32.3.6 `DEF_1T_OP (imm_gen_op, l_sfts_imm_t_table, gen_op_sfts_imm)`
- 6.32.3.7 `DEF_1T_OP (imm_gen_op, l_sfleu_imm_t_table, gen_op_sfleu_imm)`
- 6.32.3.8 `DEF_1T_OP (imm_gen_op, l_sfles_imm_t_table, gen_op_sfles_imm)`
- 6.32.3.9 `DEF_1T_OP (imm_gen_op, l_sfgtu_imm_t_table, gen_op_sfgtu_imm)`
- 6.32.3.10 `DEF_1T_OP (generic_gen_op, l_sftu_null_t_table, gen_op_sftu_null)`
- 6.32.3.11 `DEF_1T_OP (generic_gen_op, l_sfgtu_null_t_table, gen_op_sfgtu_null)`
- 6.32.3.12 `DEF_1T_OP (imm_gen_op, l_sfgts_imm_t_table, gen_op_sfgts_imm)`
- 6.32.3.13 `DEF_1T_OP (generic_gen_op, l_sfts_null_t_table, gen_op_sfts_null)`
- 6.32.3.14 `DEF_1T_OP (generic_gen_op, l_sfgts_null_t_table, gen_op_sfgts_null)`
- 6.32.3.15 `DEF_1T_OP (imm_gen_op, l_sfgeu_imm_t_table, gen_op_sfgeu_imm)`
- 6.32.3.16 `DEF_1T_OP (generic_gen_op, l_sfgeu_null_t_table, gen_op_sfgeu_null)`
- 6.32.3.17 `DEF_1T_OP (generic_gen_op, l_sfges_imm_t_table, gen_op_sfges_imm)`
- 6.32.3.18 `DEF_1T_OP (imm_gen_op, l_sfges_imm_t_table, gen_op_sfges_imm)`
- 6.32.3.19 `DEF_1T_OP (generic_gen_op, l_sfges_null_t_table, gen_op_sfges_null)`

6.32.3.107 `static int find_t (struct op_queue * opq, unsigned int reg)` [static]

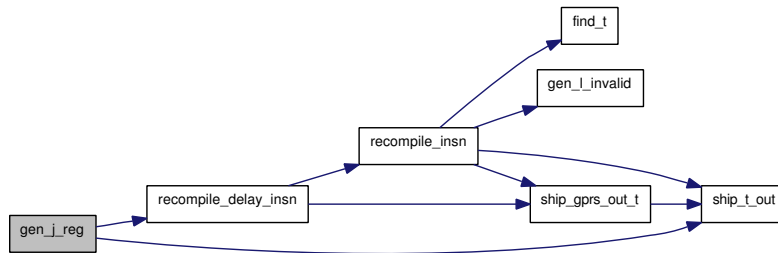
6.32.3.108 `static void gen_j_imm (struct op_queue * opq, oraddr_t off)` [static]

Here is the call graph for this function:



6.32.3.109 `static void gen_j_reg (struct op_queue * opq, unsigned int gpr)` [static]

Here is the call graph for this function:



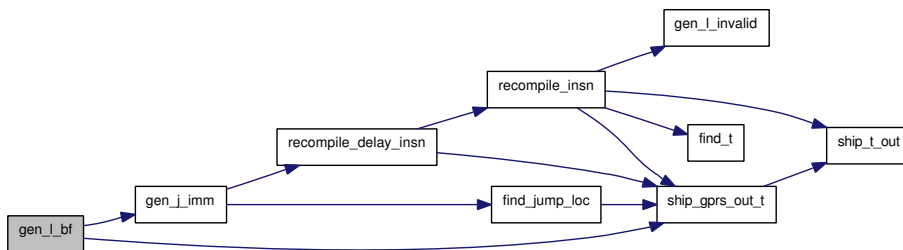
6.32.3.110 `void gen_l_add (struct op_queue * opq, int param_t[3], int delay_slot)`

6.32.3.111 `void gen_l_addc (struct op_queue * opq, int param_t[3], int delay_slot)`

6.32.3.112 `void gen_l_and (struct op_queue * opq, int param_t[3], int delay_slot)`

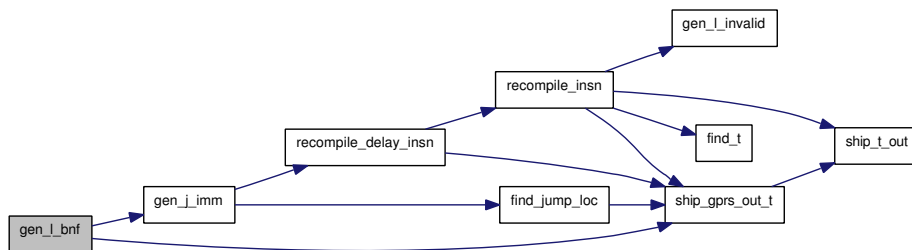
6.32.3.113 `void gen_l_bf (struct op_queue * opq, int param_t[3], int delay_slot)`

Here is the call graph for this function:



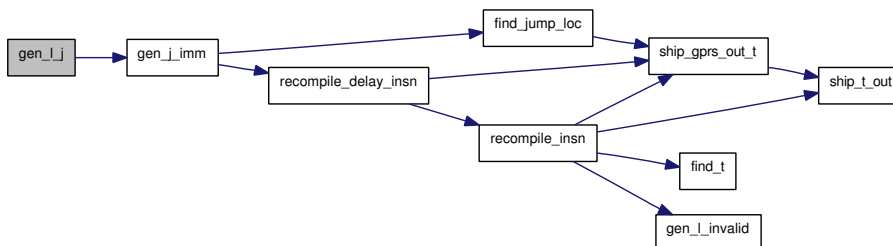
6.32.3.114 void gen_l_bnf (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:



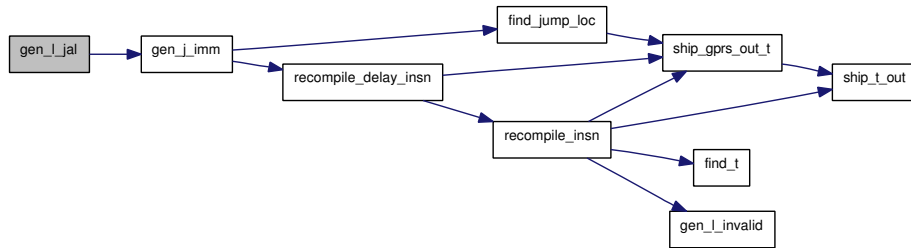
- 6.32.3.115 void gen_l_cmov (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.116 void gen_l_cust1 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.117 void gen_l_cust2 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.118 void gen_l_cust3 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.119 void gen_l_cust4 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.120 void gen_l_cust5 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.121 void gen_l_cust6 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.122 void gen_l_cust7 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.123 void gen_l_cust8 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.124 void gen_l_div (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.125 void gen_l_divu (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.126 void gen_l_extbs (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.127 void gen_l_extbz (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.128 void gen_l_exths (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.129 void gen_l_exthz (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.130 void gen_l_extws (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.131 void gen_l_extwz (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.132 void gen_l_ff1 (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.133 void gen_l_invalid (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.134 void gen_l_j (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:

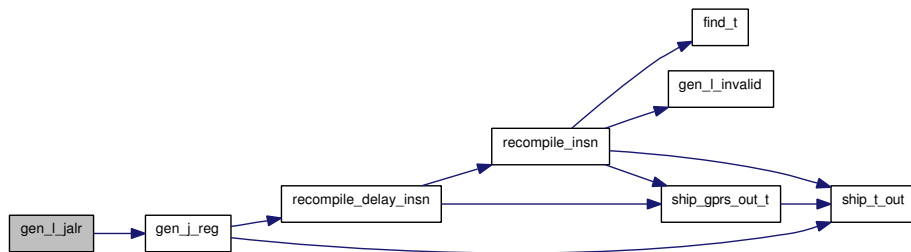


6.32.3.135 void gen_l_jal (struct op_queue * opq, int param_t[3], int delay_slot)

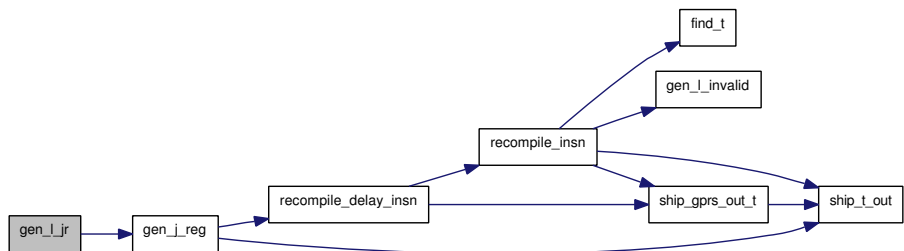
Here is the call graph for this function:

**6.32.3.136 void gen_l_jalr (struct op_queue * opq, int param_t[3], int delay_slot)**

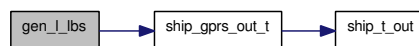
Here is the call graph for this function:

**6.32.3.137 void gen_l_jr (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

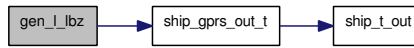
**6.32.3.138 void gen_l_lbs (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

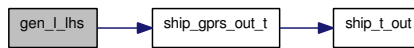


6.32.3.139 void gen_l_lbz (struct op_queue * opq, int param_t[3], int delay_slot)

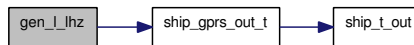
Here is the call graph for this function:

**6.32.3.140 void gen_l_lhs (struct op_queue * opq, int param_t[3], int delay_slot)**

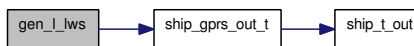
Here is the call graph for this function:

**6.32.3.141 void gen_l_lhz (struct op_queue * opq, int param_t[3], int delay_slot)**

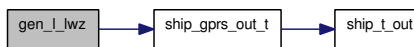
Here is the call graph for this function:

**6.32.3.142 void gen_l_lws (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**6.32.3.143 void gen_l_lwz (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:



6.32.3.144 void `gen_l_mac` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

6.32.3.145 void `gen_l_macrc` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

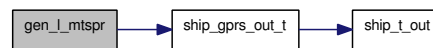
6.32.3.146 void `gen_l_mfspr` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

6.32.3.147 void `gen_l_movhi` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

6.32.3.148 void `gen_l_msb` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

6.32.3.149 void `gen_l_mtspr` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:

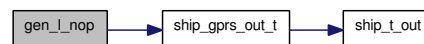


6.32.3.150 void `gen_l_mul` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

6.32.3.151 void `gen_l_mulu` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

6.32.3.152 void `gen_l_nop` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:

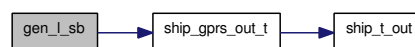


6.32.3.153 void `gen_l_or` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

6.32.3.154 void `gen_l_rfe` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

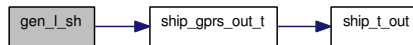
6.32.3.155 void `gen_l_sb` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



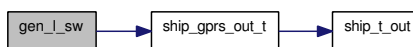
- 6.32.3.156 void gen_l_sfeq (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.157 void gen_l_sfges (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.158 void gen_l_sfgeu (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.159 void gen_l_sfgts (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.160 void gen_l_sfgtu (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.161 void gen_l_sfles (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.162 void gen_l_sfleu (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.163 void gen_l_sfits (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.164 void gen_l_sftu (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.165 void gen_l_sfne (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.166 void gen_l_sh (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:



- 6.32.3.167 void gen_l_sll (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.168 void gen_l_sra (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.169 void gen_l_srl (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.170 void gen_l_sub (struct op_queue * opq, int param_t[3], int delay_slot)
- 6.32.3.171 void gen_l_sw (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:



6.32.3.172 void `gen_l_sys` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

6.32.3.173 void `gen_l_trap` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

6.32.3.174 void `gen_l_xor` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

6.32.3.175 void `gen_lf_add_s` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

Here is the call graph for this function:



6.32.3.176 void `gen_lf_div_s` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

Here is the call graph for this function:



6.32.3.177 void `gen_lf_ftoi_s` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

Here is the call graph for this function:



6.32.3.178 void `gen_lf_itof_s` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

Here is the call graph for this function:



6.32.3.179 void `gen_lf_madd_s` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

Here is the call graph for this function:



6.32.3.180 void gen_lf_mul_s (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:

**6.32.3.181 void gen_lf_rem_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**6.32.3.182 void gen_lf_sfeq_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**6.32.3.183 void gen_lf_sfge_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**6.32.3.184 void gen_lf_sfgt_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**6.32.3.185 void gen_lf_sfle_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:



6.32.3.186 void `gen_lf_sflt_s` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



6.32.3.187 void `gen_lf_sfne_s` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



6.32.3.188 void `gen_lf_sub_s` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



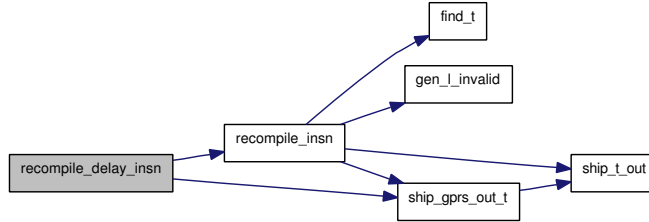
6.32.3.189 static void `gen_op_mark_loc` (`struct op_queue * opq`, `int end`) [static]

Here is the call graph for this function:



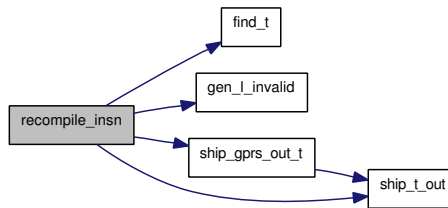
6.32.3.194 static void recompile_delay_insn (struct op_queue * opq) [static]

Here is the call graph for this function:



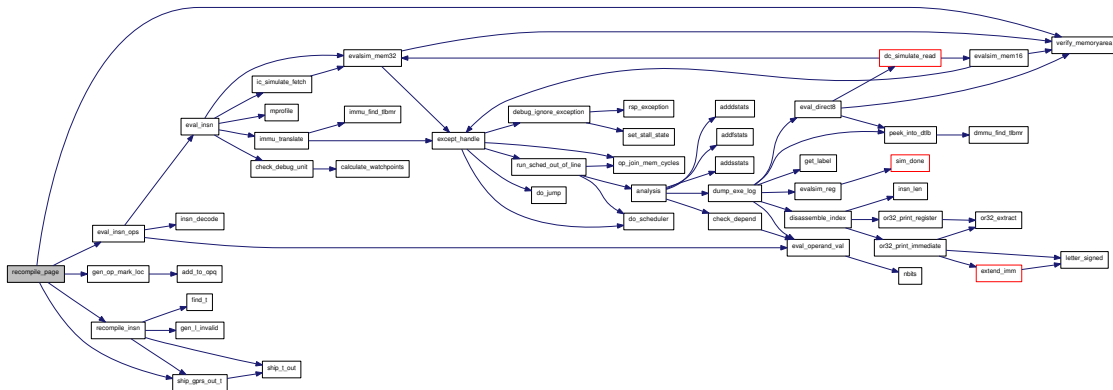
6.32.3.195 static void recompile_insn (struct op_queue * opq, int delay_insn) [static]

Here is the call graph for this function:



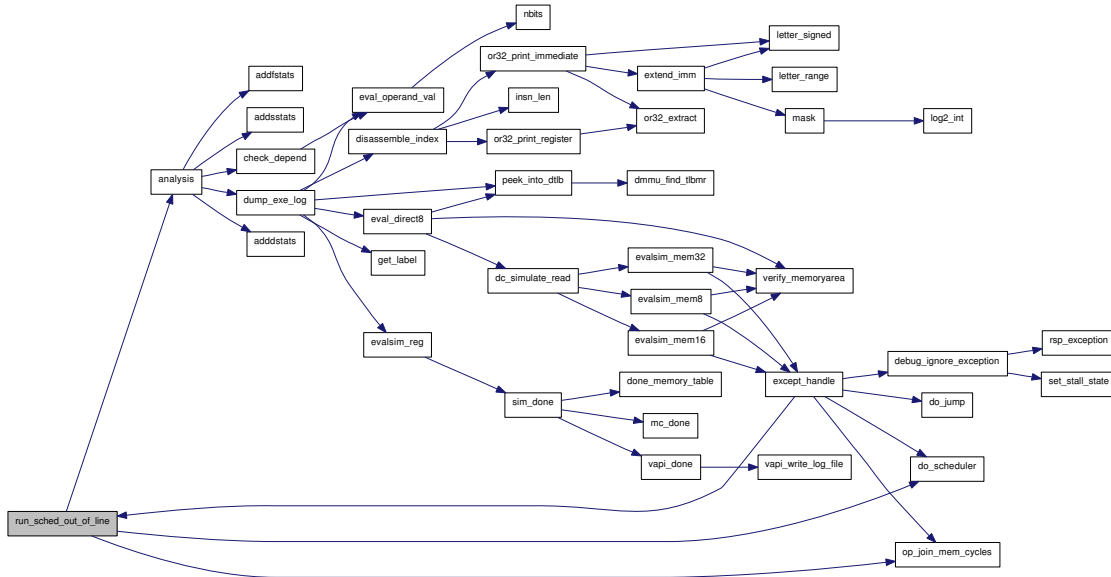
6.32.3.196 void recompile_page (struct dyn_page * dyn)

Here is the call graph for this function:



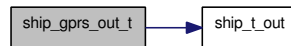
6.32.3.197 void run_sched_out_of_line (void)

Here is the call graph for this function:



6.32.3.198 static void ship_gprs_out_t (struct op_queue * opq) [static]

Here is the call graph for this function:



6.32.3.199 static void ship_t_out (struct op_queue * opq, unsigned int t) [static]

6.32.4 Variable Documentation

6.32.4.1 uorreg_t __op_param1

6.32.4.2 uorreg_t __op_param2

6.32.4.3 uorreg_t __op_param3

6.32.4.4 int do_stats

Whether we are doing statistical analysis. Globally available

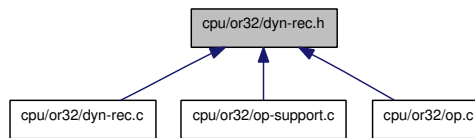
6.32.4.5 `const generic_gen_op set_pc_delay_gpr[32]` [static]

6.32.4.6 `void* sigsegv_addr = NULL` [static]

6.32.4.7 `int sigsegv_state = 0` [static]

6.33 cpu/or32/dyn-rec.h File Reference

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



Data Structures

- struct [dyn_page](#)

Defines

- #define [IMMU_GOT_ENABLED](#) 1
- #define [IMMU_GOT_DISABLED](#) 2
- #define [xglue\(x, y\)](#) x ## y
- #define [glue\(x, y\)](#) xglue(x, y)

Typedefs

- typedef void(* [gen_code_ent](#))(void)

Functions

- void [recompile_page](#) (struct [dyn_page](#) *dyn)
- struct [dyn_page](#) * [new_dp](#) (oraddr_t page)
- void [add_to_opq](#) (struct [op_queue](#) *opq, int end, int op)
- void [add_to_op_params](#) (struct [op_queue](#) *opq, int end, unsigned long param)
- void * [enough_host_page](#) (struct [dyn_page](#) *dp, void *cur, unsigned int *len, unsigned int amount)
- void [init_dyn_recomp](#) (void)
- void [run_sched_out_of_line](#) (void)
- void [recheck_immu](#) (int got_en_dis)
- void [enter_dyn_code](#) (oraddr_t addr, struct [dyn_page](#) *dp)
- void [dyn_checkwrite](#) (oraddr_t addr)
- void [dyn_main](#) (void)

Variables

- void * [rec_stack_base](#)

6.33.1 Define Documentation

6.33.1.1 `#define glue(x, y) xglue(x, y)`

6.33.1.2 `#define IMMU_GOT_DISABLED 2`

6.33.1.3 `#define IMMU_GOT_ENABLED 1`

6.33.1.4 `#define xglue(x, y) x ## y`

6.33.2 Typedef Documentation

6.33.2.1 `typedef void(* gen_code_ent)(void)`

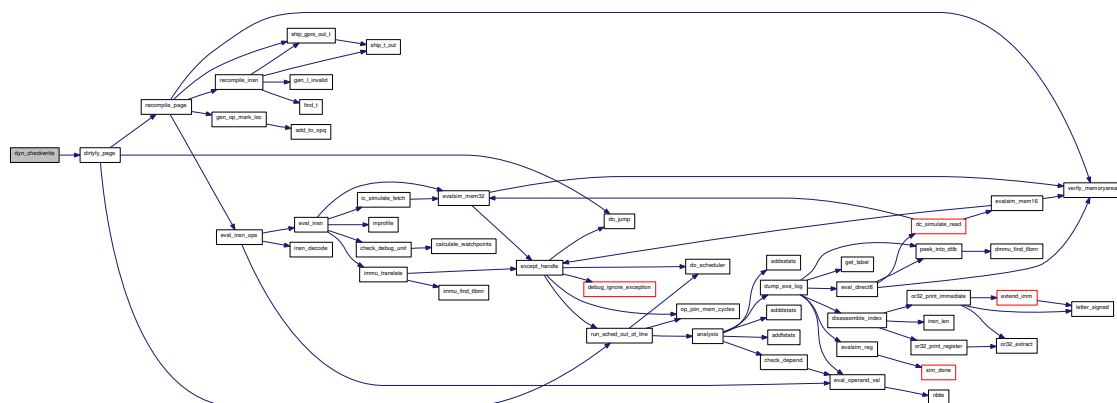
6.33.3 Function Documentation

6.33.3.1 `void add_to_op_params (struct op_queue * opq, int end, unsigned long param)`

6.33.3.2 `void add_to_opq (struct op_queue * opq, int end, int op)`

6.33.3.3 `void dyn_checkwrite (oraddr_t addr)`

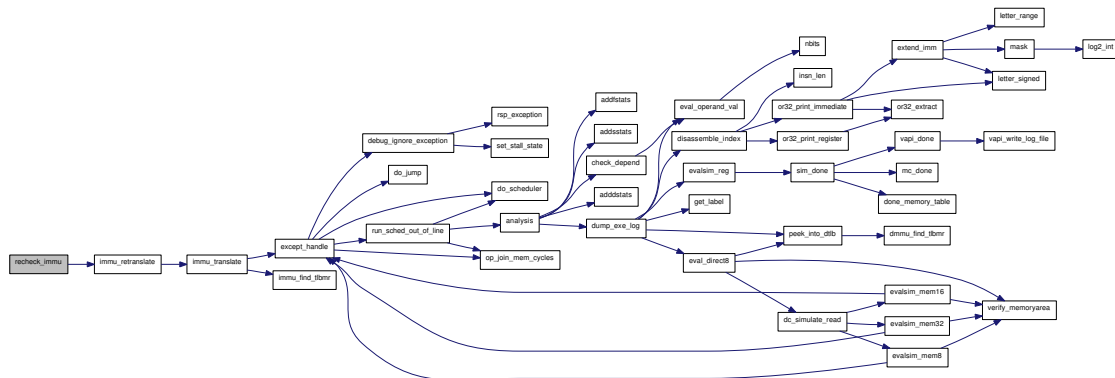
Here is the call graph for this function:



6.33.3.8 struct dyn_page* new_dp (oraddr_t page) [read]

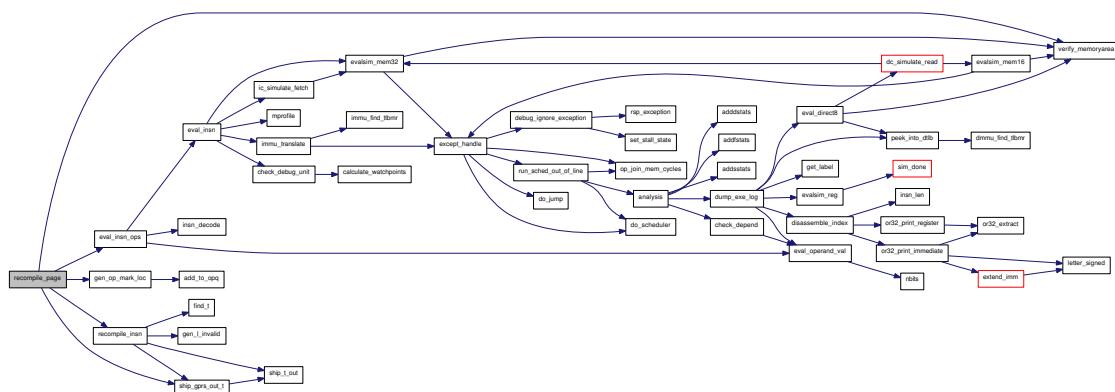
6.33.3.9 void recheck_immu (int got_en_dis)

Here is the call graph for this function:



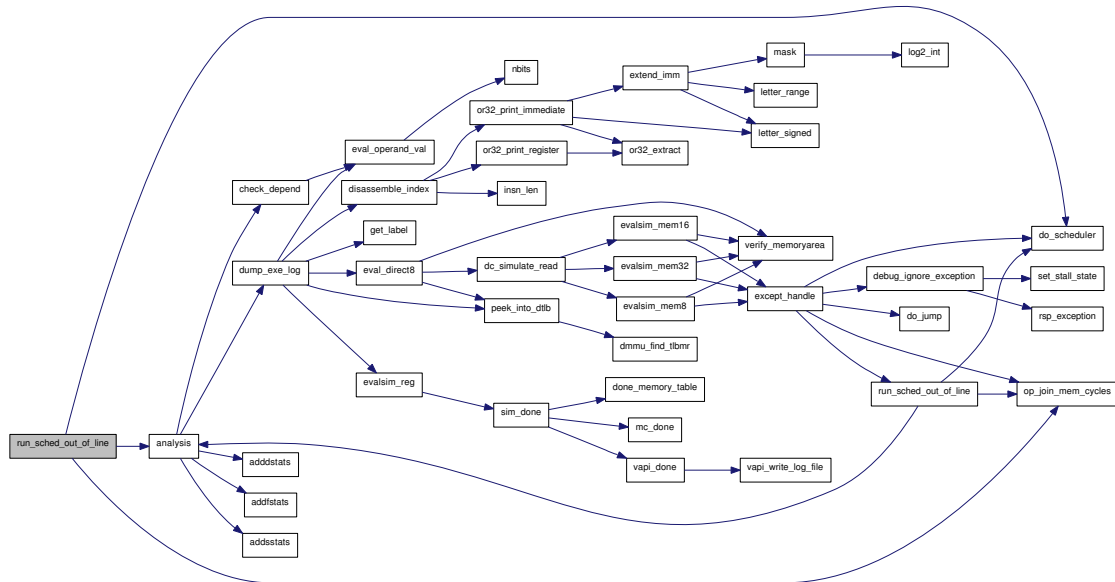
6.33.3.10 void recompile_page (struct dyn_page * dyn)

Here is the call graph for this function:



6.33.3.11 void run_sched_out_of_line (void)

Here is the call graph for this function:



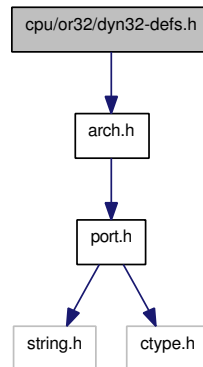
6.33.4 Variable Documentation

6.33.4.1 void* rec_stack_base

6.34 cpu/or32/dyn32-defs.h File Reference

```
#include "arch.h"
```

Include dependency graph for dyn32-defs.h:



Data Structures

- struct [op_queue](#)

Functions

- void `gen_l_add` PARAMS ((struct [op_queue](#) *, int *, int))
- void `l_none` (struct [op_queue](#) *opq, int *param_t, int delay_slot)

6.34.1 Function Documentation

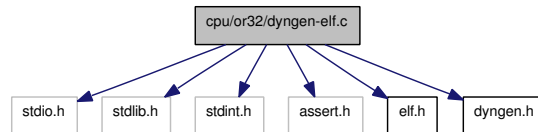
6.34.1.1 void `l_none` (struct [op_queue](#) * *opq*, int * *param_t*, int *delay_slot*)

6.34.1.2 void `gen_lf_sub_s` PARAMS ((struct [op_queue](#) *, int *, int))

6.35 cpu/or32/dyngen-elf.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <assert.h>
#include <elf.h>
#include "dyngen.h"
```

Include dependency graph for dyngen-elf.c:



Data Structures

- struct [elf_obj](#)

Functions

- void * [elf_open_obj](#) (const char *file)
- void [elf_close_obj](#) (void *e_obj)
- static [Elf32_Sym](#) * [elf_find_func](#) (struct [elf_obj](#) *obj, unsigned int func)
- char * [elf_get_func_name](#) (void *e_obj, unsigned int func)
- unsigned int [elf_get_func_len](#) (void *e_obj, unsigned int func)
- void * [elf_get_func_start](#) (void *e_obj, unsigned int func)
- static char * [elf_get_sym_name](#) (struct [elf_obj](#) *obj, unsigned int sym)
- int [elf_get_func_reloc](#) (void *e_obj, unsigned int func, unsigned int relocn, struct [reloc](#) *reloc)

Variables

- struct [bff](#) [bffb](#)

6.35.1 Function Documentation

6.35.1.1 void [elf_close_obj](#) (void * *e_obj*)

6.35.1.2 static [Elf32_Sym](#)* [elf_find_func](#) (struct [elf_obj](#) * *obj*, unsigned int *func*) [static]

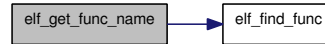
6.35.1.3 unsigned int [elf_get_func_len](#) (void * *e_obj*, unsigned int *func*)

Here is the call graph for this function:

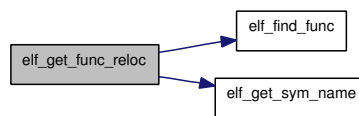


6.35.1.4 `char* elf_get_func_name (void * e_obj, unsigned int func)`

Here is the call graph for this function:

**6.35.1.5** `int elf_get_func_reloc (void * e_obj, unsigned int func, unsigned int relocn, struct reloc * reloc)`

Here is the call graph for this function:

**6.35.1.6** `void* elf_get_func_start (void * e_obj, unsigned int func)`

Here is the call graph for this function:

**6.35.1.7** `static char* elf_get_sym_name (struct elf_obj * obj, unsigned int sym)` [static]**6.35.1.8** `void* elf_open_obj (const char * file)`**6.35.2 Variable Documentation****6.35.2.1 struct bff bffs**

Initial value:

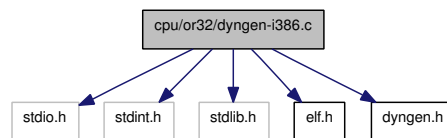
```

{
  elf_open_obj,
  elf_close_obj,
  elf_get_func_name,
  elf_get_func_start,
  elf_get_func_len,
  elf_get_func_reloc }
  
```

6.36 cpu/or32/dyngen-i386.c File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <elf.h>
#include "dyngen.h"
```

Include dependency graph for dyngen-i386.c:



Defines

- #define [RET_OPCODE](#) 0xc3

Functions

- unsigned int [i386_get_real_func_len](#) (void *f_start, unsigned int f_len, char *name)
- void [i386_gen_reloc](#) (FILE *f, struct [reloc](#) *reloc, unsigned int param)
- void [i386_gen_func_reloc](#) (FILE *f, struct [reloc](#) *reloc)

Variables

- struct [archf](#) [archfs](#)

6.36.1 Define Documentation

6.36.1.1 #define RET_OPCODE 0xc3

6.36.2 Function Documentation

6.36.2.1 void i386_gen_func_reloc (FILE *f, struct *reloc* *reloc)

6.36.2.2 void i386_gen_reloc (FILE *f, struct *reloc* *reloc, unsigned int param)

6.36.2.3 unsigned int i386_get_real_func_len (void *f_start, unsigned int f_len, char *name)

6.36.3 Variable Documentation

6.36.3.1 struct *archf* *archfs*

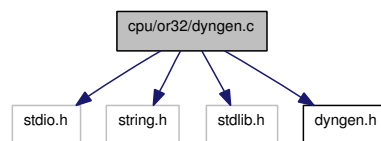
Initial value:

```
{
  i386_get_real_func_len,
  i386_gen_reloc,
  i386_gen_func_reloc
}
```

6.37 cpu/or32/dyngen.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "dyngen.h"
```

Include dependency graph for dyngen.c:



Defines

- #define `OP_FUNC_PREFIX` "op_"
- #define `OP_FUNC_PARAM_PREFIX` "__op_param"
- #define `MAX_PARAMS` (3 + 1)

Functions

- static void `gen_func_proto` (FILE *f, const char *name, int *params)
- int `main` (int argc, char **argv)

Variables

- static const char * `c_file_head`
- static const char * `gen_code_proto`
- static const char * `c_sw_file_head`
- static const char * `c_sw_file_tail`
- static const char * `c_rel_file_head`
- static const char * `c_rel_file_tail`

6.37.1 Define Documentation

6.37.1.1 `#define MAX_PARAMS (3 + 1)`

6.37.1.2 `#define OP_FUNC_PARAM_PREFIX "__op_param"`

6.37.1.3 `#define OP_FUNC_PREFIX "op_"`

6.37.2 Function Documentation

6.37.2.1 `static void gen_func_proto (FILE *f, const char *name, int *params) [static]`

6.37.2.2 `int main (int argc, char **argv)`

Here is the call graph for this function:



6.37.3 Variable Documentation

6.37.3.1 `const char* c_file_head [static]`

Initial value:

```

#include "config.h"\n"
"\n"
#include <inttypes.h>\n"
"\n"
#include "arch.h"\n"
#include "opcode/or32.h"\n"
#include "spr-defs.h"\n"
#include "i386-regs.h"\n"
#include "abstract.h"\n"
"\n"
#include "dyn-rec.h"\n"
#include "%s"\n"
"\n"
  
```

6.37.3.2 `const char* c_rel_file_head [static]`

6.37.3.3 `const char* c_rel_file_tail [static]`

Initial value:

```

    }\n"
  }\n"
  opq = opq->next;\n"
}\n"
}
  
```

6.37.3.4 `const char* c_sw_file_head` [static]

6.37.3.5 `const char* c_sw_file_tail` [static]

Initial value:

```
"    }\n"
"  }\n"
"  opq = opq->next;\n"
"  pc += 4;\n"
"  }\n"
"\n"
"  dp->host_len = host_cur - dp->host_page;\n"
"  dp->host_page = realloc(dp->host_page, dp->host_len);\n"
"}\n"
```

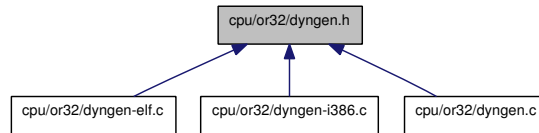
6.37.3.6 `const char* gen_code_proto` [static]

Initial value:

```
"void gen_code(struct op_queue *opq, struct dyn_page *dp);\n"
"void patch_relocs(struct op_queue *opq, void *host_page);\n"
"\n"
```

6.38 cpu/or32/dyngen.h File Reference

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



Data Structures

- struct [reloc](#)
- struct [bff](#)
- struct [archf](#)

Variables

- struct [bff](#) [bffs](#)
- struct [archf](#) [archfs](#)

6.38.1 Variable Documentation

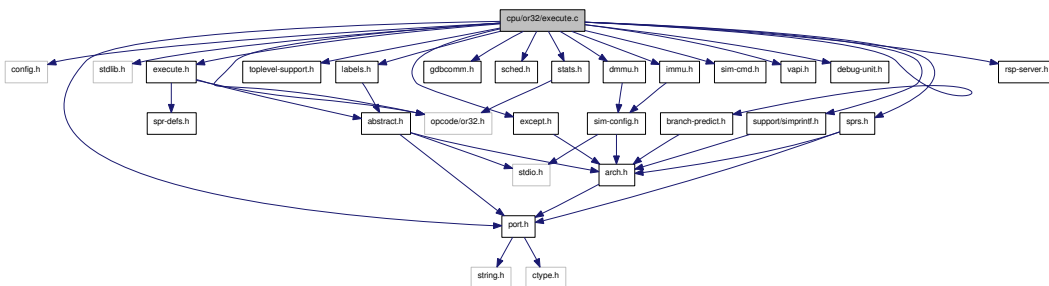
6.38.1.1 struct archf archfs

6.38.1.2 struct bff bffs

6.39 cpu/or32/execute.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "execute.h"
#include "toplevel-support.h"
#include "except.h"
#include "labels.h"
#include "gdbcomm.h"
#include "sched.h"
#include "stats.h"
#include "opcode/or32.h"
#include "dmmu.h"
#include "immu.h"
#include "sim-cmd.h"
#include "vapi.h"
#include "debug-unit.h"
#include "branch-predict.h"
#include "support/simprintf.h"
#include "sprs.h"
#include "rsp-server.h"
```

Include dependency graph for execute.c:



Functions

- static void [decode_execute](#) (struct [iqueue_entry](#) *current)
- [uorreg_t evalsim_reg](#) (unsigned int regno)
- void [setsim_reg](#) (unsigned int regno, [uorreg_t](#) value)
- [uorreg_t eval_operand_val](#) (uint32_t insn, struct insn_op_struct *opd)
- static int [check_depend](#) (struct [iqueue_entry](#) *prev, struct [iqueue_entry](#) *next)
- static int [fetch](#) ()

- static void `update_pc` ()
- void `analysis` (struct `iqueue_entry` **current*)
- static void `sbuf_store` (int cyc)
- static void `sbuf_load` ()
- void `dump_exe_log` ()
- void `dumpreg` ()
- static void `decode_execute_wrapper` (struct `iqueue_entry` **current*)
- void `cpu_reset` ()
- int `cpu_clock` ()
- void `l_invalid` ()
- void `exec_main` ()

Variables

- struct `cpu_state` `cpu_state`
- `oraddr_t` `pcnext`
- int `sbuf_wait_cyc` = 0
- int `sbuf_total_cyc` = 0
- int `do_stats` = 0
- struct `hist_exec` * `hist_exec_tail` = NULL
- static int `multissue` [20]
- static int `issued_per_cycle` = 4
- static int `sbuf_head` = 0
- static int `sbuf_tail` = 0
- static int `sbuf_count` = 0
- static int `sbuf_buf` [MAX_SBUF_LEN] = { 0 }
- static int `sbuf_prev_cycles` = 0
- static int `breakpoint`
- static int `next_delay_insn`

6.39.1 Function Documentation

6.39.1.1 void `analysis` (struct `iqueue_entry` * *current*)

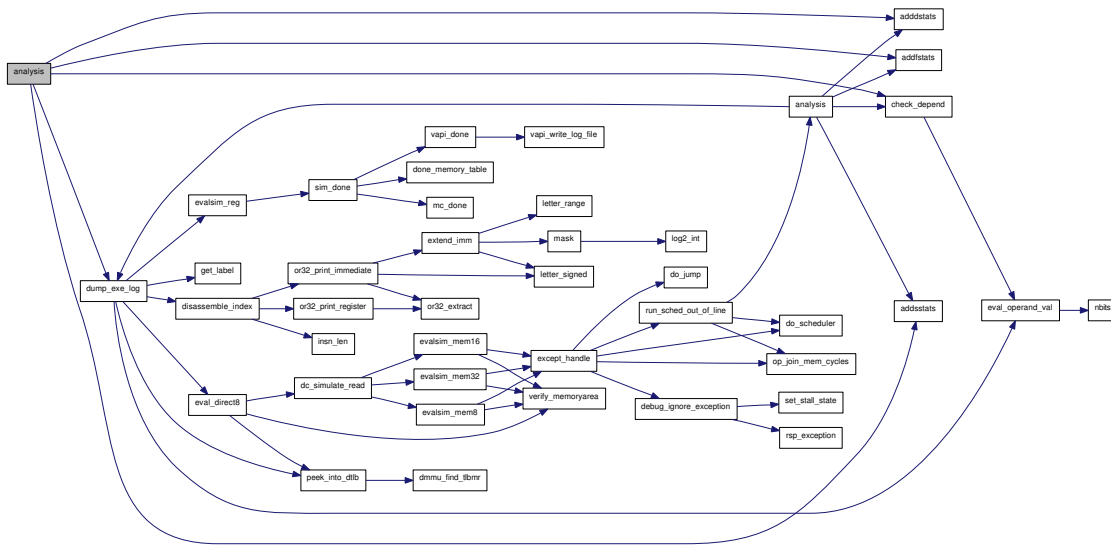
Perform analysis of the instruction being executed

This could be static for SIMPLE_EXECUTION, but made global for general use.

Parameters:

← *current* The instruction being executed

Here is the call graph for this function:



6.39.1.2 static int check_depend (struct iqueue_entry * prev, struct iqueue_entry * next) [static]

Does source operand depend on computation of dest operand?

Cycle t Cycle t+1 dst: irrelevant src: immediate always 0 dst: reg1 direct src: reg2 direct 0 if reg1 != reg2
dst: reg1 disp src: reg2 direct always 0 dst: reg1 direct src: reg2 disp 0 if reg1 != reg2 dst: reg1 disp src:
reg2 disp always 1 (store must finish before load) dst: flag src: flag always 1

Parameters:

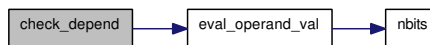
← *prev* Previous instruction

← *next* Next instruction

Returns:

Non-zero if yes.

Here is the call graph for this function:



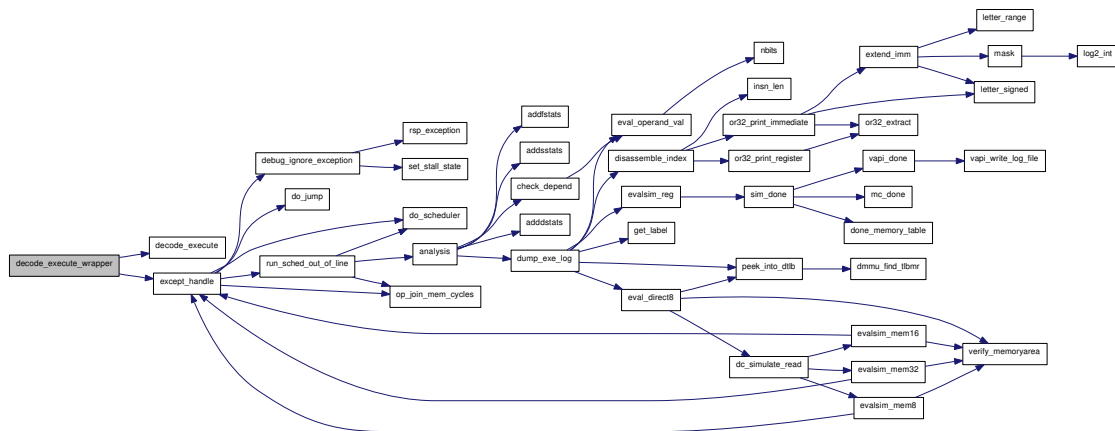
6.39.1.3 int cpu_clock ()

Simulates one CPU clock cycle

Returns:

non-zero if a breakpoint is hit, zero otherwise.

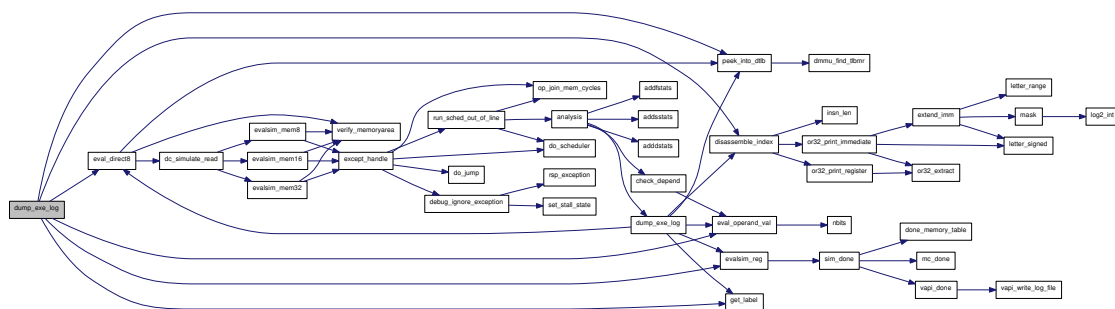
Here is the call graph for this function:



6.39.1.7 void dump_exe_log ()

Outputs disassembled instruction

Here is the call graph for this function:

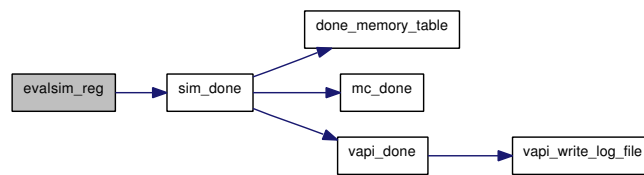


6.39.1.8 void dumpreg ()

Dump registers

Supports the CLI 'r' and 't' commands

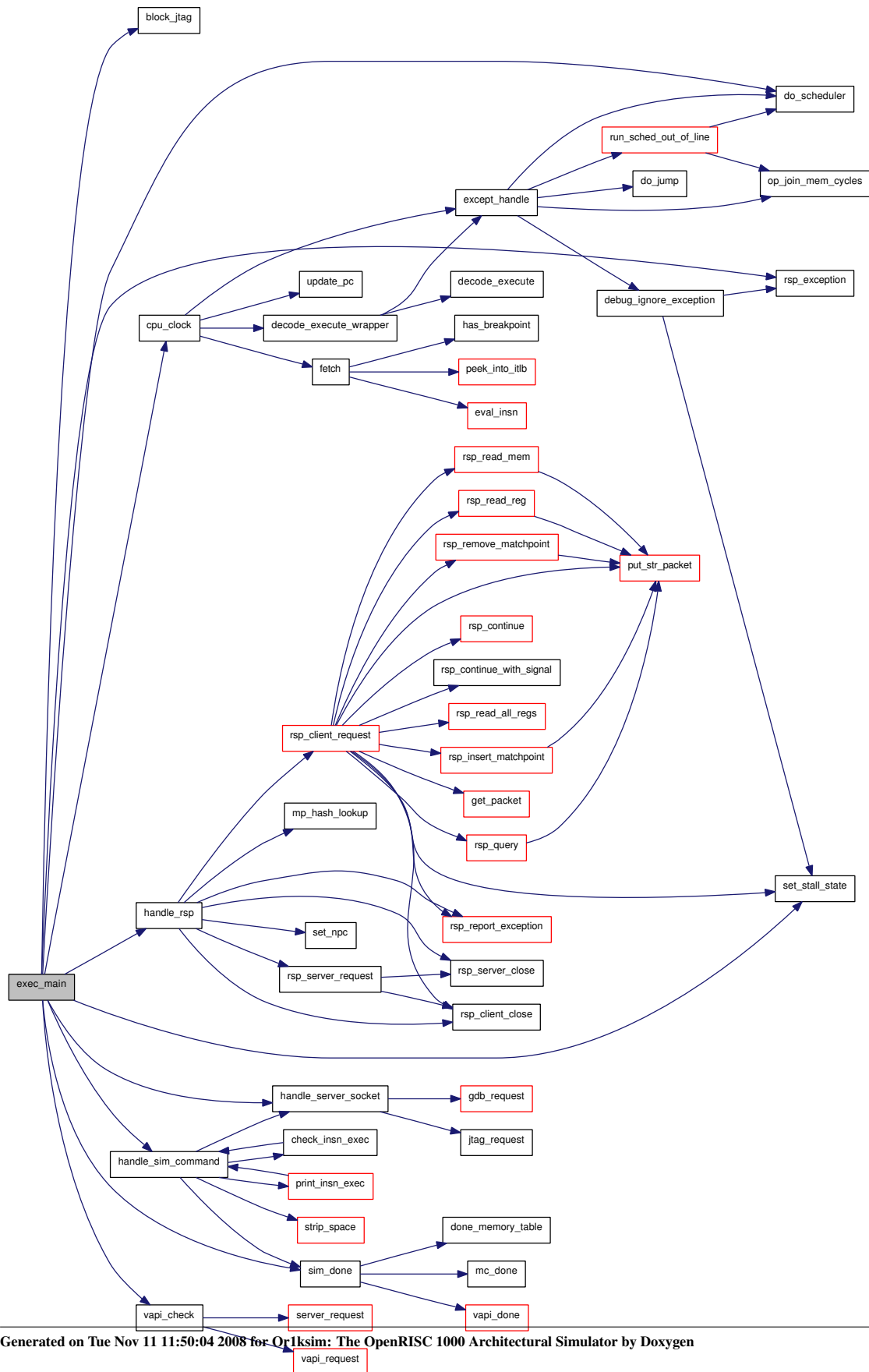
Here is the call graph for this function:



6.39.1.11 void exec_main ()

The main execution loop

Here is the call graph for this function:



6.39.1.12 static int fetch () [static]

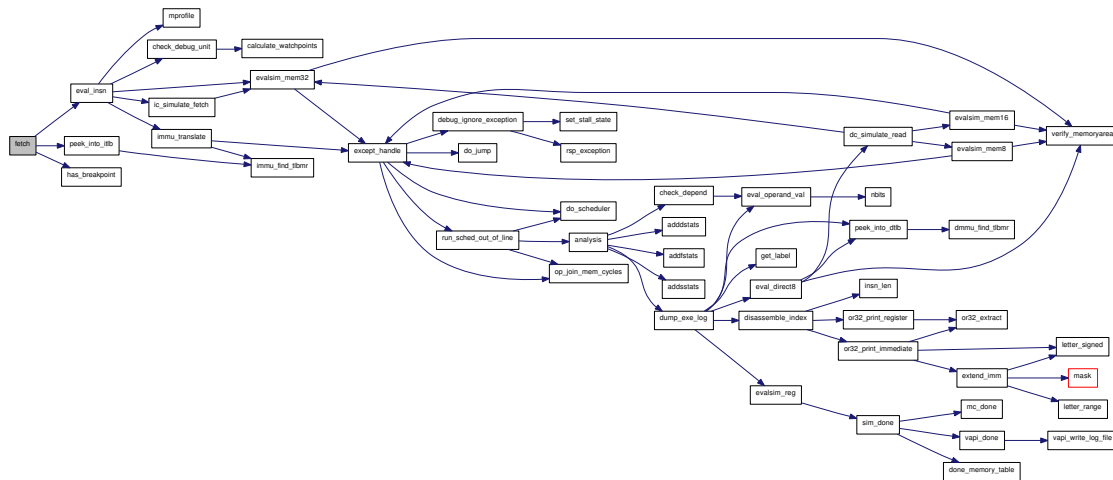
Should instruction NOT be executed?

Modified by CZ 26/05/01 for new mode execution.

Returns:

Nonzero if instruction should NOT be executed

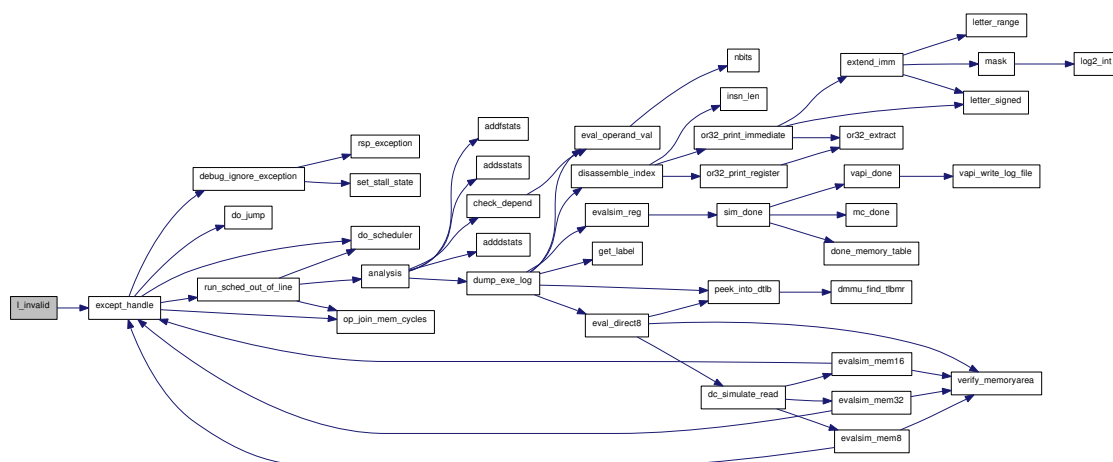
Here is the call graph for this function:



6.39.1.13 void l_invalid ()

If decoding cannot be found, call this function

Here is the call graph for this function:



6.39.1.14 static void sbuf_load () [static]

Store buffer analysis for load instructions

Previous stores should commit, before any load

6.39.1.15 static void sbuf_store (int cyc) [static]

Store buffer analysis for store instructions

Stores are accumulated and committed when IO is idle

Parameters:

← *cyc* Number of cycles being analysed

6.39.1.16 void setsim_reg (unsigned int regno, uorreg_t value)

Set a specific register with value

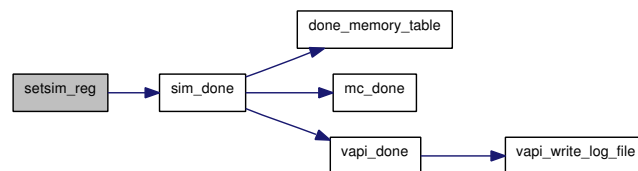
Implementation specific. Abort if we are given a duff register.

Parameters:

← *regno* The register of interest

← *value* The value to be set

Here is the call graph for this function:

**6.39.1.17 static void update_pc () [static]**

This code actually updates the PC value

6.39.2 Variable Documentation**6.39.2.1 int breakpoint [static]****6.39.2.2 struct cpu_state cpu_state**

Current cpu state. Globally available.

6.39.2.3 int do_stats = 0

Whether we are doing statistical analysis. Globally available

6.39.2.4 struct hist_exec* hist_exec_tail = NULL

History of execution. Globally available

6.39.2.5 int issued_per_cycle = 4 [static]**6.39.2.6 int multissue[20] [static]****6.39.2.7 int next_delay_insn [static]****6.39.2.8 oraddr_t pcnext**

Temporary program counter. Globally available

6.39.2.9 int sbuf_buf[MAX_SBUF_LEN] = { 0 } [static]**6.39.2.10 int sbuf_count = 0 [static]****6.39.2.11 int sbuf_head = 0 [static]****6.39.2.12 int sbuf_prev_cycles = 0 [static]****6.39.2.13 int sbuf_tail = 0 [static]****6.39.2.14 int sbuf_total_cyc = 0**

Number of total store cycles. Globally available

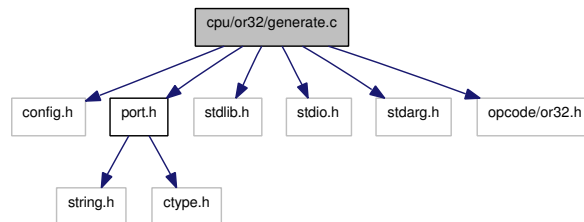
6.39.2.15 int sbuf_wait_cyc = 0

Num cycles waiting for stores to complete. Globally available

6.40 cpu/or32/generate.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include <stdarg.h>
#include "opcode/or32.h"
```

Include dependency graph for generate.c:



Functions

- static int [shift_fprintf](#) (int level, FILE *f, const char *fmt,...)
- int [output_function](#) (FILE *fo, const char *func_name, int level)
- static int [gen_eval_operands](#) (FILE *fo, int insn_index, int level)
- static int [output_call](#) (FILE *fo, int index, int level)
- static int [generate_header](#) (FILE *fo)
- static int [generate_footer](#) (FILE *fo)
- static int [generate_body](#) (FILE *fo, unsigned long *a, unsigned long cur_mask, int level)
- int [main](#) (int argc, char *argv[])

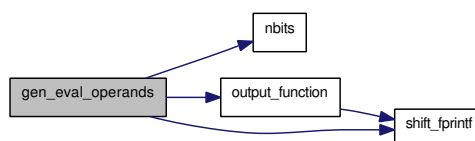
Variables

- static char * [in_file](#)
- static char * [out_file](#)
- static int [write_to_reg](#)
- static int [out_lines](#) = 0

6.40.1 Function Documentation

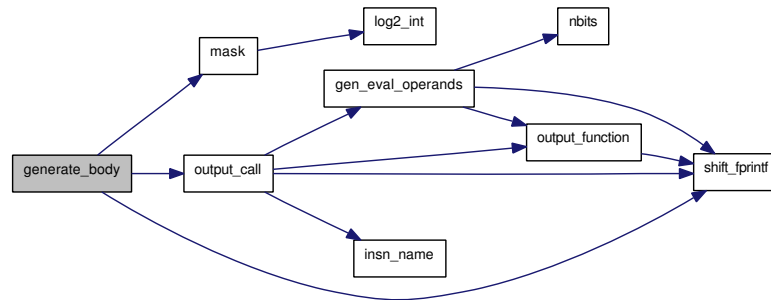
6.40.1.1 static int [gen_eval_operands](#) (FILE *fo, int *insn_index*, int *level*) [static]

Here is the call graph for this function:



6.40.1.2 static int generate_body (FILE *fo, unsigned long *a, unsigned long cur_mask, int level) [static]

Here is the call graph for this function:

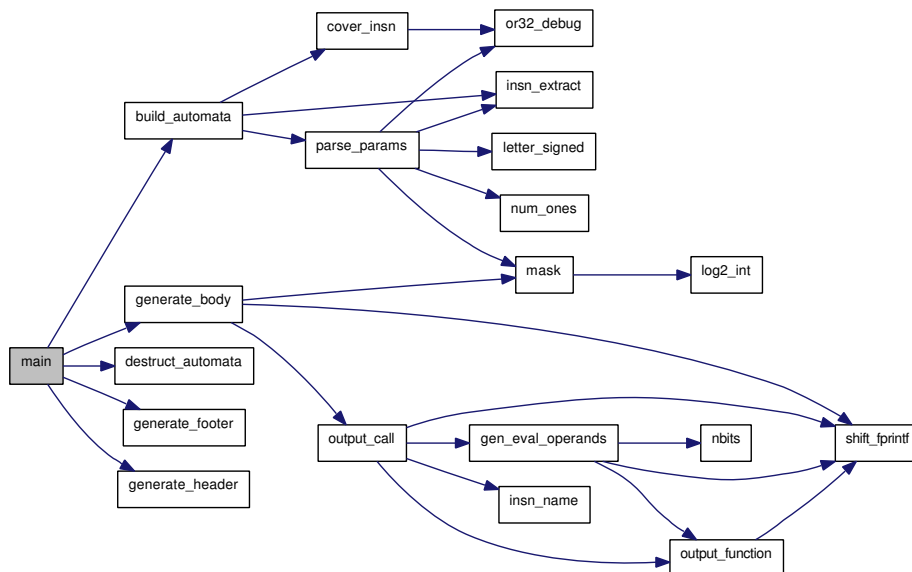


6.40.1.3 static int generate_footer (FILE *fo) [static]

6.40.1.4 static int generate_header (FILE *fo) [static]

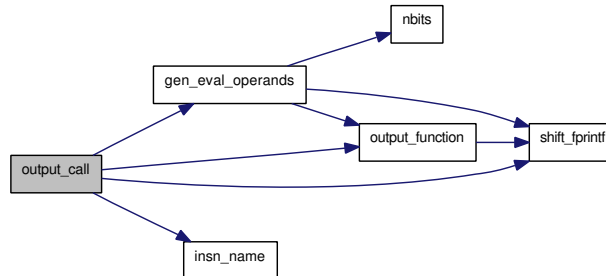
6.40.1.5 int main (int argc, char *argv[])

Here is the call graph for this function:



6.40.1.6 `static int output_call (FILE *fo, int index, int level)` [static]

Here is the call graph for this function:

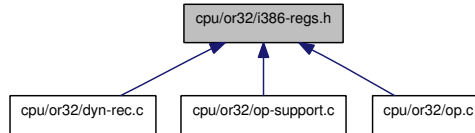
**6.40.1.7** `int output_function (FILE *fo, const char *func_name, int level)`

Here is the call graph for this function:

**6.40.1.8** `static int shift_fprintf (int level, FILE *f, const char *fmt, ...)` [static]**6.40.2** Variable Documentation**6.40.2.1** `char* in_file` [static]**6.40.2.2** `char* out_file` [static]**6.40.2.3** `int out_lines = 0` [static]**6.40.2.4** `int write_to_reg` [static]

6.41 cpu/or32/i386-regs.h File Reference

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



Defines

- #define `CPU_STATE_REG` "ebp"
- #define `T0_REG` "ebx"
- #define `T1_REG` "esi"
- #define `T2_REG` "edi"
- #define `NUM_T_REGS` 3

6.41.1 Define Documentation

6.41.1.1 #define `CPU_STATE_REG` "ebp"

6.41.1.2 #define `NUM_T_REGS` 3

6.41.1.3 #define `T0_REG` "ebx"

6.41.1.4 #define `T1_REG` "esi"

6.41.1.5 #define `T2_REG` "edi"

6.42 cpu/or32/insnset.c File Reference

Functions

- [INSTRUCTION \(l_add\)](#)
- [INSTRUCTION \(l_addc\)](#)
- [INSTRUCTION \(l_sw\)](#)
- [INSTRUCTION \(l_sb\)](#)
- [INSTRUCTION \(l_sh\)](#)
- [INSTRUCTION \(l_lwz\)](#)
- [INSTRUCTION \(l_lbs\)](#)
- [INSTRUCTION \(l_lbz\)](#)
- [INSTRUCTION \(l_lhs\)](#)
- [INSTRUCTION \(l_lhz\)](#)
- [INSTRUCTION \(l_movhi\)](#)
- [INSTRUCTION \(l_and\)](#)
- [INSTRUCTION \(l_or\)](#)
- [INSTRUCTION \(l_xor\)](#)
- [INSTRUCTION \(l_sub\)](#)
- [INSTRUCTION \(l_mul\)](#)
- [INSTRUCTION \(l_div\)](#)
- [INSTRUCTION \(l_divu\)](#)
- [INSTRUCTION \(l_sll\)](#)
- [INSTRUCTION \(l_sra\)](#)
- [INSTRUCTION \(l_srl\)](#)
- [INSTRUCTION \(l_bf\)](#)
- [INSTRUCTION \(l_bnf\)](#)
- [INSTRUCTION \(l_j\)](#)
- [INSTRUCTION \(l_jal\)](#)
- [INSTRUCTION \(l_jalr\)](#)
- [INSTRUCTION \(l_jr\)](#)
- [INSTRUCTION \(l_rfe\)](#)
- [INSTRUCTION \(l_nop\)](#)
- [INSTRUCTION \(l_sfeq\)](#)
- [INSTRUCTION \(l_sfne\)](#)
- [INSTRUCTION \(l_sfgts\)](#)
- [INSTRUCTION \(l_sfges\)](#)
- [INSTRUCTION \(l_sflts\)](#)
- [INSTRUCTION \(l_sfls\)](#)
- [INSTRUCTION \(l_sfgtu\)](#)
- [INSTRUCTION \(l_sfgeu\)](#)
- [INSTRUCTION \(l_sfltu\)](#)
- [INSTRUCTION \(l_sfleu\)](#)
- [INSTRUCTION \(l_extbs\)](#)
- [INSTRUCTION \(l_extbz\)](#)
- [INSTRUCTION \(l_exths\)](#)
- [INSTRUCTION \(l_exthz\)](#)
- [INSTRUCTION \(l_extws\)](#)
- [INSTRUCTION \(l_extwz\)](#)
- [INSTRUCTION \(l_mtspr\)](#)

- [INSTRUCTION](#) (l_mfspr)
- [INSTRUCTION](#) (l_sys)
- [INSTRUCTION](#) (l_trap)
- [INSTRUCTION](#) (l_mac)
- [INSTRUCTION](#) (l_msb)
- [INSTRUCTION](#) (l_macrc)
- [INSTRUCTION](#) (l_cmov)
- [INSTRUCTION](#) (l_ff1)
- [INSTRUCTION](#) (lf_add_s)
- [INSTRUCTION](#) (lf_div_s)
- [INSTRUCTION](#) (lf_ftoi_s)
- [INSTRUCTION](#) (lf_itof_s)
- [INSTRUCTION](#) (lf_madd_s)
- [INSTRUCTION](#) (lf_mul_s)
- [INSTRUCTION](#) (lf_rem_s)
- [INSTRUCTION](#) (lf_sfeq_s)
- [INSTRUCTION](#) (lf_sfge_s)
- [INSTRUCTION](#) (lf_sfgt_s)
- [INSTRUCTION](#) (lf_sfle_s)
- [INSTRUCTION](#) (lf_sflt_s)
- [INSTRUCTION](#) (lf_sfne_s)
- [INSTRUCTION](#) (lf_sub_s)
- [INSTRUCTION](#) (l_cust1)
- [INSTRUCTION](#) (l_cust2)
- [INSTRUCTION](#) (l_cust3)
- [INSTRUCTION](#) (l_cust4)

6.42.1 Function Documentation

6.42.1.1 INSTRUCTION (l_cust4)

6.42.1.2 INSTRUCTION (l_cust3)

6.42.1.3 INSTRUCTION (l_cust2)

6.42.1.4 INSTRUCTION (l_cust1)

6.42.1.5 INSTRUCTION (lf_sub_s)

6.42.1.6 INSTRUCTION (lf_sfne_s)

6.42.1.7 INSTRUCTION (lf_sflt_s)

6.42.1.8 INSTRUCTION (lf_sfle_s)

6.42.1.9 INSTRUCTION (lf_sfgt_s)

6.42.1.10 INSTRUCTION (lf_sfge_s)

6.42.1.11 INSTRUCTION (lf_sfeq_s)

6.42.1.12 INSTRUCTION (lf_rem_s)

6.42.1.13 INSTRUCTION (lf_mul_s)

6.42.1.14 INSTRUCTION (lf_madd_s)

6.42.1.15 INSTRUCTION (lf_itof_s)

6.42.1.16 INSTRUCTION (lf_ftoi_s)

6.42.1.17 INSTRUCTION (lf_div_s)

6.42.1.18 INSTRUCTION (lf_add_s)

6.42.1.19 INSTRUCTION (l_ff1)

6.42.1.20 INSTRUCTION (l_cmov)

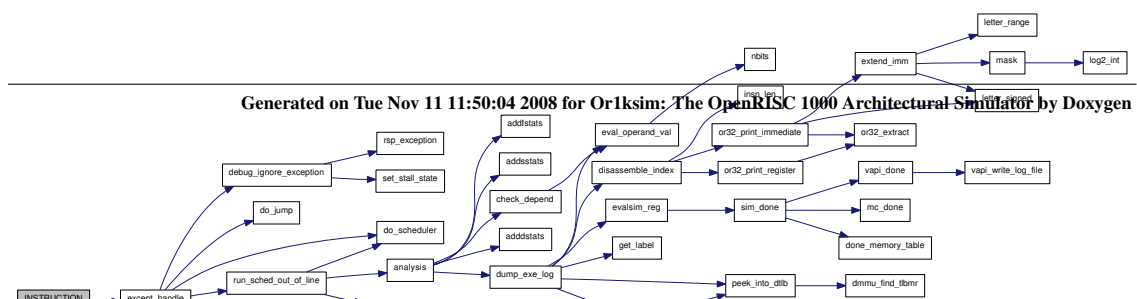
6.42.1.21 INSTRUCTION (l_macrc)

6.42.1.22 INSTRUCTION (l_msb)

6.42.1.23 INSTRUCTION (l_mac)

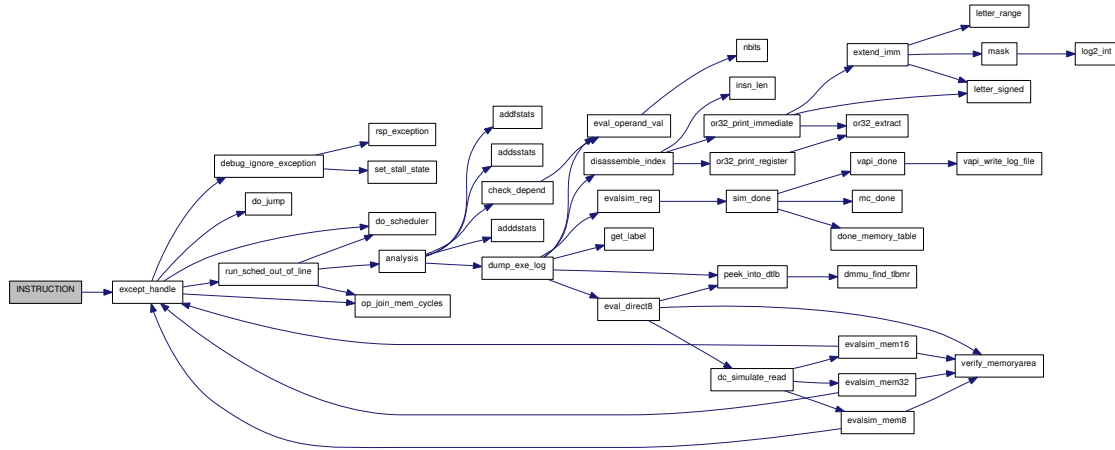
6.42.1.24 INSTRUCTION (l_trap)

Here is the call graph for this function:



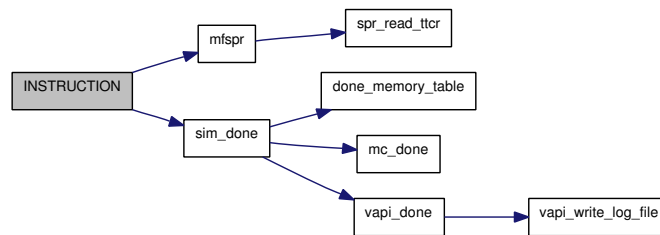
6.42.1.25 INSTRUCTION (l_sys)

Here is the call graph for this function:



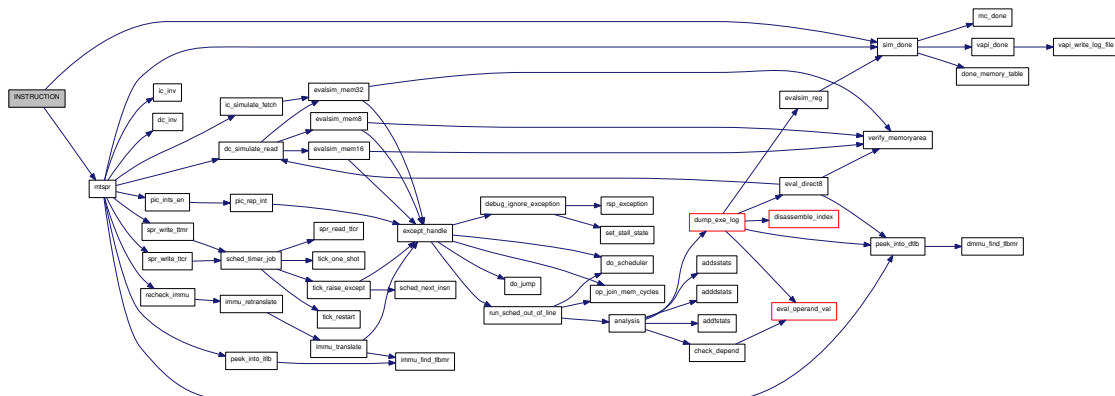
6.42.1.26 INSTRUCTION (l_mfspr)

Here is the call graph for this function:



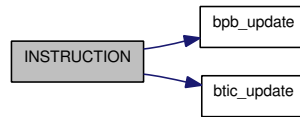
6.42.1.27 INSTRUCTION (l_mtspr)

Here is the call graph for this function:

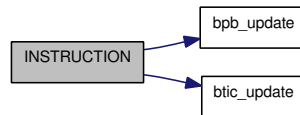


6.42.1.49 INSTRUCTION (l_j)**6.42.1.50 INSTRUCTION (l_bnf)**

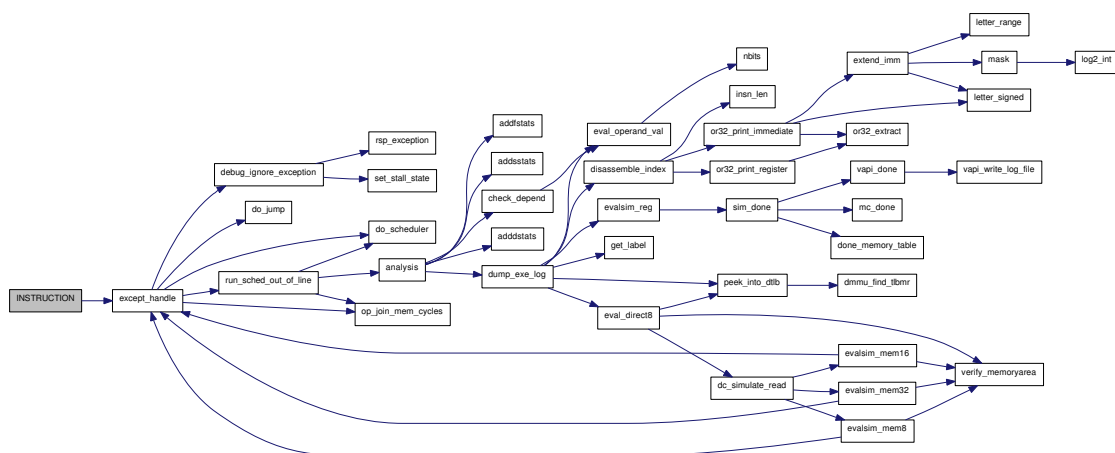
Here is the call graph for this function:

**6.42.1.51 INSTRUCTION (l_bf)**

Here is the call graph for this function:

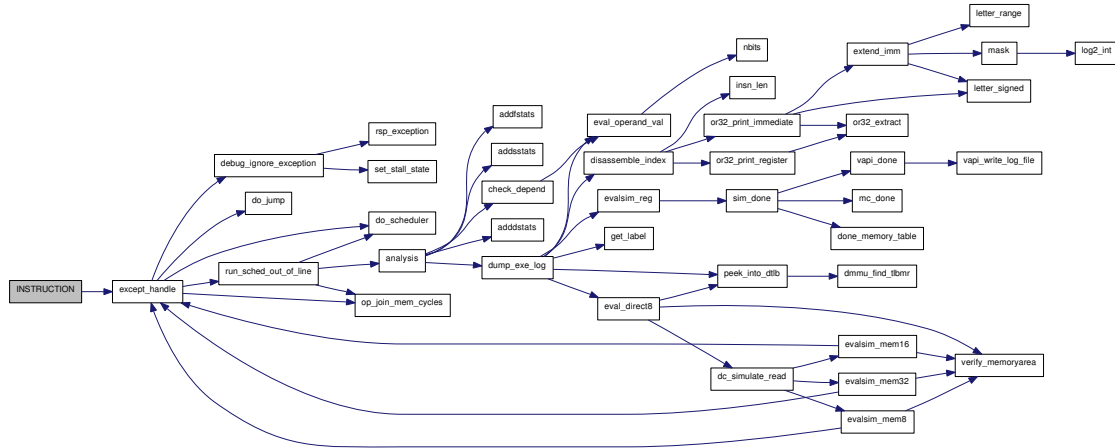
**6.42.1.52 INSTRUCTION (l_sr1)****6.42.1.53 INSTRUCTION (l_sra)****6.42.1.54 INSTRUCTION (l_sl1)****6.42.1.55 INSTRUCTION (l_divu)**

Here is the call graph for this function:



6.42.1.56 INSTRUCTION (l_div)

Here is the call graph for this function:



6.42.1.57 INSTRUCTION (l_mul)

6.42.1.58 INSTRUCTION (l_sub)

6.42.1.59 INSTRUCTION (l_xor)

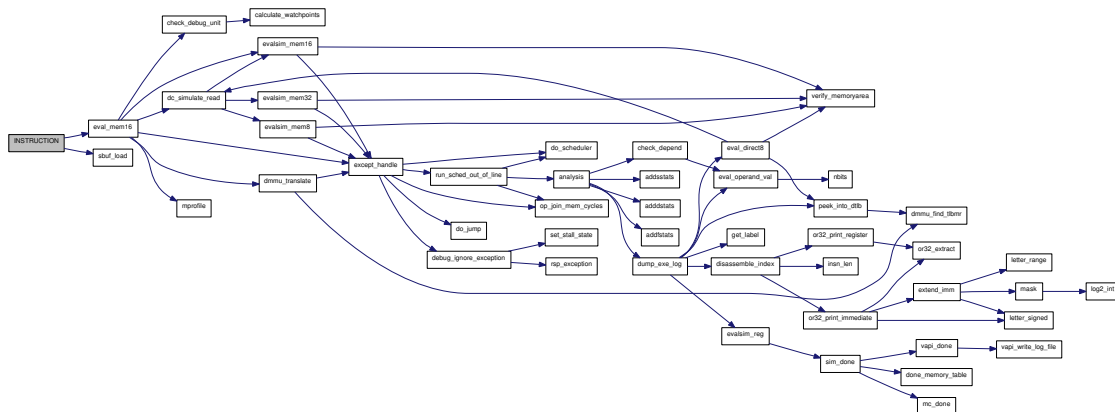
6.42.1.60 INSTRUCTION (l_or)

6.42.1.61 INSTRUCTION (l_and)

6.42.1.62 INSTRUCTION (l_movhi)

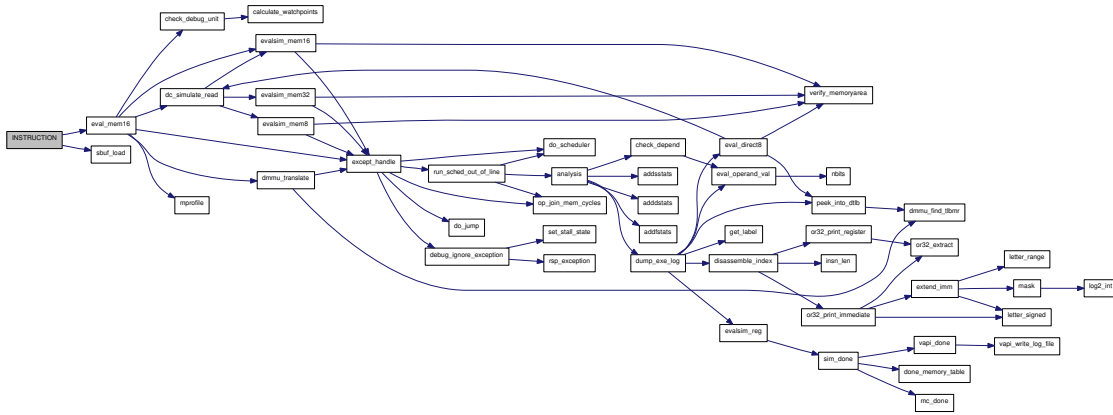
6.42.1.63 INSTRUCTION (l_lhz)

Here is the call graph for this function:



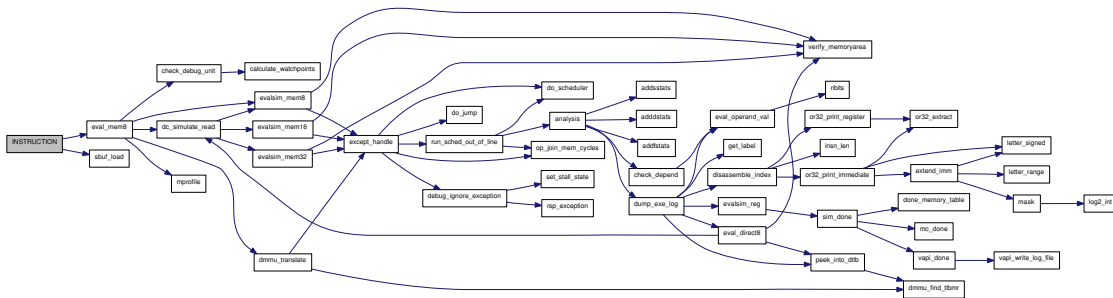
6.42.1.64 INSTRUCTION (l_lhs)

Here is the call graph for this function:



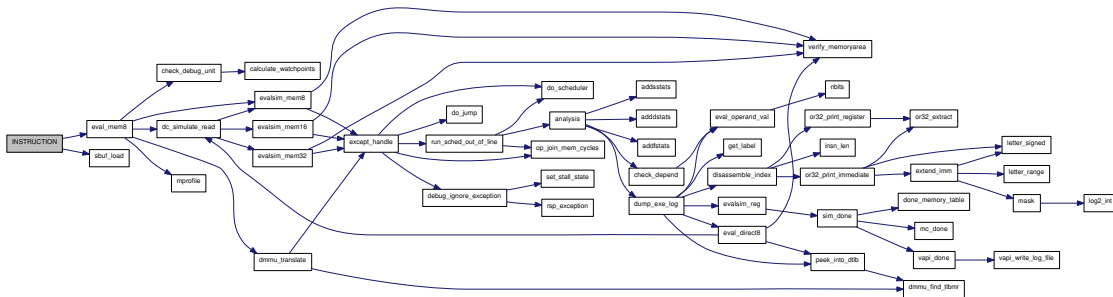
6.42.1.65 INSTRUCTION (l_ibz)

Here is the call graph for this function:



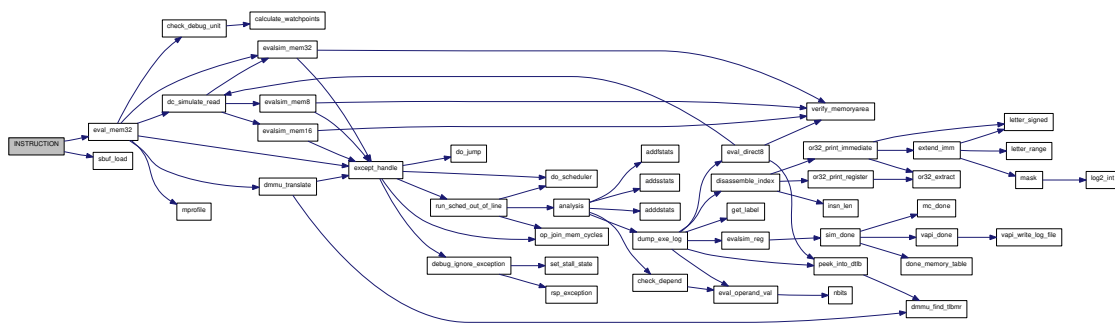
6.42.1.66 INSTRUCTION (l_lbs)

Here is the call graph for this function:



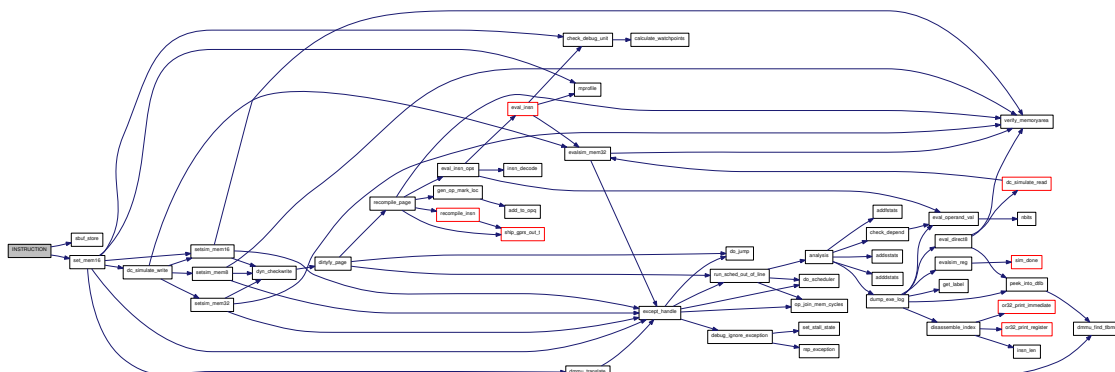
6.42.1.67 INSTRUCTION (l_lwz)

Here is the call graph for this function:



6.42.1.68 INSTRUCTION (l_sh)

Here is the call graph for this function:



6.43 cpu/or32/op-1t-op.h File Reference

Functions

- `__or_dynop void glue (op_imm, T)(void)`
- `__or_dynop void glue (op_clear, T)(void)`
- `__or_dynop void glue (op_check_null_except_delay, T)(void)`
- `__or_dynop void glue (op_check_null_except, T)(void)`
- `__or_dynop void glue (op_calc_insn_ea, T)(void)`
- `__or_dynop void glue (op_macrc, T)(void)`
- `__or_dynop void glue (op_mac_imm, T)(void)`

6.43.1 Function Documentation

6.43.1.1 `__or_dynop void glue (op_mac_imm, T)`

6.43.1.2 `__or_dynop void glue (op_macrc, T)`

6.43.1.3 `__or_dynop void glue (op_calc_insn_ea, T)`

6.43.1.4 `__or_dynop void glue (op_check_null_except, T)`

Here is the call graph for this function:



6.43.1.5 `__or_dynop void glue (op_check_null_except_delay, T)`

Here is the call graph for this function:

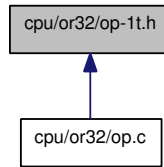


6.43.1.6 `__or_dynop void glue (op_clear, T)`

6.43.1.7 `__or_dynop void glue (op_imm, T)`

6.44 cpu/or32/op-1t.h File Reference

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



Defines

- #define [OP_1T](#)
- #define [T](#) glue(_, T0)

6.44.1 Define Documentation

6.44.1.1 #define OP_1T

6.44.1.2 #define T glue(_, T0)

6.45 cpu/or32/op-2t-op.h File Reference

Functions

- `__or_dynop void glue (op_move, T)(void)`
- `__or_dynop void glue (op_ff1, T)(void)`
- `__or_dynop void glue (op_neg, T)(void)`

6.45.1 Function Documentation

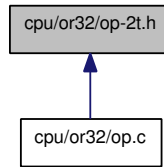
6.45.1.1 `__or_dynop void glue (op_neg, T)`

6.45.1.2 `__or_dynop void glue (op_ff1, T)`

6.45.1.3 `__or_dynop void glue (op_move, T)`

6.46 cpu/or32/op-2t.h File Reference

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



Defines

- #define [OP_2T](#)
- #define [T](#) glue(glue(glue(_, T0), _), T1)

6.46.1 Define Documentation

6.46.1.1 #define OP_2T

6.46.1.2 #define T glue(glue(glue(_, T0), _), T1)

6.47 cpu/or32/op-3t-op.h File Reference

Functions

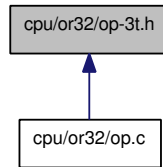
- `__or_dynop void glue (op_cmov, T)(void)`

6.47.1 Function Documentation

6.47.1.1 `__or_dynop void glue (op_cmov, T)`

6.48 cpu/or32/op-3t.h File Reference

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



Defines

- #define [OP_3T](#)
- #define [T](#) glue(glue(glue(glue(glue(_, T0), _), T1), _), T2)

6.48.1 Define Documentation

6.48.1.1 #define OP_3T

6.48.1.2 #define T glue(glue(glue(glue(glue(_, T0), _), T1), _), T2)

6.49 cpu/or32/op-arith-op.h File Reference

6.50 `cpu/or32/op-comp-op.h` File Reference

6.51 cpu/or32/op-extend-op.h File Reference

Functions

- void [glue](#) (glue(op_, EXT_NAME), T)(void)

6.51.1 Function Documentation

6.51.1.1 void glue (glue(op_, EXT_NAME), T)

6.52 cpu/or32/op-ff1-op.h File Reference

Functions

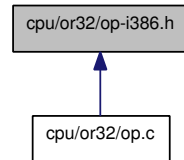
- `__or_dynop void glue (glue(glue(op_ff1_, DST_T), _), SRC_T)(void)`

6.52.1 Function Documentation

6.52.1.1 `__or_dynop void glue (glue(glue(op_ff1_, DST_T), _), SRC_T)`

6.53 cpu/or32/op-i386.h File Reference

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



Defines

- #define `OP_JUMP(x)` `asm("jmp *%0" : : "rm" (x))`
- #define `FORCE_RET` `asm volatile ("")`
- #define `SPEEDY_CALL(func)` `asm("call "#func"\n")`

Functions

- `asm (" .align 2\n"" .p2align 4,,15\n"" .globl op_do_jump\n"" .type op_do_jump,@function\n""op_do_jump:\n"" ret\n"" ret\n""1:\n"" .size op_do_jump,1b-op_do_jump\n")`

6.53.1 Define Documentation

6.53.1.1 #define `FORCE_RET` `asm volatile ("")`

6.53.1.2 #define `OP_JUMP(x)` `asm("jmp *%0" : : "rm" (x))`

6.53.1.3 #define `SPEEDY_CALL(func)` `asm("call "#func"\n")`

6.53.2 Function Documentation

6.53.2.1 `asm (" .align 2\n"" .p2align 4, 15\n"" .globl op_do_jump\n"" .type op_do_jump, @function\n""op_do_jump:\n""ret\n""ret\n""1:\n"" .size op_do_jump, 1b-op_do_jump\n")`

6.54 `cpu/or32/op-lwhb-op.h` File Reference

6.55 cpu/or32/op-mac-op.h File Reference

Functions

- `__or_dynop void glue (glue(op_, OP_NAME), T)(void)`

6.55.1 Function Documentation

6.55.1.1 `__or_dynop void glue (glue(op_, OP_NAME), T)`

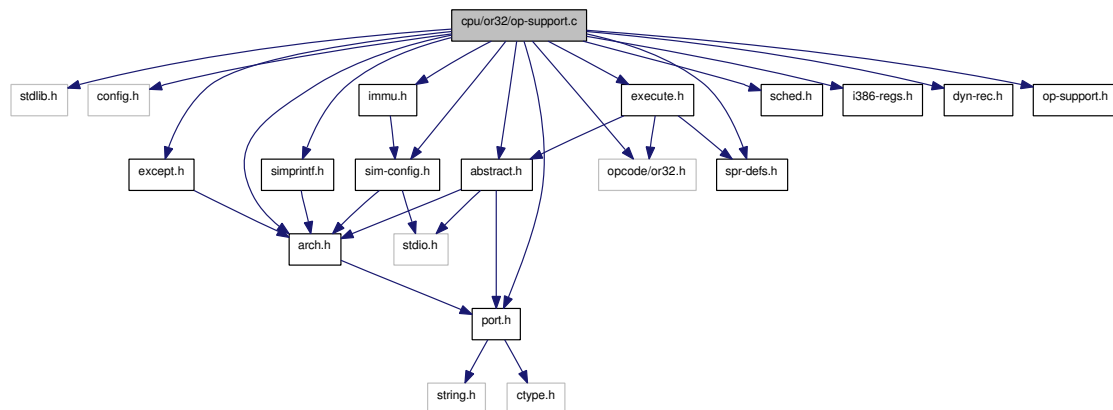
6.57 cpu/or32/op-support.c File Reference

```

#include <stdlib.h>
#include "config.h"
#include "port.h"
#include "arch.h"
#include "opcode/or32.h"
#include "sim-config.h"
#include "spr-defs.h"
#include "except.h"
#include "immu.h"
#include "abstract.h"
#include "execute.h"
#include "sched.h"
#include "i386-regs.h"
#include "dyn-rec.h"
#include "op-support.h"
#include "simprintf.h"

```

Include dependency graph for op-support.c:



Functions

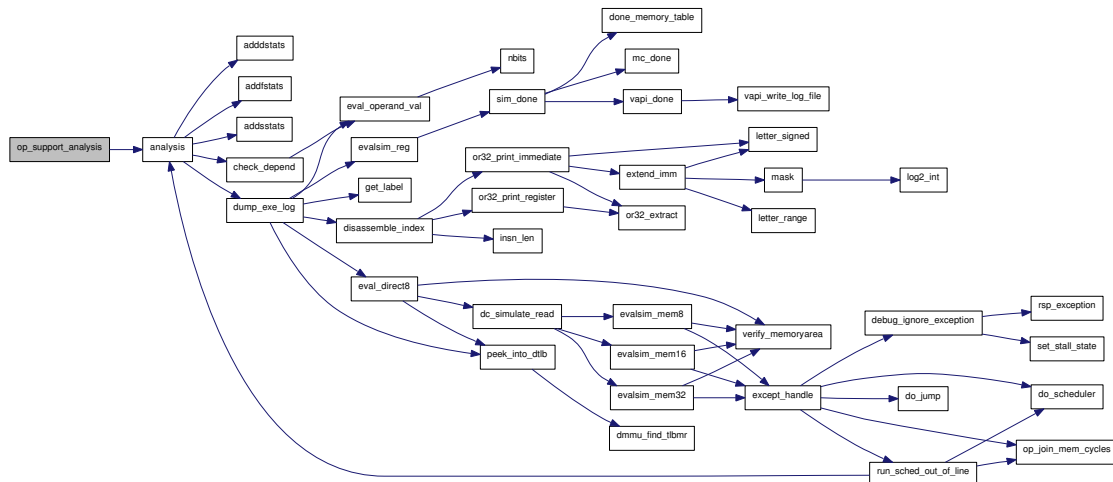
- void [op_support_nop_exit](#) (void)
- void [op_support_nop_reset](#) (void)
- void [op_support_nop_printf](#) (void)
- void [op_support_nop_report](#) (void)
- void [op_support_nop_report_imm](#) (int imm)
- void [do_jump](#) (oraddr_t addr)
- void [op_support_analysis](#) (void)

6.57.1 Function Documentation

6.57.1.1 void do_jump (oraddr_t addr)

6.57.1.2 void op_support_analysis (void)

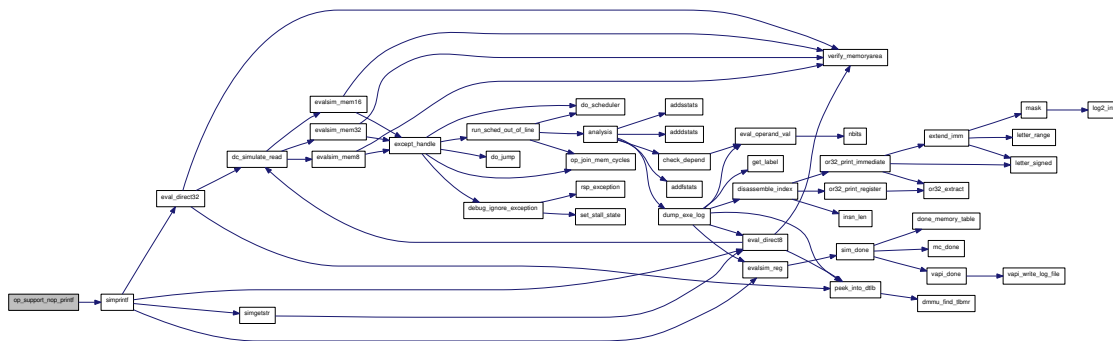
Here is the call graph for this function:



6.57.1.3 void op_support_nop_exit (void)

6.57.1.4 void op_support_nop_printf (void)

Here is the call graph for this function:



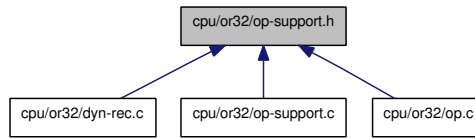
6.57.1.5 void op_support_nop_report (void)

6.57.1.6 void op_support_nop_report_imm (int imm)

6.57.1.7 void op_support_nop_reset (void)

6.58 cpu/or32/op-support.h File Reference

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



Functions

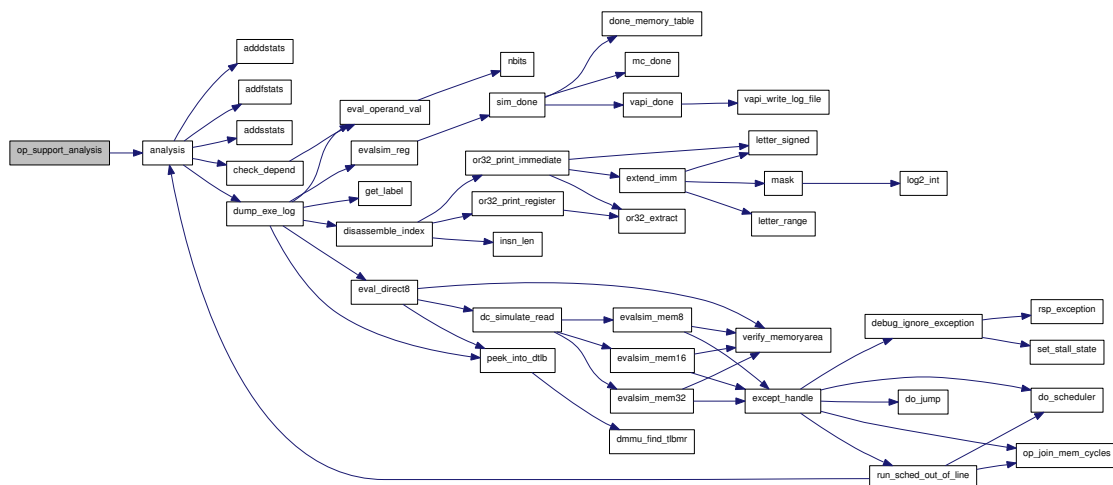
- void [op_support_nop_exit](#) (void)
- void [op_support_nop_reset](#) (void)
- void [op_support_nop_printf](#) (void)
- void [op_support_nop_report](#) (void)
- void [op_support_nop_report_imm](#) (int imm)
- void [op_support_analysis](#) (void)
- void [do_jump](#) (oraddr_t addr)
- void [upd_reg_from_t](#) (oraddr_t pc, int bound)

6.58.1 Function Documentation

6.58.1.1 void do_jump (oraddr_t addr)

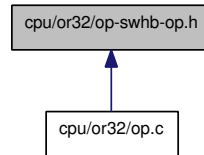
6.58.1.2 void op_support_analysis (void)

Here is the call graph for this function:



6.59 cpu/or32/op-swhb-op.h File Reference

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



Functions

- `__or_dynop void glue (glue(op_, S_OP_NAME), _clear_imm)(void)`

6.59.1 Function Documentation

6.59.1.1 `__or_dynop void glue (glue(op_, S_OP_NAME), _clear_imm)`

6.60 cpu/or32/op-t-reg-mov-op.h File Reference

Functions

- `__or_dynop void glue (op_gtt_gpr1, T)(void)`
- `__or_dynop void glue (op_gtt_gpr2, T)(void)`
- `__or_dynop void glue (op_gtt_gpr3, T)(void)`
- `__or_dynop void glue (op_gtt_gpr4, T)(void)`
- `__or_dynop void glue (op_gtt_gpr5, T)(void)`
- `__or_dynop void glue (op_gtt_gpr6, T)(void)`
- `__or_dynop void glue (op_gtt_gpr7, T)(void)`
- `__or_dynop void glue (op_gtt_gpr8, T)(void)`
- `__or_dynop void glue (op_gtt_gpr9, T)(void)`
- `__or_dynop void glue (op_gtt_gpr10, T)(void)`
- `__or_dynop void glue (op_gtt_gpr11, T)(void)`
- `__or_dynop void glue (op_gtt_gpr12, T)(void)`
- `__or_dynop void glue (op_gtt_gpr13, T)(void)`
- `__or_dynop void glue (op_gtt_gpr14, T)(void)`
- `__or_dynop void glue (op_gtt_gpr15, T)(void)`
- `__or_dynop void glue (op_gtt_gpr16, T)(void)`
- `__or_dynop void glue (op_gtt_gpr17, T)(void)`
- `__or_dynop void glue (op_gtt_gpr18, T)(void)`
- `__or_dynop void glue (op_gtt_gpr19, T)(void)`
- `__or_dynop void glue (op_gtt_gpr20, T)(void)`
- `__or_dynop void glue (op_gtt_gpr21, T)(void)`
- `__or_dynop void glue (op_gtt_gpr22, T)(void)`
- `__or_dynop void glue (op_gtt_gpr23, T)(void)`
- `__or_dynop void glue (op_gtt_gpr24, T)(void)`
- `__or_dynop void glue (op_gtt_gpr25, T)(void)`
- `__or_dynop void glue (op_gtt_gpr26, T)(void)`
- `__or_dynop void glue (op_gtt_gpr27, T)(void)`
- `__or_dynop void glue (op_gtt_gpr28, T)(void)`
- `__or_dynop void glue (op_gtt_gpr29, T)(void)`
- `__or_dynop void glue (op_gtt_gpr30, T)(void)`
- `__or_dynop void glue (op_gtt_gpr31, T)(void)`
- `__or_dynop void glue (op_ttg_gpr1, T)(void)`
- `__or_dynop void glue (op_ttg_gpr2, T)(void)`
- `__or_dynop void glue (op_ttg_gpr3, T)(void)`
- `__or_dynop void glue (op_ttg_gpr4, T)(void)`
- `__or_dynop void glue (op_ttg_gpr5, T)(void)`
- `__or_dynop void glue (op_ttg_gpr6, T)(void)`
- `__or_dynop void glue (op_ttg_gpr7, T)(void)`
- `__or_dynop void glue (op_ttg_gpr8, T)(void)`
- `__or_dynop void glue (op_ttg_gpr9, T)(void)`
- `__or_dynop void glue (op_ttg_gpr10, T)(void)`
- `__or_dynop void glue (op_ttg_gpr11, T)(void)`
- `__or_dynop void glue (op_ttg_gpr12, T)(void)`
- `__or_dynop void glue (op_ttg_gpr13, T)(void)`
- `__or_dynop void glue (op_ttg_gpr14, T)(void)`
- `__or_dynop void glue (op_ttg_gpr15, T)(void)`

- `__or_dynop void glue (op_ttg_gpr16, T)(void)`
- `__or_dynop void glue (op_ttg_gpr17, T)(void)`
- `__or_dynop void glue (op_ttg_gpr18, T)(void)`
- `__or_dynop void glue (op_ttg_gpr19, T)(void)`
- `__or_dynop void glue (op_ttg_gpr20, T)(void)`
- `__or_dynop void glue (op_ttg_gpr21, T)(void)`
- `__or_dynop void glue (op_ttg_gpr22, T)(void)`
- `__or_dynop void glue (op_ttg_gpr23, T)(void)`
- `__or_dynop void glue (op_ttg_gpr24, T)(void)`
- `__or_dynop void glue (op_ttg_gpr25, T)(void)`
- `__or_dynop void glue (op_ttg_gpr26, T)(void)`
- `__or_dynop void glue (op_ttg_gpr27, T)(void)`
- `__or_dynop void glue (op_ttg_gpr28, T)(void)`
- `__or_dynop void glue (op_ttg_gpr29, T)(void)`
- `__or_dynop void glue (op_ttg_gpr30, T)(void)`
- `__or_dynop void glue (op_ttg_gpr31, T)(void)`

6.60.1 Function Documentation

6.60.1.1 `__or_dynop void glue (op_ttg_gpr31, T)`

6.60.1.2 `__or_dynop void glue (op_ttg_gpr30, T)`

6.60.1.3 `__or_dynop void glue (op_ttg_gpr29, T)`

6.60.1.4 `__or_dynop void glue (op_ttg_gpr28, T)`

6.60.1.5 `__or_dynop void glue (op_ttg_gpr27, T)`

6.60.1.6 `__or_dynop void glue (op_ttg_gpr26, T)`

6.60.1.7 `__or_dynop void glue (op_ttg_gpr25, T)`

6.60.1.8 `__or_dynop void glue (op_ttg_gpr24, T)`

6.60.1.9 `__or_dynop void glue (op_ttg_gpr23, T)`

6.60.1.10 `__or_dynop void glue (op_ttg_gpr22, T)`

6.60.1.11 `__or_dynop void glue (op_ttg_gpr21, T)`

6.60.1.12 `__or_dynop void glue (op_ttg_gpr20, T)`

6.60.1.13 `__or_dynop void glue (op_ttg_gpr19, T)`

6.60.1.14 `__or_dynop void glue (op_ttg_gpr18, T)`

6.60.1.15 `__or_dynop void glue (op_ttg_gpr17, T)`

6.60.1.16 `__or_dynop void glue (op_ttg_gpr16, T)`

6.60.1.17 `__or_dynop void glue (op_ttg_gpr15, T)`

6.60.1.18 `__or_dynop void glue (op_ttg_gpr14, T)`

6.60.1.19 `__or_dynop void glue (op_ttg_gpr13, T)`

6.60.1.20 `__or_dynop void glue (op_ttg_gpr12, T)`

6.60.1.21 `__or_dynop void glue (op_ttg_gpr11, T)`

6.60.1.22 `__or_dynop void glue (op_ttg_gpr10, T)`

6.60.1.23 `__or_dynop void glue (op_ttg_gpr9, T)`

6.60.1.24 `__or_dynop void glue (op_ttg_gpr8, T)`

6.60.1.25 `__or_dynop void glue (op_ttg_gpr7, T)`

6.60.1.26 `__or_dynop void glue (op_ttg_gpr6, T)`

6.60.1.27 `__or_dynop void glue (op_ttg_gpr5, T)`

6.60.1.28 `__or_dynop void glue (op_ttg_gpr4, T)`

6.60.1.29 `__or_dynop void glue (op_ttg_gpr3, T)`

6.60.1.30 `__or_dynop void glue (op_ttg_gpr2, T)`

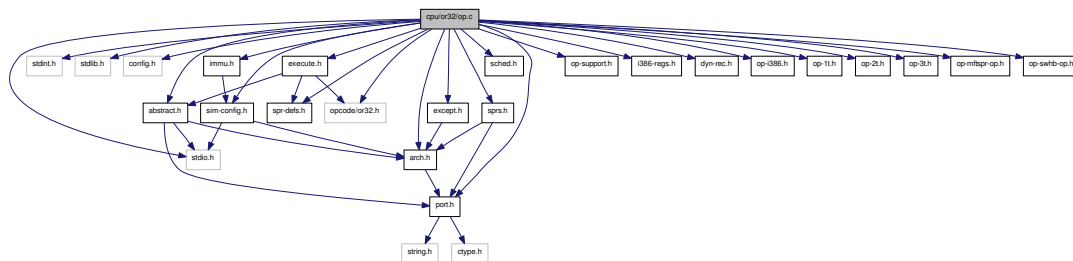
6.61 cpu/or32/op.c File Reference

```

#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include "config.h"
#include "port.h"
#include "arch.h"
#include "spr-defs.h"
#include "opcode/or32.h"
#include "sim-config.h"
#include "except.h"
#include "abstract.h"
#include "execute.h"
#include "sprs.h"
#include "sched.h"
#include "immu.h"
#include "op-support.h"
#include "i386-regs.h"
#include "dyn-rec.h"
#include "op-i386.h"
#include "op-1t.h"
#include "op-2t.h"
#include "op-3t.h"
#include "op-mftspr-op.h"
#include "op-swhb-op.h"

```

Include dependency graph for op.c:



Defines

- `#define __or_dynop`

- #define OP_PARAM1 ((uorreg_t)(&__op_param1))
- #define OP_PARAM2 ((uorreg_t)(&__op_param2))
- #define OP_PARAM3 ((uorreg_t)(&__op_param3))
- #define OP_FILE "op-1t-op.h"
- #define OP_FILE "op-2t-op.h"
- #define OP_FILE "op-3t-op.h"
- #define OP_FILE "op-arith-op.h"
- #define OP_EXTRA
- #define OP /
- #define OP_CAST(x) (orreg_t)(x)
- #define OP_NAME div
- #define OP /
- #define OP_CAST(x) (x)
- #define OP_NAME divu
- #define OP *
- #define OP_CAST(x) (x)
- #define OP_NAME mulu
- #define OP -
- #define OP_CAST(x) (orreg_t)(x)
- #define OP_NAME sub
- #define OP_EXTRA + ((env → sprs[SPR_SR] & SPR_SR_CY) >> 10)
- #define OP +
- #define OP_CAST(x) (orreg_t)(x)
- #define OP_NAME addc
- #define OP_EXTRA
- #define OP +
- #define OP_CAST(x) (orreg_t)(x)
- #define OP_NAME add
- #define OP &
- #define OP_CAST(x) (x)
- #define OP_NAME and
- #define OP *
- #define OP_CAST(x) (orreg_t)(x)
- #define OP_NAME mul
- #define OP |
- #define OP_CAST(x) (x)
- #define OP_NAME or
- #define OP <<
- #define OP_CAST(x) (x)
- #define OP_NAME sll
- #define OP >>
- #define OP_CAST(x) (orreg_t)(x)
- #define OP_NAME sra
- #define OP >>
- #define OP_CAST(x) (x)
- #define OP_NAME srl
- #define OP ^
- #define OP_CAST(x) (x)
- #define OP_NAME xor
- #define OP_FILE "op-extend-op.h"

- #define EXT_NAME extbs
- #define EXT_TYPE int8_t
- #define EXT_CAST (orreg_t)
- #define EXT_NAME extbz
- #define EXT_TYPE uint8_t
- #define EXT_CAST (uorreg_t)
- #define EXT_NAME exths
- #define EXT_TYPE int16_t
- #define EXT_CAST (orreg_t)
- #define EXT_NAME exthz
- #define EXT_TYPE uint16_t
- #define EXT_CAST (uorreg_t)
- #define OP_FILE "op-comp-op.h"
- #define COMP ==
- #define COMP_NAME sfeq
- #define COMP_CAST(x) (x)
- #define COMP !=
- #define COMP_NAME sfne
- #define COMP_CAST(x) (x)
- #define COMP >
- #define COMP_NAME sfgtu
- #define COMP_CAST(x) (x)
- #define COMP >=
- #define COMP_NAME sfgeu
- #define COMP_CAST(x) (x)
- #define COMP <
- #define COMP_NAME sfltu
- #define COMP_CAST(x) (x)
- #define COMP <=
- #define COMP_NAME sfleu
- #define COMP_CAST(x) (x)
- #define COMP >
- #define COMP_NAME sfgts
- #define COMP_CAST(x) (orreg_t)(x)
- #define COMP >=
- #define COMP_NAME sfges
- #define COMP_CAST(x) (orreg_t)(x)
- #define COMP <
- #define COMP_NAME sflts
- #define COMP_CAST(x) (orreg_t)(x)
- #define COMP <=
- #define COMP_NAME sfles
- #define COMP_CAST(x) (orreg_t)(x)
- #define OP_FILE "op-t-reg-mov-op.h"
- #define OP_FILE "op-mftspr-op.h"
- #define OP_FILE "op-mac-op.h"
- #define OP +=
- #define OP_NAME mac
- #define OP -=
- #define OP_NAME msb

- #define `OP_FILE` "op-lwhb-op.h"
- #define `LS_OP_NAME` lbz
- #define `LS_OP_CAST`
- #define `LS_OP_FUNC` eval_mem8
- #define `LS_OP_NAME` lbs
- #define `LS_OP_CAST` (int8_t)
- #define `LS_OP_FUNC` eval_mem8
- #define `LS_OP_NAME` lhz
- #define `LS_OP_CAST`
- #define `LS_OP_FUNC` eval_mem16
- #define `LS_OP_NAME` lhs
- #define `LS_OP_CAST` (int16_t)
- #define `LS_OP_FUNC` eval_mem16
- #define `LS_OP_NAME` lwz
- #define `LS_OP_CAST`
- #define `LS_OP_FUNC` eval_mem32
- #define `LS_OP_NAME` lws
- #define `LS_OP_CAST` (int32_t)
- #define `LS_OP_FUNC` eval_mem32
- #define `OP_FILE` "op-swhb-op.h"
- #define `S_OP_NAME` sb
- #define `S_FUNC` set_mem8
- #define `S_OP_NAME` sh
- #define `S_FUNC` set_mem16
- #define `S_OP_NAME` sw
- #define `S_FUNC` set_mem32

Functions

- register struct `cpu_state` *env `asm` (CPU_STATE_REG)
- register uint32_t t0 `asm` (T0_REG)
- register uint32_t t1 `asm` (T1_REG)
- register uint32_t t2 `asm` (T2_REG)
- static void `save_t_bound` (oraddr_t pc)
- void `do_sched_wrap` (void)
- void `do_sched_wrap_delay` (void)
- void `enter_dyn_code` (oraddr_t addr, struct `dyn_page` *dp)
- __or_dynop void `op_set_pc_pc_delay` (void)
- __or_dynop void `op_set_pc_delay_imm` (void)
- __or_dynop void `op_set_pc_delay_pc` (void)
- __or_dynop void `op_clear_pc_delay` (void)
- __or_dynop void `op_do_jump_delay` (void)
- __or_dynop void `op_clear_delay_insn` (void)
- __or_dynop void `op_set_delay_insn` (void)
- __or_dynop void `op_check_delay_slot` (void)
- __or_dynop void `op_jump_imm` (void)
- __or_dynop void `op_set_flag` (void)
- __or_dynop void `op_clear_flag` (void)
- __or_dynop void `op_check_flag` (void)
- __or_dynop void `op_check_flag_delay` (void)

- `__or_dynop` void `op_check_not_flag` (void)
- `__or_dynop` void `op_check_not_flag_delay` (void)
- `__or_dynop` void `op_add_pc` (void)
- `__or_dynop` void `op_nop_exit` (void)
- `__or_dynop` void `op_nop_reset` (void)
- `__or_dynop` void `op_nop_printf` (void)
- `__or_dynop` void `op_nop_report` (void)
- `__or_dynop` void `op_nop_report_imm` (void)
- `__or_dynop` void `op_analysis` (void)
- `__or_dynop` void `op_move_gpr1_pc_delay` (void)
- `__or_dynop` void `op_move_gpr2_pc_delay` (void)
- `__or_dynop` void `op_move_gpr3_pc_delay` (void)
- `__or_dynop` void `op_move_gpr4_pc_delay` (void)
- `__or_dynop` void `op_move_gpr5_pc_delay` (void)
- `__or_dynop` void `op_move_gpr6_pc_delay` (void)
- `__or_dynop` void `op_move_gpr7_pc_delay` (void)
- `__or_dynop` void `op_move_gpr8_pc_delay` (void)
- `__or_dynop` void `op_move_gpr9_pc_delay` (void)
- `__or_dynop` void `op_move_gpr10_pc_delay` (void)
- `__or_dynop` void `op_move_gpr11_pc_delay` (void)
- `__or_dynop` void `op_move_gpr12_pc_delay` (void)
- `__or_dynop` void `op_move_gpr13_pc_delay` (void)
- `__or_dynop` void `op_move_gpr14_pc_delay` (void)
- `__or_dynop` void `op_move_gpr15_pc_delay` (void)
- `__or_dynop` void `op_move_gpr16_pc_delay` (void)
- `__or_dynop` void `op_move_gpr17_pc_delay` (void)
- `__or_dynop` void `op_move_gpr18_pc_delay` (void)
- `__or_dynop` void `op_move_gpr19_pc_delay` (void)
- `__or_dynop` void `op_move_gpr20_pc_delay` (void)
- `__or_dynop` void `op_move_gpr21_pc_delay` (void)
- `__or_dynop` void `op_move_gpr22_pc_delay` (void)
- `__or_dynop` void `op_move_gpr23_pc_delay` (void)
- `__or_dynop` void `op_move_gpr24_pc_delay` (void)
- `__or_dynop` void `op_move_gpr25_pc_delay` (void)
- `__or_dynop` void `op_move_gpr26_pc_delay` (void)
- `__or_dynop` void `op_move_gpr27_pc_delay` (void)
- `__or_dynop` void `op_move_gpr28_pc_delay` (void)
- `__or_dynop` void `op_move_gpr29_pc_delay` (void)
- `__or_dynop` void `op_move_gpr30_pc_delay` (void)
- `__or_dynop` void `op_move_gpr31_pc_delay` (void)
- `__or_dynop` void `op_join_mem_cycles` (void)
- `__or_dynop` void `op_store_link_addr_gpr` (void)
- `__or_dynop` void `op_prep_rfe` (void)
- static void `prep_except` (oraddr_t epcr_base)
- `__or_dynop` void `op_prep_sys_delay` (void)
- `__or_dynop` void `op_prep_sys` (void)
- `__or_dynop` void `op_prep_trap_delay` (void)
- `__or_dynop` void `op_prep_trap` (void)
- `__or_dynop` void `op_illegal_delay` (void)
- `__or_dynop` void `op_illegal` (void)

- `__or_dynop` void [op_do_sched](#) (void)
- `__or_dynop` void [op_do_sched_delay](#) (void)
- `__or_dynop` void [op_macc](#) (void)
- `__or_dynop` void [op_store_insn_ea](#) (void)

Variables

- `uorreg_t` [__op_param1](#)
- `uorreg_t` [__op_param2](#)
- `uorreg_t` [__op_param3](#)

6.61.1 Define Documentation

6.61.1.1 `#define __or_dynop`

6.61.1.2 `#define COMP <=`

6.61.1.3 `#define COMP <`

6.61.1.4 `#define COMP >=`

6.61.1.5 `#define COMP >`

6.61.1.6 `#define COMP <=`

6.61.1.7 `#define COMP <`

6.61.1.8 `#define COMP >=`

6.61.1.9 `#define COMP >`

6.61.1.10 `#define COMP !=`

6.61.1.11 `#define COMP ==`

6.61.1.12 `#define COMP_CAST(x) (orreg_t)(x)`

6.61.1.13 `#define COMP_CAST(x) (orreg_t)(x)`

6.61.1.14 `#define COMP_CAST(x) (orreg_t)(x)`

6.61.1.15 `#define COMP_CAST(x) (orreg_t)(x)`

6.61.1.16 `#define COMP_CAST(x) (x)`

6.61.1.17 `#define COMP_CAST(x) (x)`

6.61.1.18 `#define COMP_CAST(x) (x)`

6.61.1.19 `#define COMP_CAST(x) (x)`

6.61.1.20 `#define COMP_CAST(x) (x)`

6.61.1.21 `#define COMP_CAST(x) (x)`

6.61.1.22 `#define COMP_NAME sfles`

6.61.1.23 `#define COMP_NAME sflts`

6.61.1.24 `#define COMP_NAME sfges`

6.61.1.25 `#define COMP_NAME sfgts`

6.61.1.26 `#define COMP_NAME sfleu`

6.61.1.27 `#define COMP_NAME sfltu`
Generated on Tue Nov 11 11:50:04 2008 for OpenRISC: The OpenRISC 1000 Architectural Simulator by Doxygen

6.61.1.28 `#define COMP_NAME sfgeu`

6.61.1.29 `#define COMP_NAME sfgtu`

6.61.1.30 `#define COMP_NAME sfne`

6.61.2.6 void do_sched_wrap_delay (void)

Here is the call graph for this function:

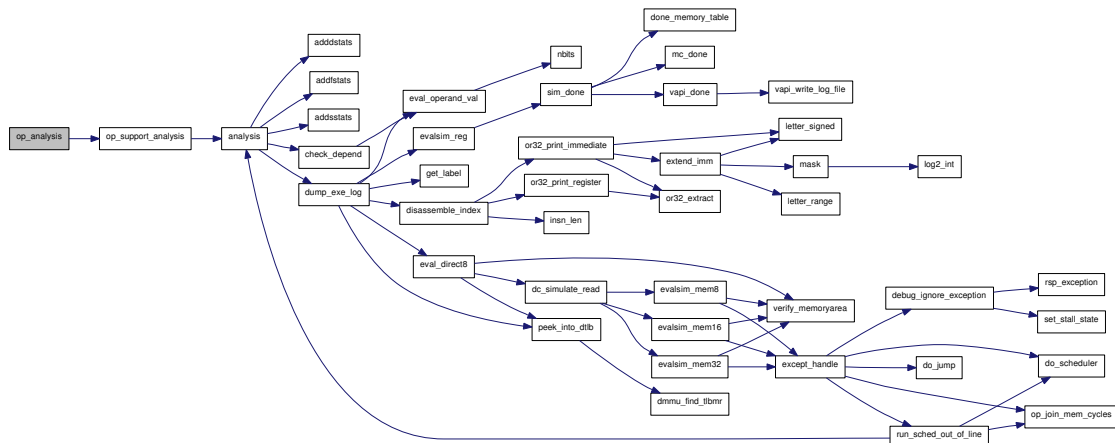


6.61.2.7 void enter_dyn_code (oraddr_t addr, struct dyn_page * dp)

6.61.2.8 __or_dynop void op_add_pc (void)

6.61.2.9 __or_dynop void op_analysis (void)

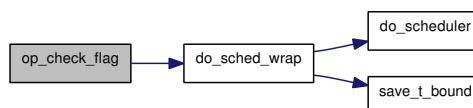
Here is the call graph for this function:



6.61.2.10 __or_dynop void op_check_delay_slot (void)

6.61.2.11 __or_dynop void op_check_flag (void)

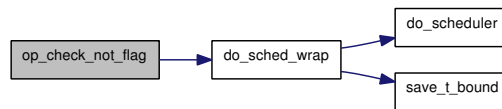
Here is the call graph for this function:



6.61.2.12 `__or_dynop void op_check_flag_delay (void)`

6.61.2.13 `__or_dynop void op_check_not_flag (void)`

Here is the call graph for this function:



6.61.2.14 `__or_dynop void op_check_not_flag_delay (void)`

6.61.2.15 `__or_dynop void op_clear_delay_insn (void)`

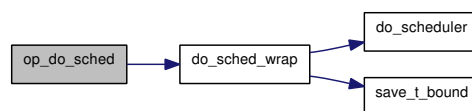
6.61.2.16 `__or_dynop void op_clear_flag (void)`

6.61.2.17 `__or_dynop void op_clear_pc_delay (void)`

6.61.2.18 `__or_dynop void op_do_jump_delay (void)`

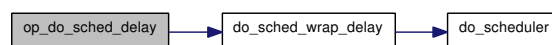
6.61.2.19 `__or_dynop void op_do_sched (void)`

Here is the call graph for this function:



6.61.2.20 `__or_dynop void op_do_sched_delay (void)`

Here is the call graph for this function:



- 6.61.2.21 `__or_dynop void op_illegal (void)`
- 6.61.2.22 `__or_dynop void op_illegal_delay (void)`
- 6.61.2.23 `__or_dynop void op_jump_imm (void)`
- 6.61.2.24 `__or_dynop void op_join_mem_cycles (void)`
- 6.61.2.25 `__or_dynop void op_macc (void)`
- 6.61.2.26 `__or_dynop void op_move_gpr10_pc_delay (void)`
- 6.61.2.27 `__or_dynop void op_move_gpr11_pc_delay (void)`
- 6.61.2.28 `__or_dynop void op_move_gpr12_pc_delay (void)`
- 6.61.2.29 `__or_dynop void op_move_gpr13_pc_delay (void)`
- 6.61.2.30 `__or_dynop void op_move_gpr14_pc_delay (void)`
- 6.61.2.31 `__or_dynop void op_move_gpr15_pc_delay (void)`
- 6.61.2.32 `__or_dynop void op_move_gpr16_pc_delay (void)`
- 6.61.2.33 `__or_dynop void op_move_gpr17_pc_delay (void)`
- 6.61.2.34 `__or_dynop void op_move_gpr18_pc_delay (void)`
- 6.61.2.35 `__or_dynop void op_move_gpr19_pc_delay (void)`
- 6.61.2.36 `__or_dynop void op_move_gpr1_pc_delay (void)`
- 6.61.2.37 `__or_dynop void op_move_gpr20_pc_delay (void)`
- 6.61.2.38 `__or_dynop void op_move_gpr21_pc_delay (void)`
- 6.61.2.39 `__or_dynop void op_move_gpr22_pc_delay (void)`
- 6.61.2.40 `__or_dynop void op_move_gpr23_pc_delay (void)`
- 6.61.2.41 `__or_dynop void op_move_gpr24_pc_delay (void)`
- 6.61.2.42 `__or_dynop void op_move_gpr25_pc_delay (void)`
- 6.61.2.43 `__or_dynop void op_move_gpr26_pc_delay (void)`
- 6.61.2.44 `__or_dynop void op_move_gpr27_pc_delay (void)`
- 6.61.2.45 `__or_dynop void op_move_gpr28_pc_delay (void)`
- 6.61.2.46 `__or_dynop void op_move_gpr29_pc_delay (void)`
- 6.61.2.47 `__or_dynop void op_move_gpr2_pc_delay (void)`
- 6.61.2.48 `__or_dynop void op_move_gpr30_pc_delay (void)`

Generated on Tue, Nov 11, 2008 10:04:20 AM for OpenRISC 1000 Functional Simulator by Doxygen

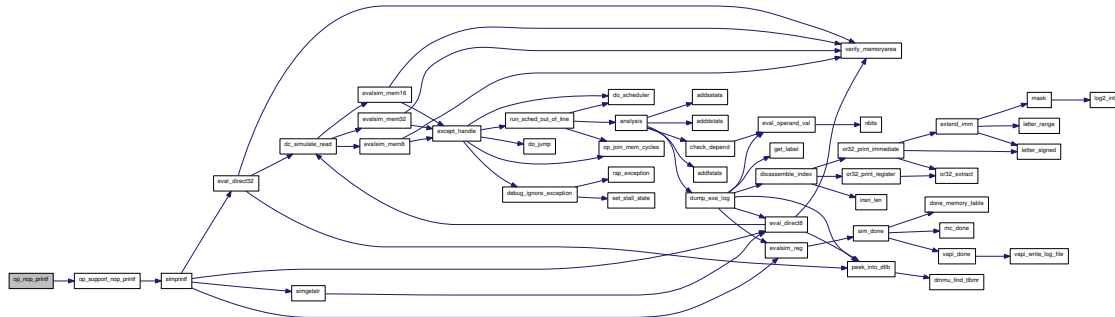
6.61.2.49 `__or_dynop void op_move_gpr31_pc_delay (void)`

6.61.2.50 `__or_dynop void op_move_gpr3_pc_delay (void)`

6.61.2.51 `__or_dynop void op_move_gpr4_pc_delay (void)`

6.61.2.58 __or_dynop void op_nop_printf (void)

Here is the call graph for this function:



6.61.2.59 __or_dynop void op_nop_report (void)

Here is the call graph for this function:



6.61.2.60 __or_dynop void op_nop_report_imm (void)

Here is the call graph for this function:



6.61.2.61 __or_dynop void op_nop_reset (void)

Here is the call graph for this function:



6.61.2.62 __or_dynop void op_prep_rfe (void)

6.61.2.63 __or_dynop void op_prep_sys (void)

Here is the call graph for this function:



6.61.2.64 `__or_dynop void op_prep_sys_delay (void)`

Here is the call graph for this function:

**6.61.2.65** `__or_dynop void op_prep_trap (void)`

Here is the call graph for this function:

**6.61.2.66** `__or_dynop void op_prep_trap_delay (void)`

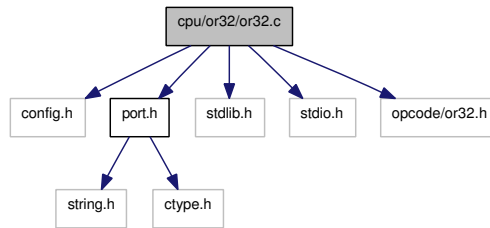
Here is the call graph for this function:

**6.61.2.67** `__or_dynop void op_set_delay_insn (void)`**6.61.2.68** `__or_dynop void op_set_flag (void)`**6.61.2.69** `__or_dynop void op_set_pc_delay_imm (void)`**6.61.2.70** `__or_dynop void op_set_pc_delay_pc (void)`**6.61.2.71** `__or_dynop void op_set_pc_pc_delay (void)`**6.61.2.72** `__or_dynop void op_store_insn_ea (void)`**6.61.2.73** `__or_dynop void op_store_link_addr_gpr (void)`**6.61.2.74** `static void prep_except (oraddr_t epcr_base) [inline, static]`**6.61.2.75** `static void save_t_bound (oraddr_t pc) [inline, static]`**6.61.3 Variable Documentation****6.61.3.1** `uorreg_t __op_param1`**6.61.3.2** `uorreg_t __op_param2`**6.61.3.3** `uorreg_t __op_param3`

6.62 cpu/or32/or32.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include "opcode/or32.h"
```

Include dependency graph for or32.c:



Defines

- #define [EFN](#) &l_none
- #define [EF\(func\)](#) EFN
- #define [EFI](#) EFN
- #define [MAX_AUTOMATA_SIZE](#) (1200)
- #define [MAX_OP_TABLE_SIZE](#) (1200)
- #define [MAX_LEN](#) (8)
- #define [MIN\(x, y\)](#) ((x) < (y) ? (x) : (y))

Functions

- int [insn_len](#) (int [insn_index](#))
- int [letter_signed](#) (char l)
- int [letter_range](#) (char l)
- int [insn_index](#) (char *[insn](#))
- CONST char * [insn_name](#) (int [index](#))
- void [l_none](#) ()
- unsigned long [insn_extract](#) (char [param_ch](#), char *[enc_initial](#))
- static void [or32_debug](#) (int [level](#), const char *[format](#),...)
- static unsigned long * [cover_insn](#) (unsigned long *[cur](#), int [pass](#), unsigned int [mask](#))
- static int [num_ones](#) (unsigned long [value](#))
- static struct [insn_op_struct](#) * [parse_params](#) (CONST struct [or32_opcode](#) *[opcode](#), struct [insn_op_struct](#) *[cur](#))
- void [build_automata](#) ()
- void [destruct_automata](#) ()
- int [insn_decode](#) (unsigned int [insn](#))
- unsigned long [extend_imm](#) (unsigned long [imm](#), char l)
- unsigned long [or32_extract](#) (char [param_ch](#), char *[enc_initial](#), unsigned long [insn](#))

- static char * [or32_print_register](#) (char *dest, char param_ch, char *encoding, unsigned long [insn](#))
- static char * [or32_print_immediate](#) (char *dest, char param_ch, char *encoding, unsigned long [insn](#))
- int [disassemble_insn](#) (unsigned long [insn](#))
- int [disassemble_index](#) (unsigned long [insn](#), int index)

Variables

- CONST struct or32_letter [or32_letters](#) []
- CONST struct or32_opcode [or32_opcodes](#) []
- CONST int [num_opcodes](#)
- static int [range_cache](#) [256] = { 0 }
- unsigned long * [automata](#)
- int [nuncovered](#)
- int [curpass](#) = 0
- struct temp_insn_struct * [ti](#)
- struct insn_op_struct * [op_data](#)
- struct insn_op_struct ** [op_start](#)
- static char [disassembled_str](#) [50]
- char * [disassembled](#) = &[disassembled_str](#)[0]

6.62.1 Define Documentation

6.62.1.1 `#define EF(func) EFN`

6.62.1.2 `#define EFI EFN`

6.62.1.3 `#define EFN &l_none`

6.62.1.4 `#define MAX_AUTOMATA_SIZE (1200)`

6.62.1.5 `#define MAX_LEN (8)`

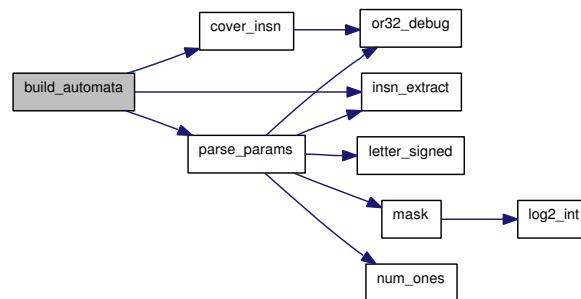
6.62.1.6 `#define MAX_OP_TABLE_SIZE (1200)`

6.62.1.7 `#define MIN(x, y) ((x) < (y) ? (x) : (y))`

6.62.2 Function Documentation

6.62.2.1 `void build_automata ()`

Here is the call graph for this function:



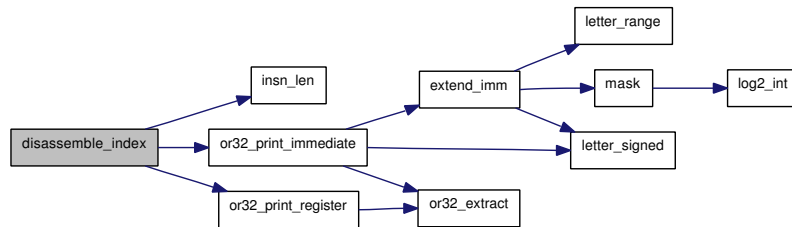
6.62.2.2 `static unsigned long* cover_insn (unsigned long * cur, int pass, unsigned int mask)`
`[static]`

Here is the call graph for this function:

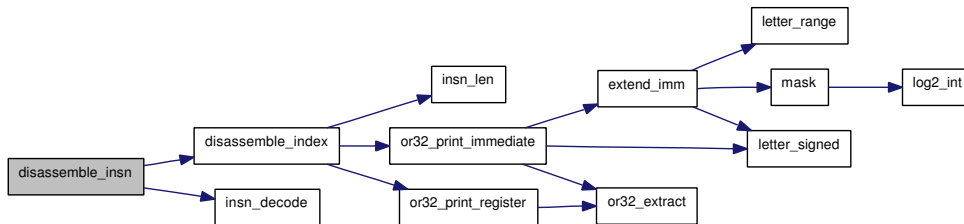


6.62.2.3 void destruct_automata ()**6.62.2.4 int disassemble_index (unsigned long *insn*, int *index*)**

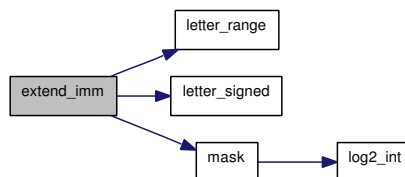
Here is the call graph for this function:

**6.62.2.5 int disassemble_insn (unsigned long *insn*)**

Here is the call graph for this function:

**6.62.2.6 unsigned long extend_imm (unsigned long *imm*, char *l*)**

Here is the call graph for this function:



6.62.2.7 int `insn_decode` (unsigned int *insn*)

6.62.2.8 unsigned long `insn_extract` (char *param_ch*, char * *enc_initial*)

6.62.2.9 int `insn_index` (char * *insn*)

6.62.2.10 int `insn_len` (int *insn_index*)

6.62.2.11 CONST char* `insn_name` (int *index*)

6.62.2.12 void `l_none` ()

6.62.2.13 int `letter_range` (char *l*)

6.62.2.14 int `letter_signed` (char *l*)

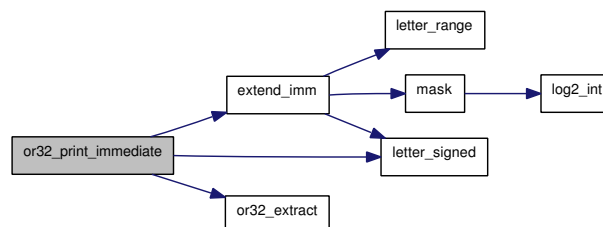
6.62.2.15 static int `num_ones` (unsigned long *value*) [static]

6.62.2.16 static void `or32_debug` (int *level*, const char * *format*, ...) [static]

6.62.2.17 unsigned long `or32_extract` (char *param_ch*, char * *enc_initial*, unsigned long *insn*)

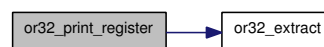
6.62.2.18 static char* `or32_print_immediate` (char * *dest*, char *param_ch*, char * *encoding*, unsigned long *insn*) [static]

Here is the call graph for this function:



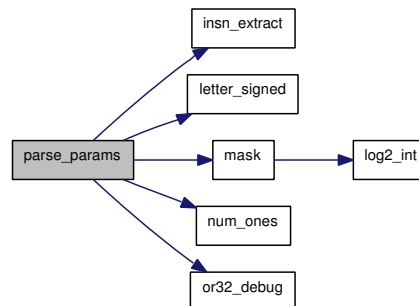
6.62.2.19 static char* `or32_print_register` (char * *dest*, char *param_ch*, char * *encoding*, unsigned long *insn*) [static]

Here is the call graph for this function:



6.62.2.20 `static struct insn_op_struct* parse_params (CONST struct or32_opcode * opcode, struct insn_op_struct * cur)` [static, read]

Here is the call graph for this function:



6.62.3 Variable Documentation

6.62.3.1 `unsigned long* automata`

6.62.3.2 `int curpass = 0`

6.62.3.3 `char* disassembled = &disassembled_str[0]`

6.62.3.4 `char disassembled_str[50]` [static]

6.62.3.5 `CONST int num_opcodes`

Initial value:

```
((sizeof (or32_opcodes)) / (sizeof (struct or32_opcode))) - 1
```

6.62.3.6 `int nuncovered`

6.62.3.7 `struct insn_op_struct* op_data`

6.62.3.8 `struct insn_op_struct ** op_start`

6.62.3.9 `CONST struct or32_letter or32_letters[]`

Initial value:

```
{
  {'A', NUM_UNSIGNED},
  {'B', NUM_UNSIGNED},
  {'D', NUM_UNSIGNED},
  {'I', NUM_SIGNED},
  {'K', NUM_UNSIGNED},
  {'L', NUM_UNSIGNED},
  {'N', NUM_SIGNED},
  {'O', NUM_UNSIGNED},
  {'\0', 0}
}
```

6.62.3.10 `CONST struct or32_opcode or32_opcodes[]`

6.62.3.11 `int range_cache[256] = { 0 } [static]`

6.62.3.12 `struct temp_insn_struct* ti`

6.63 cpu/or32/rec-i386.h File Reference

Functions

- static void * [get_sp](#) (void)

6.63.1 Function Documentation

6.63.1.1 static void* [get_sp](#) (void) [static]

6.64 `cpu/or32/sched-i386.h` File Reference

Functions

- static void [set_sched_cycle](#) (int32_t job_time)

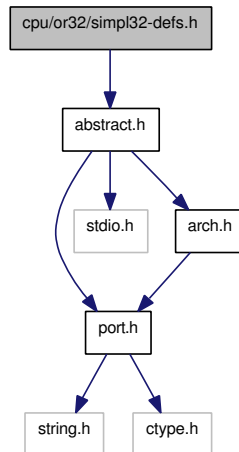
6.64.1 Function Documentation

6.64.1.1 static void `set_sched_cycle` (int32_t *job_time*) [static]

6.65 cpu/or32/simpl32-defs.h File Reference

```
#include "abstract.h"
```

Include dependency graph for simpl32-defs.h:



Functions

- void `l_invalid` PARAMS ((struct `iqueue_entry` *))
- void `l_sfgeu` PARAMS ()(struct `iqueue_entry` *)

6.65.1 Function Documentation

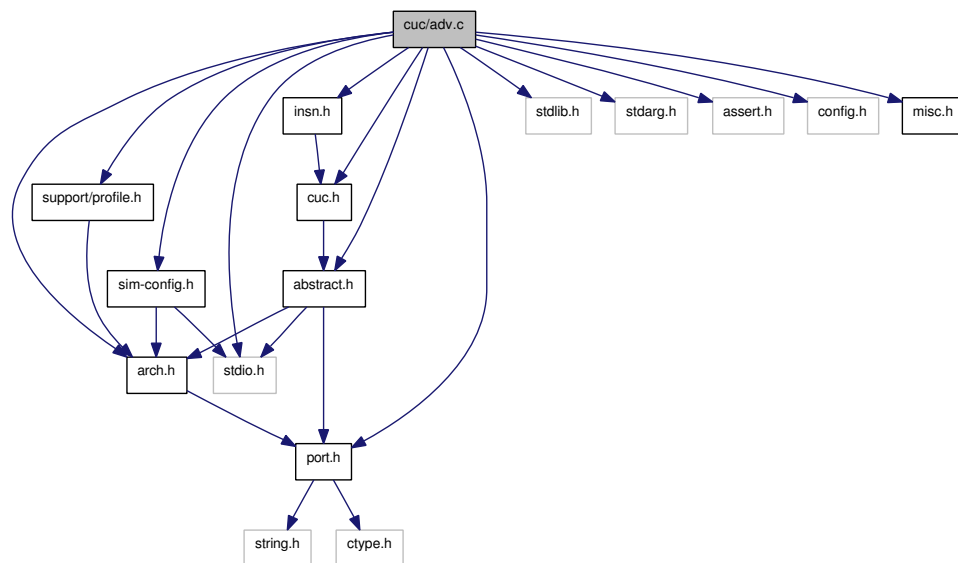
6.65.1.1 void `l_sfgeu` PARAMS ()

6.65.1.2 void `l_none` PARAMS ((struct `iqueue_entry` *))

6.66 cuc/adv.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <assert.h>
#include "config.h"
#include "port.h"
#include "arch.h"
#include "sim-config.h"
#include "abstract.h"
#include "cuc.h"
#include "insn.h"
#include "support/profile.h"
#include "misc.h"
```

Include dependency graph for adv.c:



Functions

- static void [mark_successors](#) ([cuc_func](#) *f, int b, int m, int stopb)
- static unsigned long [mask](#) (unsigned long c)
- void [insert_conditional_facts](#) ([cuc_func](#) *f)
- static unsigned long [max_op](#) ([cuc_func](#) *f, int ref, int o)
- static unsigned long [calc_max](#) ([cuc_func](#) *f, int ref)
- void [detect_max_values](#) ([cuc_func](#) *f)

6.66.1 Function Documentation

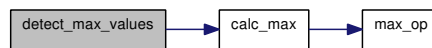
6.66.1.1 `static unsigned long calc_max (cuc_func *f, int ref)` [static]

Here is the call graph for this function:



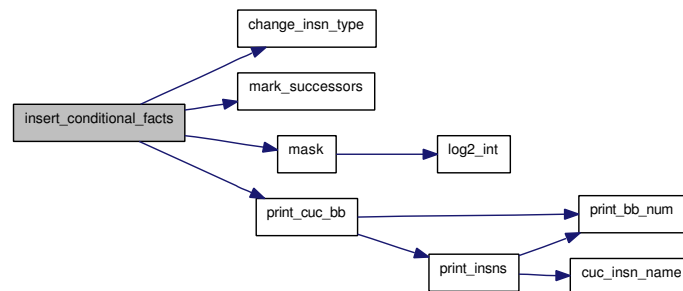
6.66.1.2 `void detect_max_values (cuc_func *f)`

Here is the call graph for this function:



6.66.1.3 `void insert_conditional_facts (cuc_func *f)`

Here is the call graph for this function:



6.66.1.4 `static void mark_successors (cuc_func *f, int b, int m, int stopb)` [static]

6.66.1.5 `static unsigned long mask (unsigned long c)` [static]

Here is the call graph for this function:

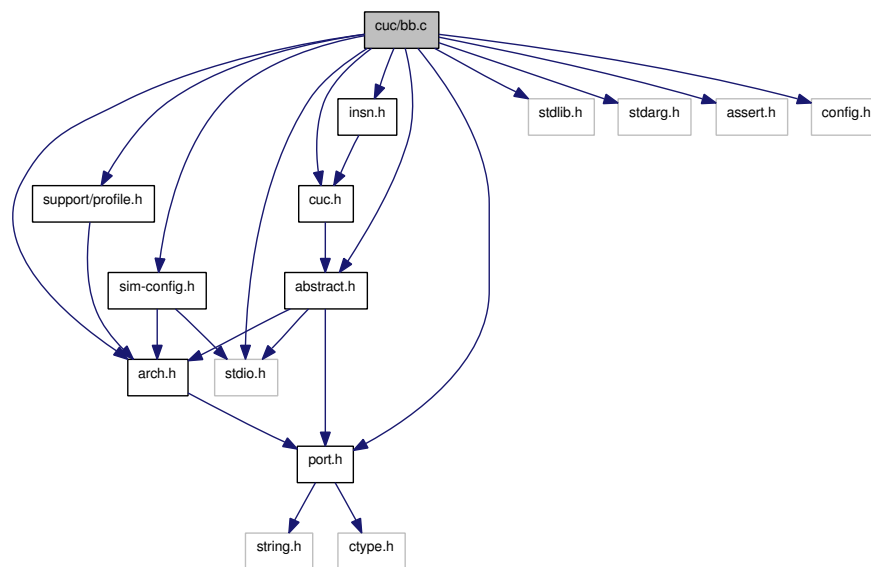


6.66.1.6 `static unsigned long max_op (cuc_func *f, int ref, int o)` [static]

6.67 cuc/bb.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <assert.h>
#include "config.h"
#include "port.h"
#include "arch.h"
#include "sim-config.h"
#include "abstract.h"
#include "cuc.h"
#include "insn.h"
#include "support/profile.h"
```

Include dependency graph for bb.c:



Functions

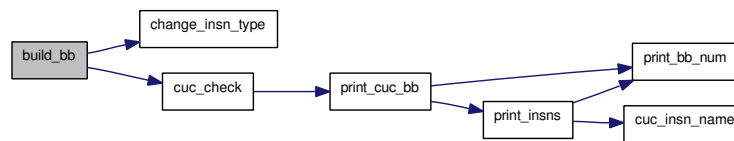
- void [print_bb_num](#) (int num)
- void [print_cuc_bb](#) (cuc_func *f, char *s)
- void [cpy_bb](#) (cuc_bb *dest, cuc_bb *src)
- cuc_func * [dup_func](#) (cuc_func *f)
- void [free_func](#) (cuc_func *f)
- void [recalc_last_used_reg](#) (cuc_func *f, int b)
- void [detect_bb](#) (cuc_func *f)
- void [cuc_check](#) (cuc_func *f)

- void `build_bb` (`cuc_func` *f)
- static void `simplify_bb` (`cuc_func` *f, int pred, int s1, int s2, int neg)
- static void `join_bb` (`cuc_func` *f, int pred, int succ, int type)
- int `optimize_bb` (`cuc_func` *f)
- int `remove_dead_bb` (`cuc_func` *f)
- static void `reg_dep_rec` (`cuc_func` *f, int cur)
- void `reg_dep` (`cuc_func` *f)
- void `expand_bb` (`cuc_func` *f, int b)
- void `generate_bb_seq` (`cuc_func` *f, char *mp_filename, char *bb_filename)
- void `count_bb_seq` (`cuc_func` *f, int b, char *bb_filename, int *counts, int preroll, int unroll)
- static void `relocate_bb` (`cuc_bb` *bb, int b, int back, int fwd)
- static `cuc_func` * `roll_loop` (`cuc_func` *f, int b, int ntimes, int type)
- `cuc_func` * `preunroll_loop` (`cuc_func` *f, int b, int preroll, int unroll, char *bb_filename)

6.67.1 Function Documentation

6.67.1.1 void build_bb (cuc_func * f)

Here is the call graph for this function:



6.67.1.2 void count_bb_seq (cuc_func * f, int b, char * bb_filename, int * counts, int preroll, int unroll)

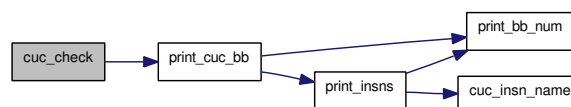
6.67.1.3 void cpy_bb (cuc_bb * dest, cuc_bb * src)

Here is the call graph for this function:



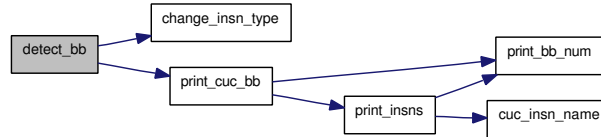
6.67.1.4 void cuc_check (cuc_func * f)

Here is the call graph for this function:

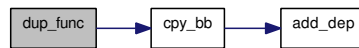


6.67.1.5 void detect_bb (cuc_func *f)

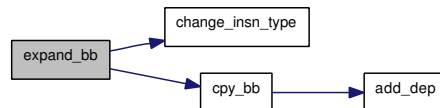
Here is the call graph for this function:

**6.67.1.6 cuc_func* dup_func (cuc_func *f)**

Here is the call graph for this function:

**6.67.1.7 void expand_bb (cuc_func *f, int b)**

Here is the call graph for this function:

**6.67.1.8 void free_func (cuc_func *f)**

Here is the call graph for this function:

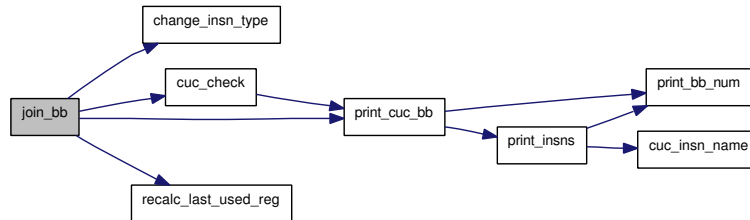
**6.67.1.9 void generate_bb_seq (cuc_func *f, char *mp_filename, char *bb_filename)**

Here is the call graph for this function:



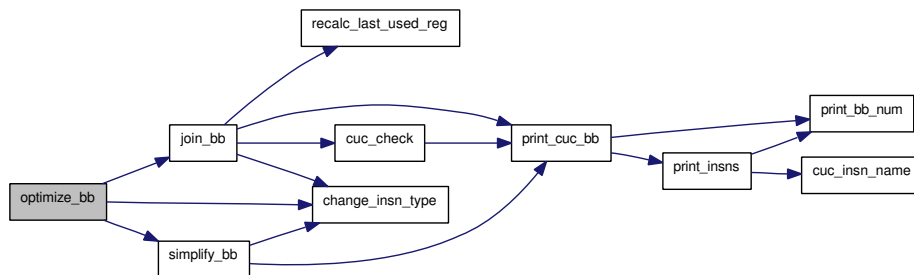
6.67.1.10 static void join_bb (cuc_func *f, int pred, int succ, int type) [static]

Here is the call graph for this function:



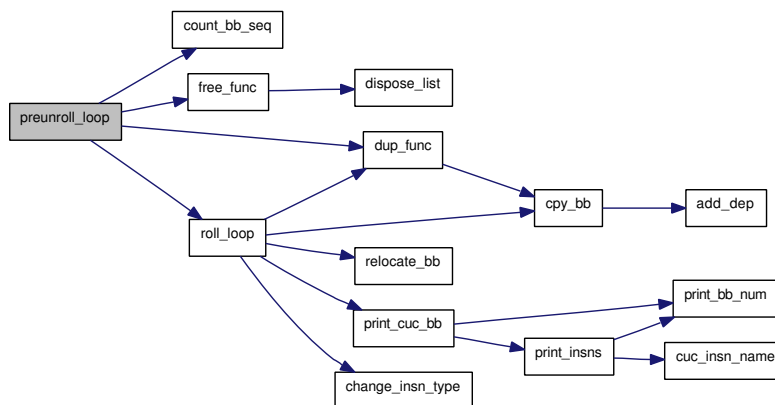
6.67.1.11 int optimize_bb (cuc_func *f)

Here is the call graph for this function:



6.67.1.12 cuc_func* preunroll_loop (cuc_func *f, int b, int preroll, int unroll, char *bb_filename)

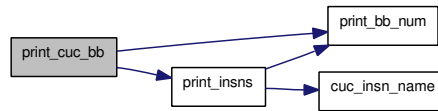
Here is the call graph for this function:



6.67.1.13 void print_bb_num (int num)

6.67.1.14 void print_cuc_bb (cuc_func *f, char *s)

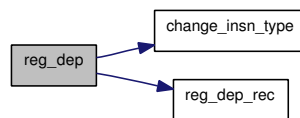
Here is the call graph for this function:



6.67.1.15 void recalc_last_used_reg (cuc_func *f, int b)

6.67.1.16 void reg_dep (cuc_func *f)

Here is the call graph for this function:



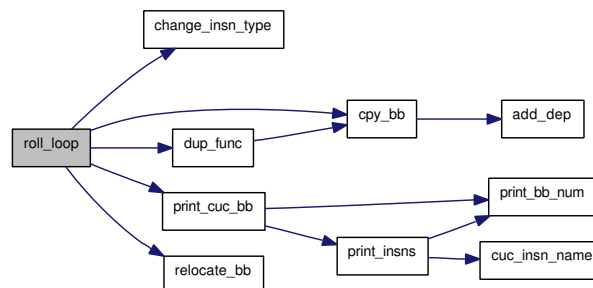
6.67.1.17 static void reg_dep_rec (cuc_func *f, int cur) [static]

6.67.1.18 static void relocate_bb (cuc_bb *bb, int b, int back, int fwd) [static]

6.67.1.19 int remove_dead_bb (cuc_func *f)

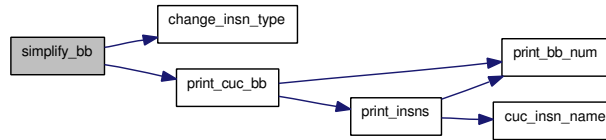
6.67.1.20 static cuc_func* roll_loop (cuc_func *f, int b, int ntimes, int type) [static]

Here is the call graph for this function:



6.67.1.21 static void `simplify_bb` (`cuc_func *f`, `int pred`, `int s1`, `int s2`, `int neg`) [static]

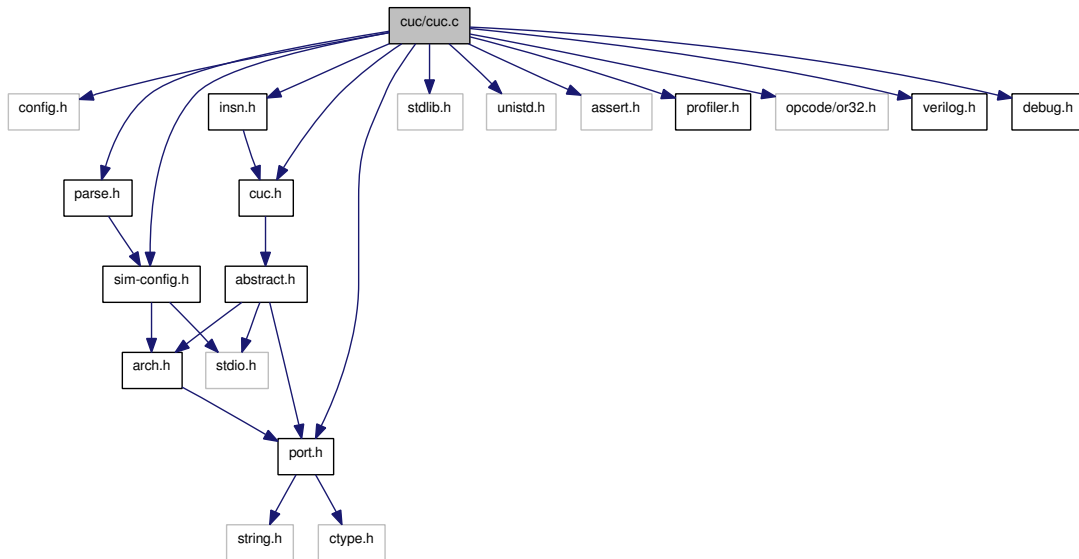
Here is the call graph for this function:



6.68 cuc/cuc.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <unistd.h>
#include <assert.h>
#include "cuc.h"
#include "sim-config.h"
#include "profiler.h"
#include "insn.h"
#include "opcode/or32.h"
#include "parse.h"
#include "verilog.h"
#include "debug.h"
```

Include dependency graph for cuc.c:



Functions

- void [cuc_optimize](#) ([cuc_func](#) *func)
- [cuc_timings](#) *preunroll_bb (char *bb_filename, [cuc_func](#) *f, [cuc_timings](#) *timings, int b, int i, int j)
- int [tim_comp](#) ([cuc_timings](#) *a, [cuc_timings](#) *b)
- [cuc_func](#) * [analyse_function](#) (char *module_name, long orig_time, unsigned long [start_addr](#), unsigned long [end_addr](#), int memory_order, int num_runs)
- char * [gen_option](#) (char *s, int bb_no, int f_opt)
- void [print_option](#) (int bb_no, int f_opt)

- static char * [format_func_options](#) (char *s, [cuc_func](#) *f)
- static void [options_cmd](#) (int func_no, [cuc_func](#) *f)
- [cuc_func](#) * [generate_function](#) ([cuc_func](#) *rf, char *name, char *cut_filename)
- int [calc_cycles](#) ([cuc_func](#) *f)
- double [calc_size](#) ([cuc_func](#) *f)
- unsigned long [extract_function](#) (char *out_fn, unsigned long start_addr)
- static void [set_func_deps](#) ()
- void [main_cuc](#) (char *filename)
- static void [cuc_memory_order](#) (union [param_val](#) val, void *dat)
- static void [cuc_calling_conv](#) (union [param_val](#) val, void *dat)
- static void [cuc_enable_bursts](#) (union [param_val](#) val, void *dat)
- static void [cuc_no_multicycle](#) (union [param_val](#) val, void *dat)
- static void [cuc_timings_fn](#) (union [param_val](#) val, void *dat)
- void [reg_cuc_sec](#) ()

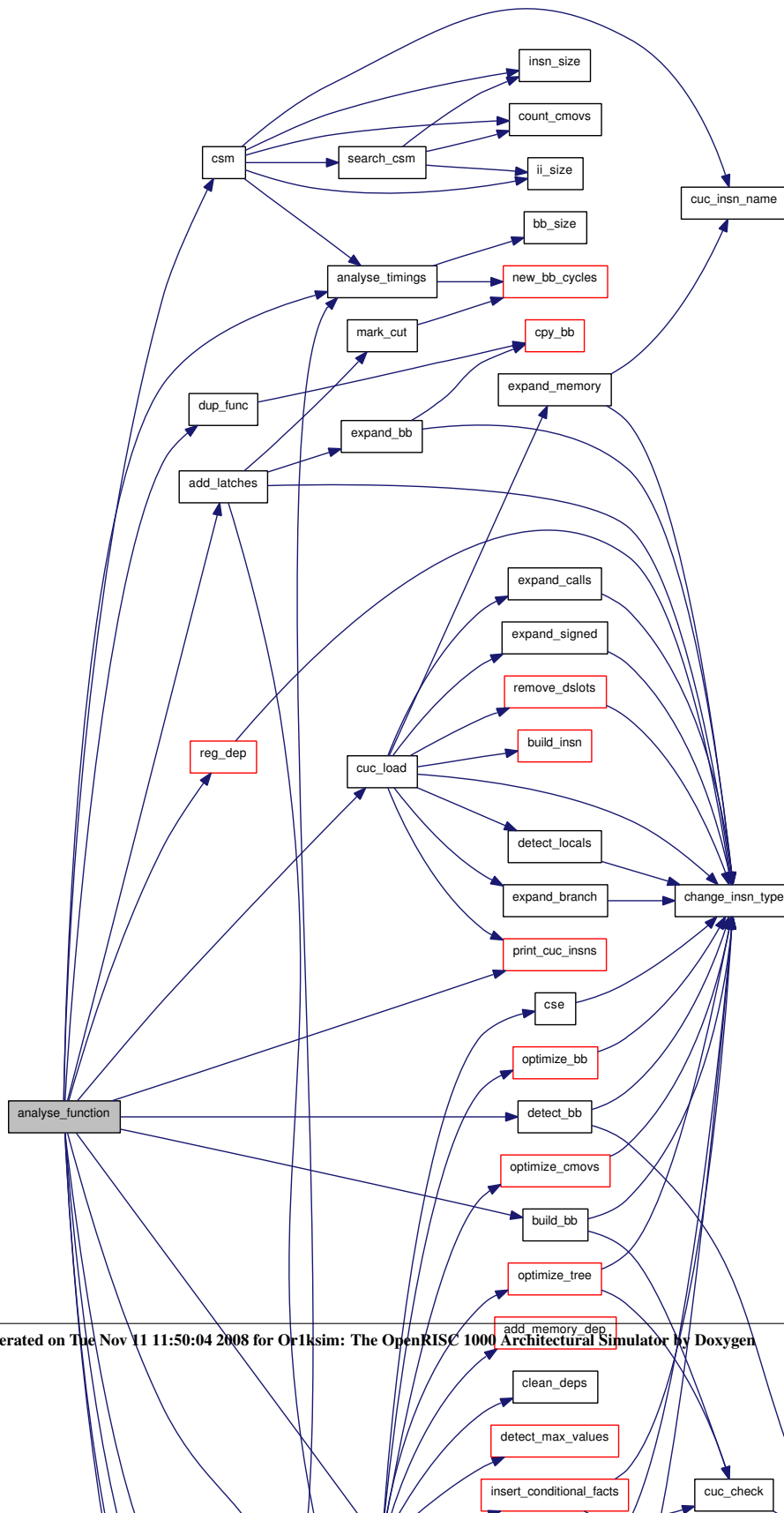
Variables

- FILE * [flog](#)
- int [cuc_debug](#) = 0
- const int [caller_saved](#) [MAX_REGS]
- static const char * [option_char](#)
- static [cuc_func](#) * [func](#) [MAX_FUNCS]
- static int [func_v](#) [MAX_FUNCS]

6.68.1 Function Documentation

6.68.1.1 cuc_func* analyse_function (char * module_name, long orig_time, unsigned long start_addr, unsigned long end_addr, int memory_order, int num_runs)

Here is the call graph for this function:



6.68.1.2 `int calc_cycles (cuc_func *f)`

6.68.1.3 `double calc_size (cuc_func *f)`

6.68.1.4 `static void cuc_calling_conv (union param_val val, void *dat)` [static]

6.68.1.5 `static void cuc_enable_bursts (union param_val val, void *dat)` [static]

6.68.1.6 `static void cuc_memory_order (union param_val val, void *dat)` [static]

Set the memory order

Value must be one of none, weak, strong or exact. Invalid values are ignored with a warning.

Parameters:

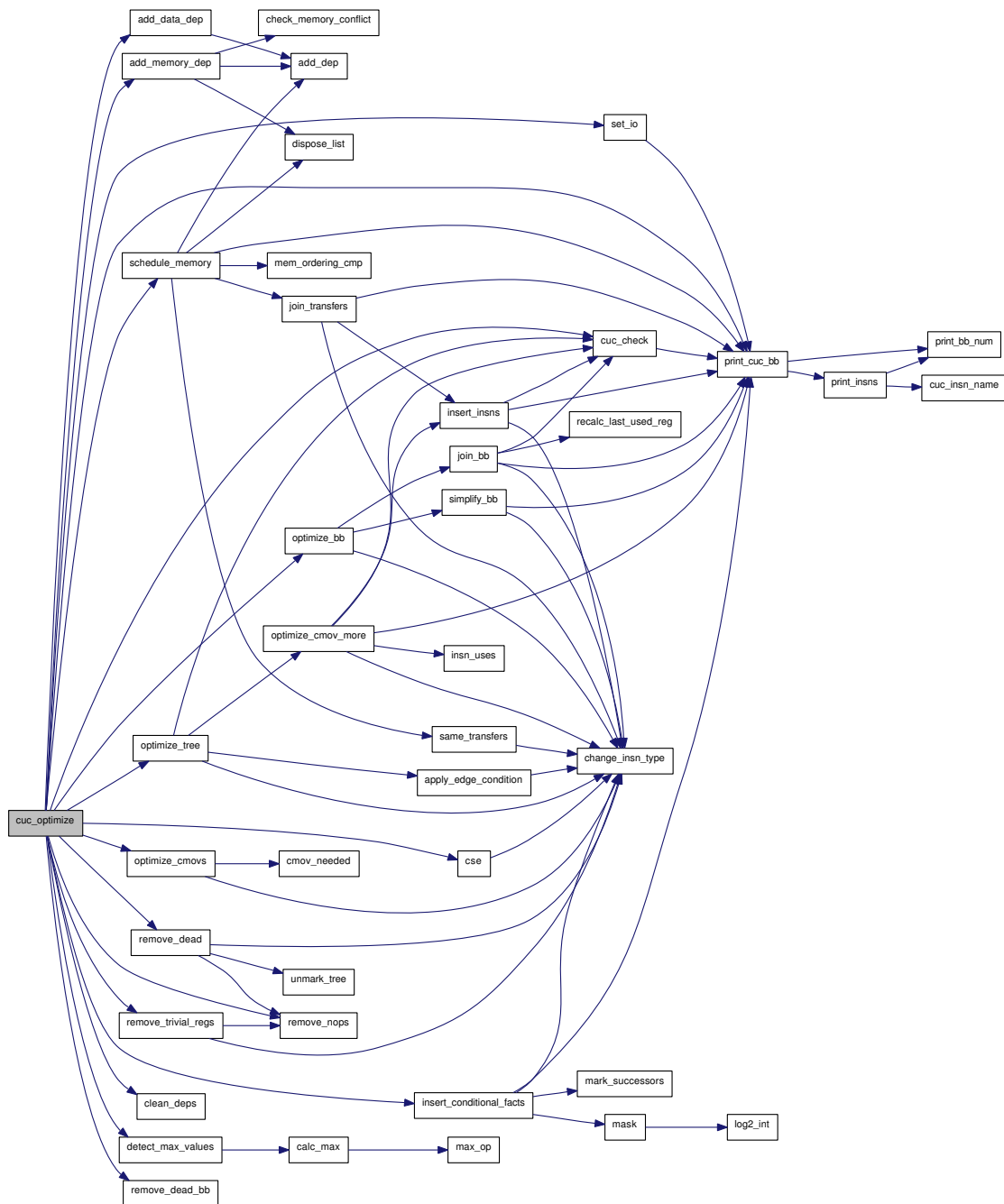
← *val* The value to use

← *dat* The [config](#) data structure (not used here)

6.68.1.7 `static void cuc_no_multicycle (union param_val val, void * dat)` [static]

6.68.1.8 `void cuc_optimize (cuc_func * func)`

Here is the call graph for this function:



6.68.1.9 static void cuc_timings_fn (union param_val val, void * dat) [static]

Set the timings file

Free any existing string.

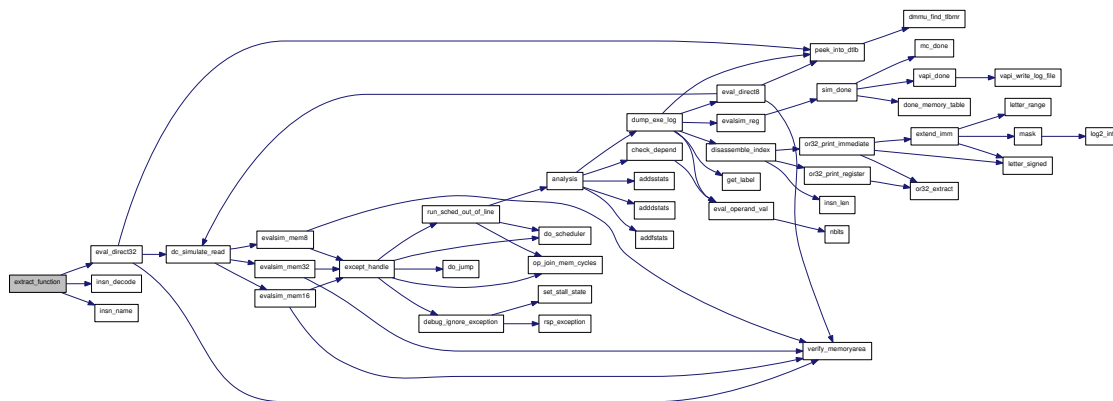
Parameters:

← *val* The value to use

← *dat* The [config](#) data structure (not used here)

6.68.1.10 unsigned long extract_function (char * out_fn, unsigned long start_addr)

Here is the call graph for this function:



6.68.1.11 static char* format_func_options (char * s, cuc_func * f) [static]

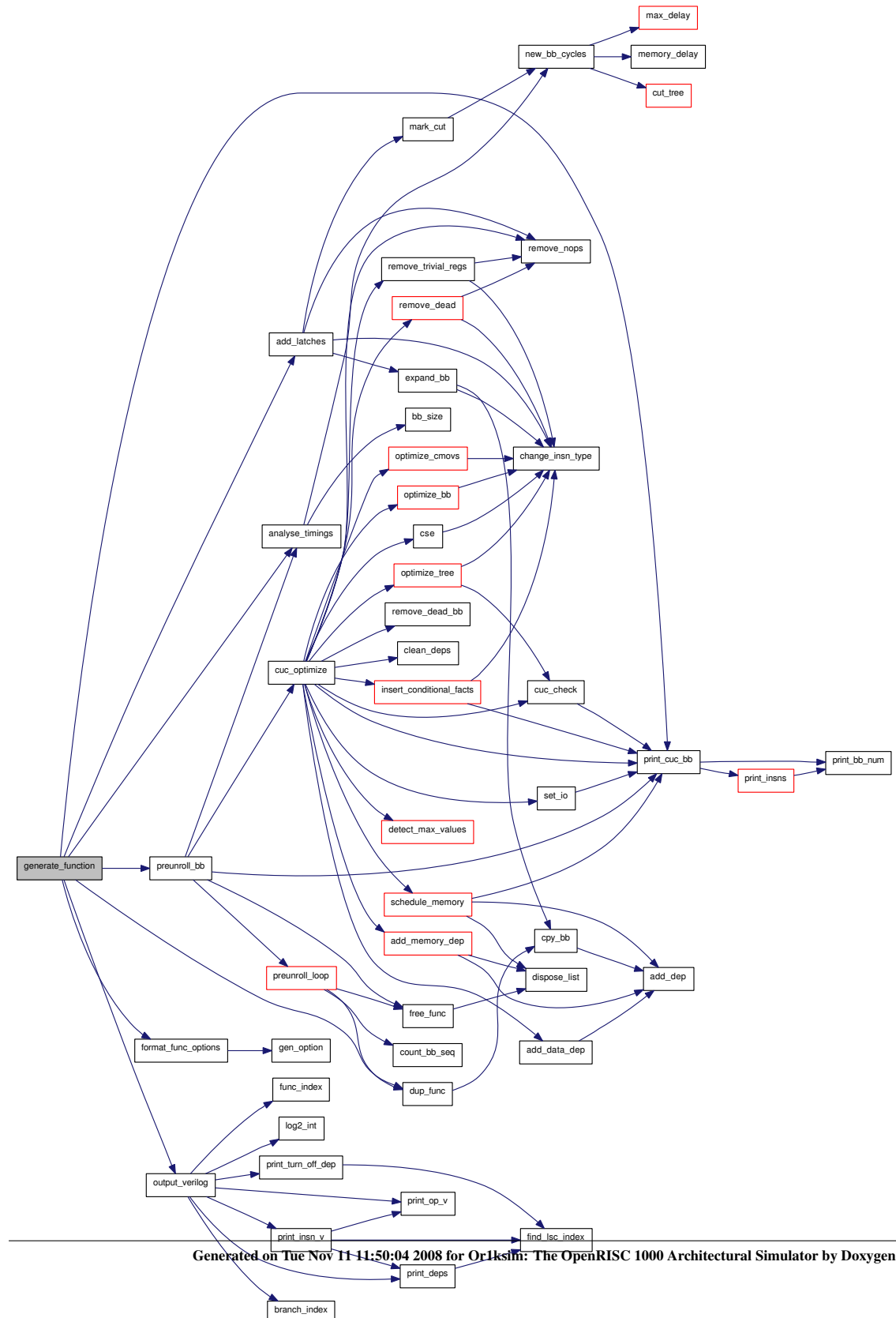
Here is the call graph for this function:



6.68.1.12 `char* gen_option (char * s, int bb_no, int f_opt)`

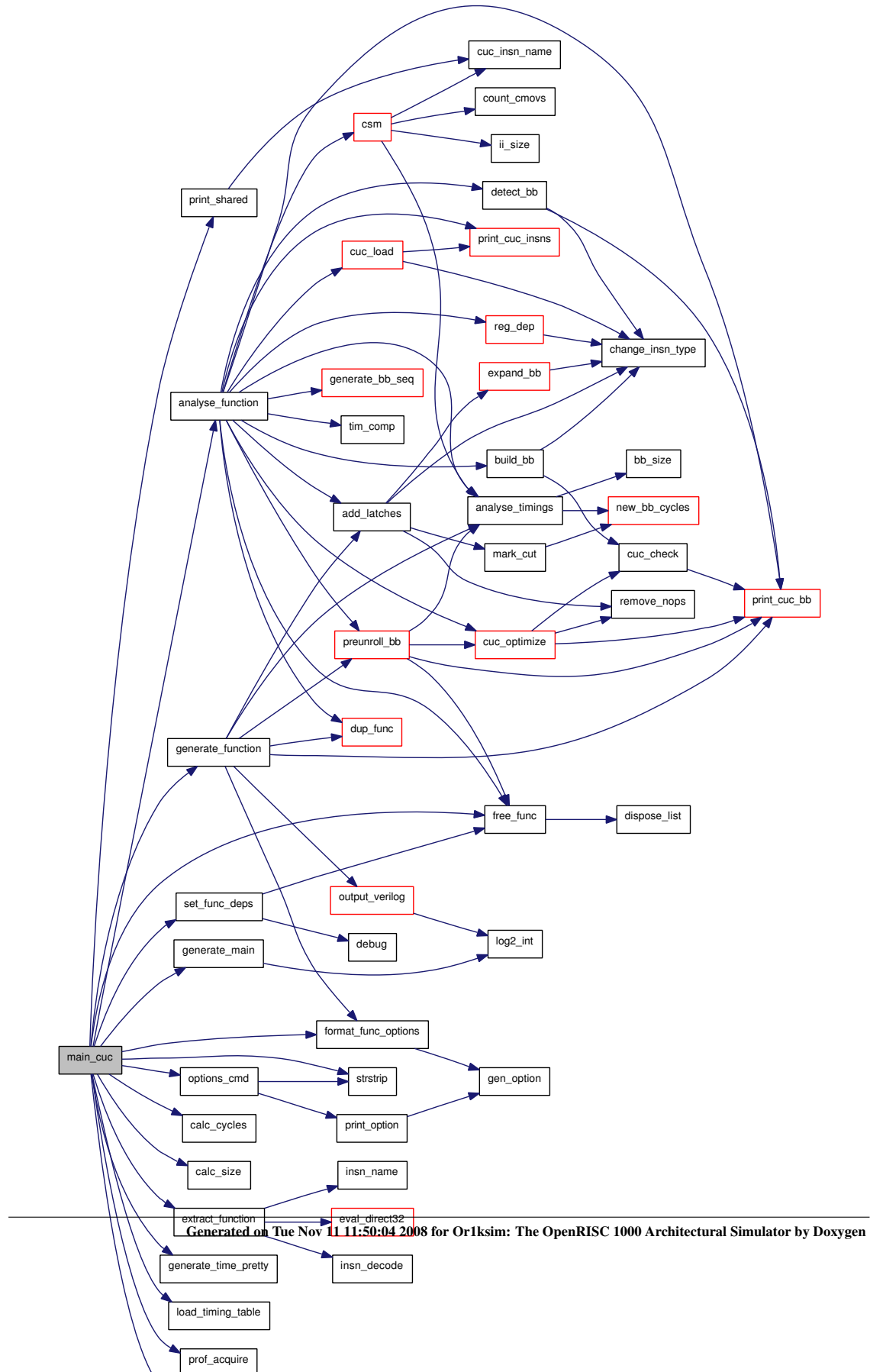
6.68.1.13 `cuc_func* generate_function (cuc_func * rf, char * name, char * cut_filename)`

Here is the call graph for this function:



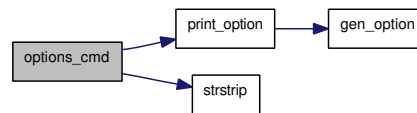
6.68.1.14 void main_cuc (char * filename)

Here is the call graph for this function:



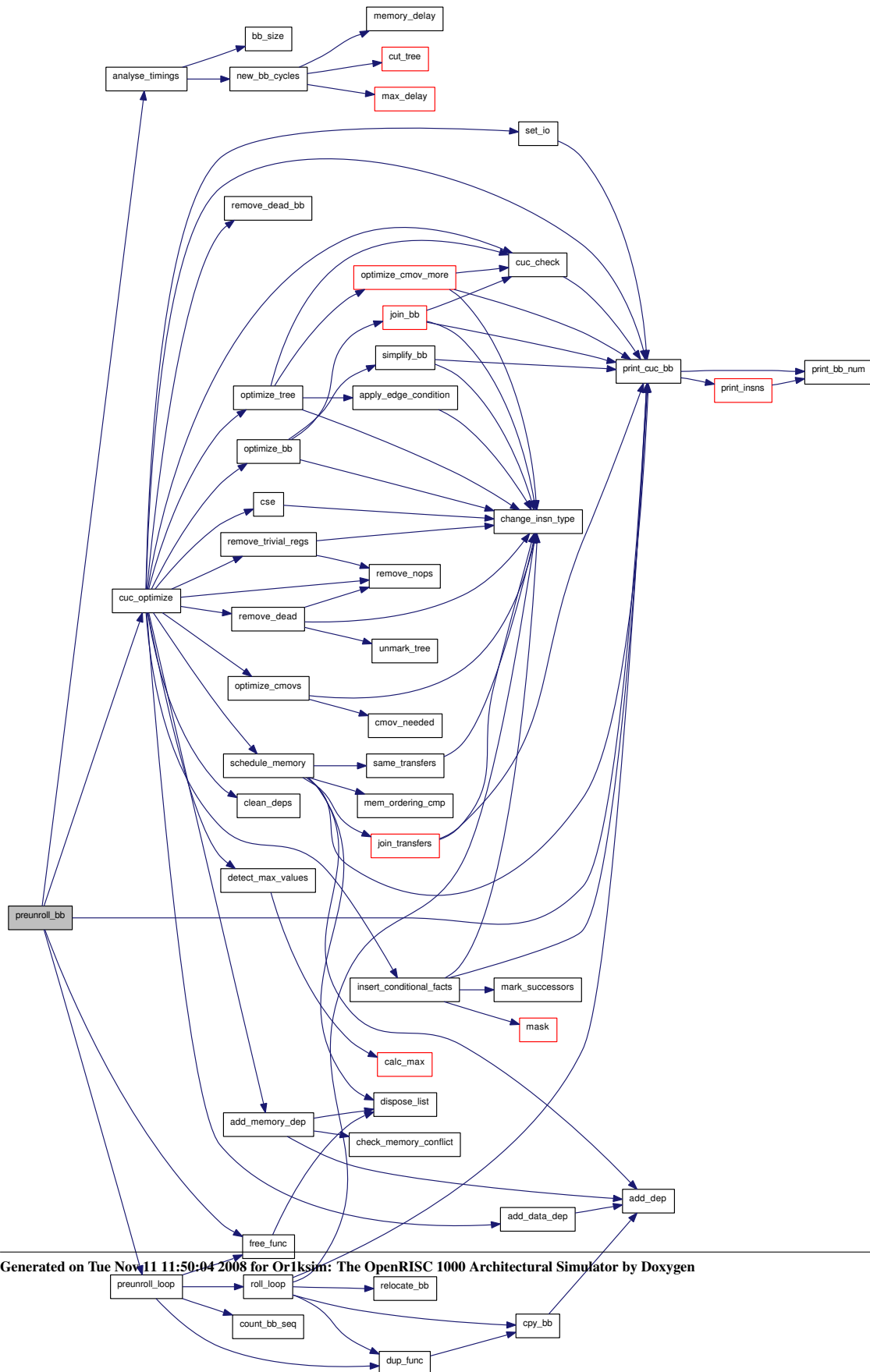
6.68.1.15 static void options_cmd (int *func_no*, cuc_func **f*) [static]

Here is the call graph for this function:



6.68.1.16 cuc_timings* preunroll_bb (char * bb_filename, cuc_func * f, cuc_timings * timings, int b, int i, int j)

Here is the call graph for this function:



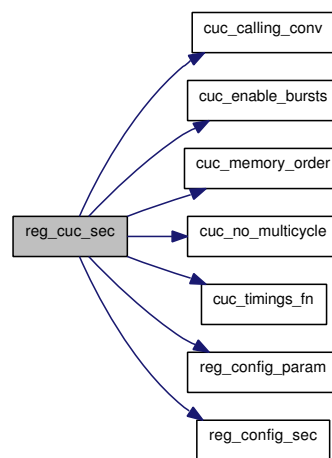
6.68.1.17 void print_option (int *bb_no*, int *f_opt*)

Here is the call graph for this function:



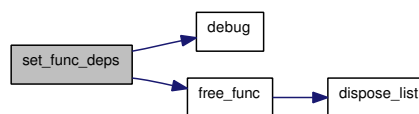
6.68.1.18 void reg_cuc_sec ()

Here is the call graph for this function:



6.68.1.19 static void set_func_deps () [static]

Here is the call graph for this function:



6.68.1.20 int tim_comp (cuc_timings * *a*, cuc_timings * *b*)

6.68.2 Variable Documentation

6.68.2.1 const int caller_saved[MAX_REGS]

Initial value:

```

{
  0, 0, 0, 1, 1, 1, 1, 1,
  1, 1, 0, 0, 0, 1, 0, 1,

```

```
    0, 1, 0, 1, 0, 1, 0, 1,  
    0, 1, 0, 1, 0, 1, 0, 1,  
    1, 1  
}
```

6.68.2.2 `int cuc_debug = 0`

6.68.2.3 `FILE* flog`

6.68.2.4 `cuc_func* func[MAX_FUNCS]` [static]

6.68.2.5 `int func_v[MAX_FUNCS]` [static]

6.68.2.6 `const char* option_char` [static]

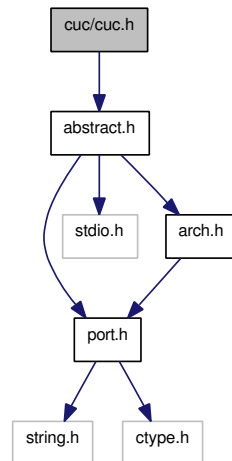
Initial value:

```
"?abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

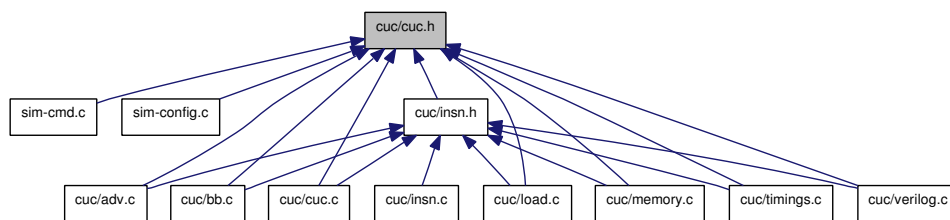
6.69 cuc/cuc.h File Reference

```
#include "abstract.h"
```

Include dependency graph for cuc.h:



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



Data Structures

- struct [_dep_list_t](#)
- struct [_csm_list](#)
- struct [cuc_shared_item](#)
- struct [cuc_timings](#)
- struct [cuc_insn](#)
- struct [cuc_bb](#)
- struct [_cuc_func](#)

Defines

- #define [MAX_INSNS](#) 0x10000
- #define [MAX_BB](#) 0x1000
- #define [MAX_REGS](#) 34
- #define [FLAG_REG](#) (MAX_REGS - 2)
- #define [LRBB_REG](#) (MAX_REGS - 1)

- #define CUC_MAX_STACK 0x1000
- #define MAX_PREROLL 32
- #define MAX_UNROLL 32
- #define IT_BRANCH 0x0001
- #define IT_INDELAY 0x0002
- #define IT_BBSTART 0x0004
- #define IT_BBEND 0x0008
- #define IT_OUTPUT 0x0010
- #define IT_SIGNED 0x0020
- #define IT_MEMORY 0x0040
- #define IT_UNUSED 0x0080
- #define IT_FLAG1 0x0100
- #define IT_FLAG2 0x0200
- #define IT_VOLATILE 0x0400
- #define IT_MEMADD 0x0800
- #define IT_COND 0x1000
- #define IT_LATCHED 0x2000
- #define IT_CUT 0x4000
- #define OPT_NONE 0x00
- #define OPT_CONST 0x01
- #define OPT_REGISTER 0x02
- #define OPT_REF 0x04
- #define OPT_JUMP 0x08
- #define OPT_DEST 0x10
- #define OPT_BB 0x20
- #define OPT_LRBB 0x40
- #define MT_WIDTH 0x007
- #define MT_BURST 0x008
- #define MT_BURSTE 0x010
- #define MT_CALL 0x020
- #define MT_LOAD 0x040
- #define MT_STORE 0x080
- #define MT_SIGNED 0x100
- #define MO_NONE 0
- #define MO_WEAK 1
- #define MO_STRONG 2
- #define MO_EXACT 3
- #define BB_INLOOP 0x01
- #define BB_OPTIONAL 0x02
- #define BB_DEAD 0x08
- #define BBID_START MAX_BB
- #define BBID_END (MAX_BB + 1)
- #define REF(bb, i) (((bb) * MAX_INSNS) + (i))
- #define REF_BB(r) ((r) / MAX_INSNS)
- #define REF_I(r) ((r) % MAX_INSNS)
- #define INSN(ref) bb[REF_BB(ref)].insn[REF_I(ref)]
- #define MIN(x, y) ((x) < (y) ? (x) : (y))
- #define MAX(x, y) ((x) > (y) ? (x) : (y))
- #define log(x...) { fprintf (flog, x); fflush (flog); }
- #define cucdebug(x, s...) { if ((x) <= cuc_debug) PRINTF (s); }
- #define CUC_WIDTH_ITERATIONS 256

Typedefs

- typedef struct `_dep_list_t` `dep_list`
- typedef struct `_csm_list` `cuc_shared_list`
- typedef struct `_cuc_func` `cuc_func`

Functions

- int `cuc_load` (char *in_fn)
- void `negate_conditional` (cuc_insn *ii)
- void `generate_bb_seq` (cuc_func *f, char *mp_filename, char *bb_filename)
- void `print_insns` (int bb, cuc_insn *insn, int size, int verbose)
- void `print_bb_num` (int num)
- void `print_cuc_bb` (cuc_func *func, char *s)
- cuc_func * `dup_func` (cuc_func *f)
- void `free_func` (cuc_func *f)
- void `csm` (cuc_func *f)
- void `csm_gen` (cuc_func *f, cuc_func *rf, cuc_shared_item *shared, int nshared)
- void `detect_bb` (cuc_func *func)
- int `optimize_bb` (cuc_func *func)
- int `optimize_cmovs` (cuc_func *func)
- int `optimize_tree` (cuc_func *func)
- int `remove_nops` (cuc_func *func)
- int `remove_dead` (cuc_func *func)
- int `remove_trivial_regs` (cuc_func *f)
- void `set_io` (cuc_func *func)
- int `remove_dead_bb` (cuc_func *func)
- int `cse` (cuc_func *f)
- void `reg_dep` (cuc_func *func)
- void `mark_cut` (cuc_func *f)
- cuc_func * `preunroll_loop` (cuc_func *func, int b, int preroll, int unroll, char *bb_filename)
- void `clean_deps` (cuc_func *func)
- int `schedule_memory` (cuc_func *func, int otype)
- void `recalc_cnts` (cuc_func *f, char *bb_filename)
- void `analyse_timings` (cuc_func *func, cuc_timings *timings)
- void `insert_conditional_facts` (cuc_func *func)
- void `detect_max_values` (cuc_func *f)
- void `insert_insns` (cuc_func *f, int ref, int n)
- void `cuc_check` (cuc_func *f)
- void `add_memory_dep` (cuc_func *f, int otype)
- void `print_cuc_insns` (char *s, int verbose)
- void `build_bb` (cuc_func *f)
- void `add_latches` (cuc_func *f)
- void `expand_bb` (cuc_func *f, int b)
- void `add_dep` (dep_list **list, int dep)
- void `dispose_list` (dep_list **list)
- void `main_cuc` (char *filename)
- void `add_data_dep` (cuc_func *f)
- void `reg_cuc_sec` ()

Variables

- int [cuc_debug](#)

- const int [caller_saved](#) [MAX_REGS]

- [cuc_insn](#) [insn](#) [MAX_INSNS]

- int [num_insn](#)

- int [reloc](#) [MAX_INSNS]

- FILE * [flog](#)

6.69.1 Define Documentation

6.69.1.1 `#define BB_DEAD 0x08`

6.69.1.2 `#define BB_INLOOP 0x01`

6.69.1.3 `#define BB_OPTIONAL 0x02`

6.69.1.4 `#define BBID_END (MAX_BB + 1)`

6.69.1.5 `#define BBID_START MAX_BB`

6.69.1.6 `#define CUC_MAX_STACK 0x1000`

6.69.1.7 `#define CUC_WIDTH_ITERATIONS 256`

6.69.1.8 `#define cucdebug(x, s...) {if ((x) <= cuc_debug) PRINTF (s);}`

6.69.1.9 `#define FLAG_REG (MAX_REGS - 2)`

6.69.1.10 `#define INSN(ref) bb[REF_BB(ref)].insn[REF_I(ref)]`

6.69.1.11 `#define IT_BBEND 0x0008`

6.69.1.12 `#define IT_BBSTART 0x0004`

6.69.1.13 `#define IT_BRANCH 0x0001`

6.69.1.14 `#define IT_COND 0x1000`

6.69.1.15 `#define IT_CUT 0x4000`

6.69.1.16 `#define IT_FLAG1 0x0100`

6.69.1.17 `#define IT_FLAG2 0x0200`

6.69.1.18 `#define IT_INDELAY 0x0002`

6.69.1.19 `#define IT_LATCHED 0x2000`

6.69.1.20 `#define IT_MEMADD 0x0800`

6.69.1.21 `#define IT_MEMORY 0x0040`

6.69.1.22 `#define IT_OUTPUT 0x0010`

6.69.1.23 `#define IT_SIGNED 0x0020`

6.69.1.24 `#define IT_UNUSED 0x0080`

6.69.1.25 `#define IT_VOLATILE 0x0400`

6.69.1.26 `#define log(x...) {fprintf (flog, x); fflush (flog); }`

6.69.1.27 `#define LRB_REG (MAX_REGS - 1)`
Generated on Tue Nov 11 14:56:34 2008 for OlsSim: The OpenMIPS 1000 Architectural Simulator by Doxygen

6.69.1.28 `#define MAX(x, y) ((x) > (y) ? (x) : (y))`

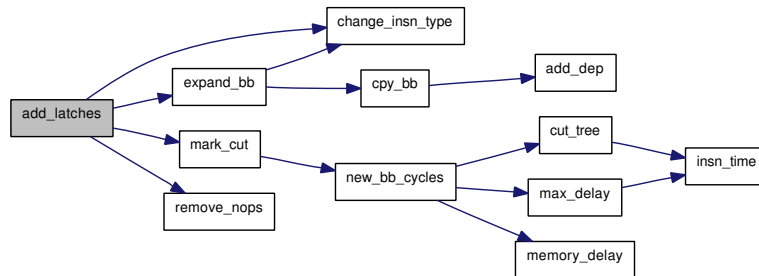
6.69.1.29 `#define MAX_BB 0x1000`

6.69.1.30 `#define MAX_INSNS 0x10000`

6.69.3.2 void add_dep (dep_list ** list, int dep)

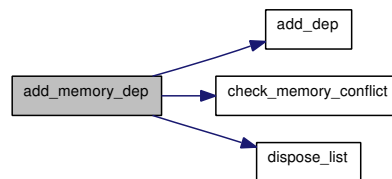
6.69.3.3 void add_latches (cuc_func * f)

Here is the call graph for this function:



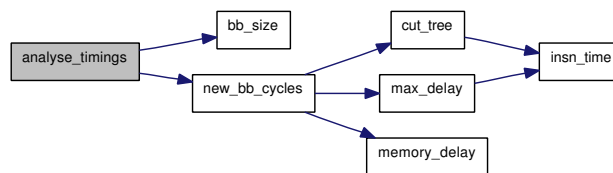
6.69.3.4 void add_memory_dep (cuc_func * f, int otype)

Here is the call graph for this function:



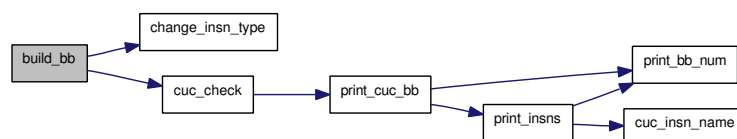
6.69.3.5 void analyse_timings (cuc_func * func, cuc_timings * timings)

Here is the call graph for this function:



6.69.3.6 void build_bb (cuc_func * f)

Here is the call graph for this function:

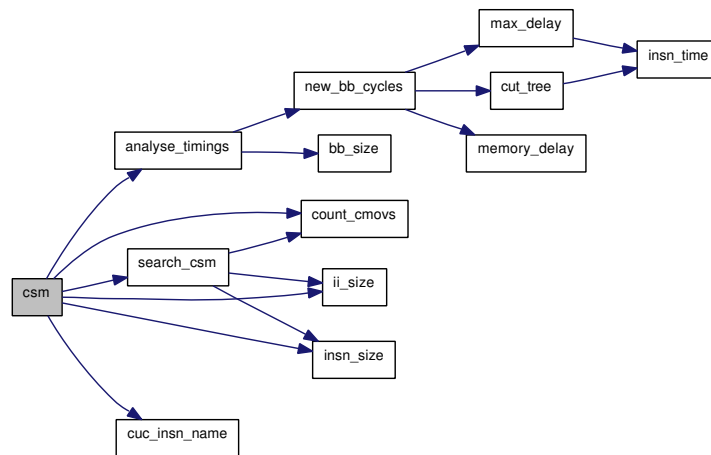


6.69.3.7 void clean_deps (cuc_func *func)**6.69.3.8 int cse (cuc_func *f)**

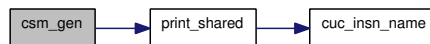
Here is the call graph for this function:

**6.69.3.9 void csm (cuc_func *f)**

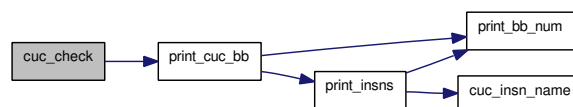
Here is the call graph for this function:

**6.69.3.10 void csm_gen (cuc_func *f, cuc_func *rf, cuc_shared_item *shared, int nshared)**

Here is the call graph for this function:

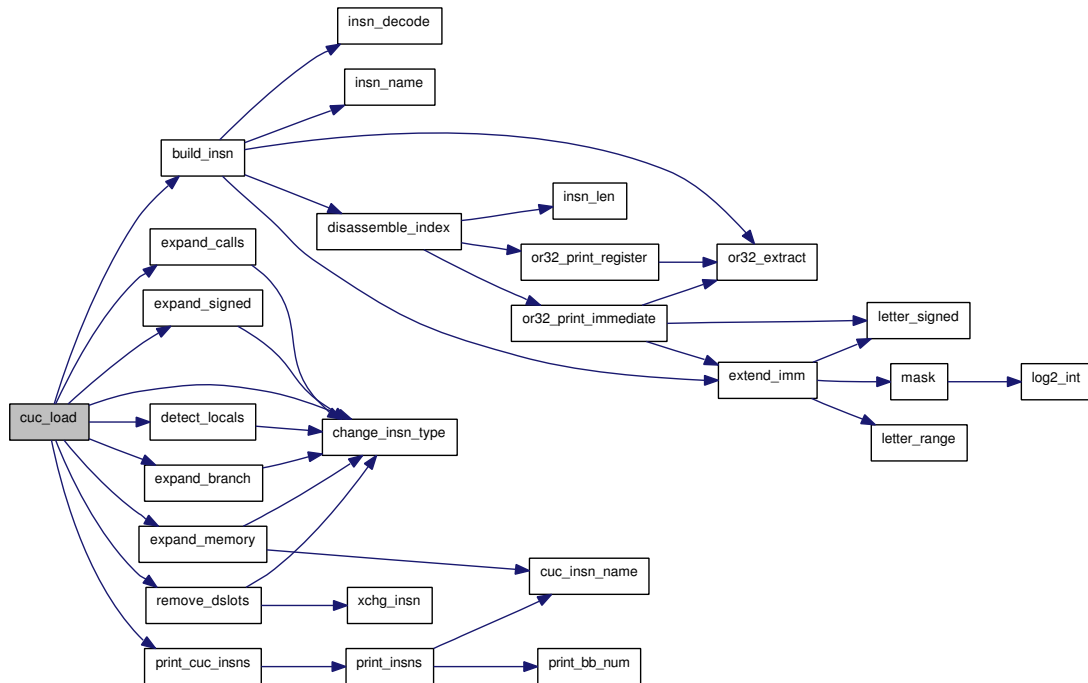
**6.69.3.11 void cuc_check (cuc_func *f)**

Here is the call graph for this function:



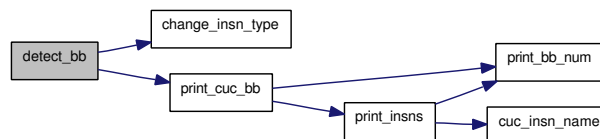
6.69.3.12 int cuc_load (char * in_fn)

Here is the call graph for this function:



6.69.3.13 void detect_bb (cuc_func * func)

Here is the call graph for this function:



6.69.3.14 void detect_max_values (cuc_func * f)

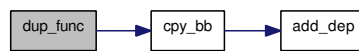
Here is the call graph for this function:



6.69.3.15 void dispose_list (dep_list ** list)

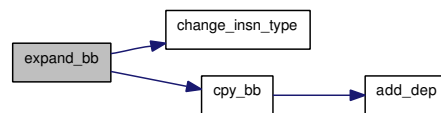
6.69.3.16 cuc_func* dup_func (cuc_func * f)

Here is the call graph for this function:



6.69.3.17 void expand_bb (cuc_func * f, int b)

Here is the call graph for this function:



6.69.3.18 void free_func (cuc_func * f)

Here is the call graph for this function:



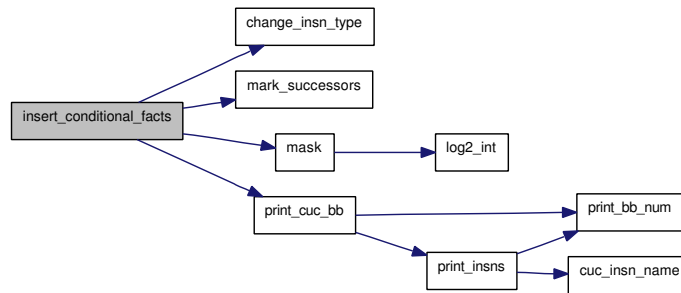
6.69.3.19 void generate_bb_seq (cuc_func * f, char * mp_filename, char * bb_filename)

Here is the call graph for this function:



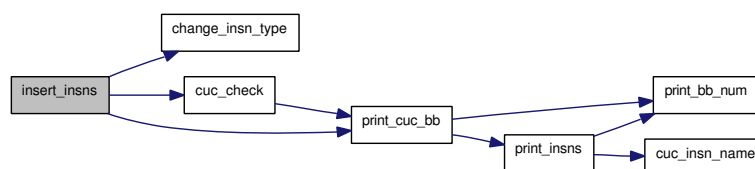
6.69.3.20 void insert_conditional_facts (cuc_func *func)

Here is the call graph for this function:



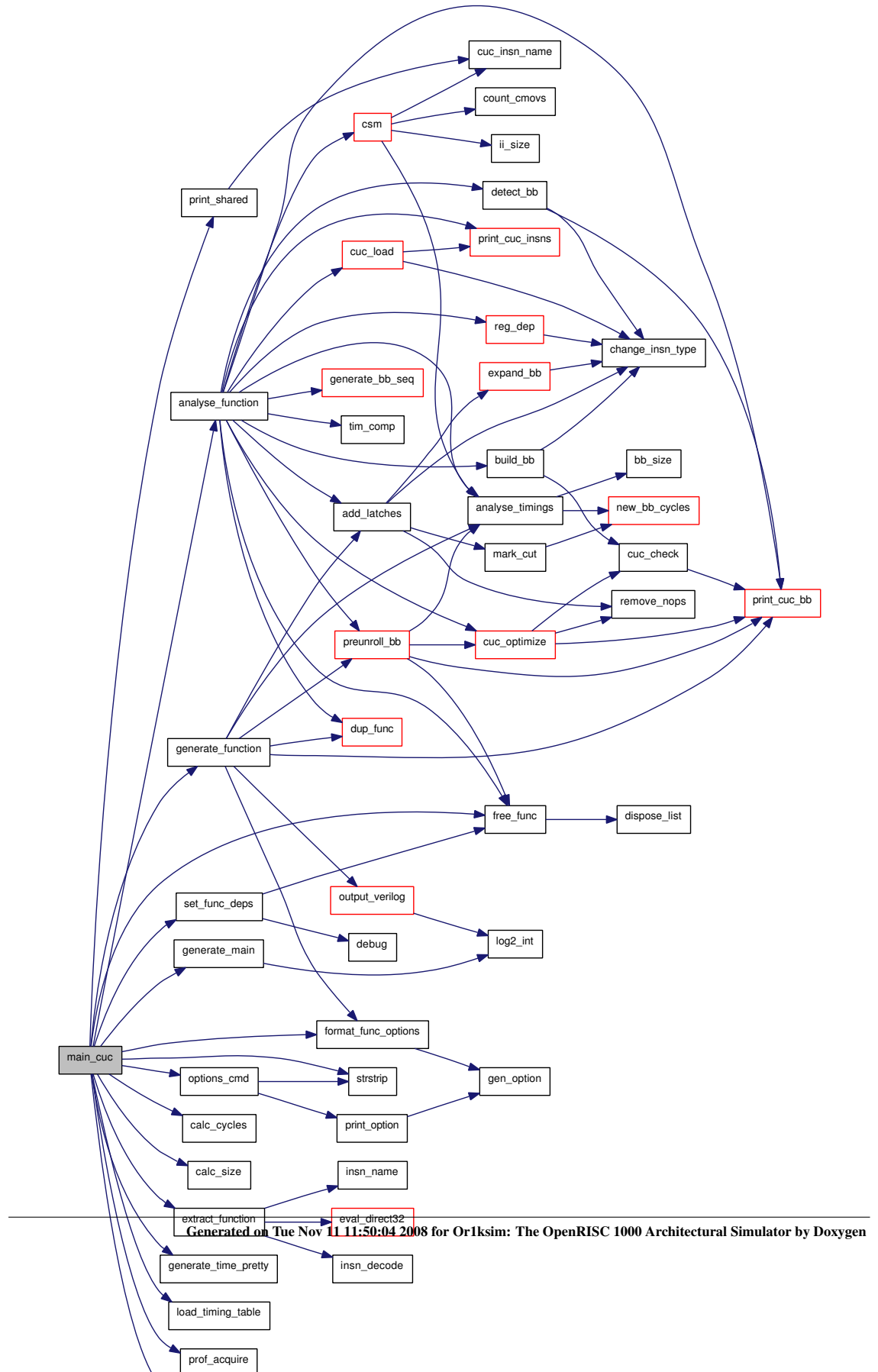
6.69.3.21 void insert_insns (cuc_func *f, int ref, int n)

Here is the call graph for this function:



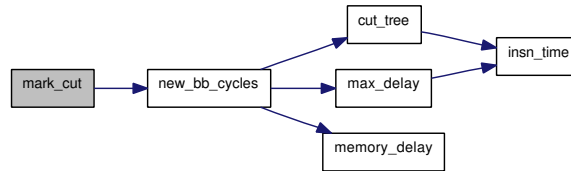
6.69.3.22 void main_cuc (char * filename)

Here is the call graph for this function:



6.69.3.23 void mark_cut (cuc_func * f)

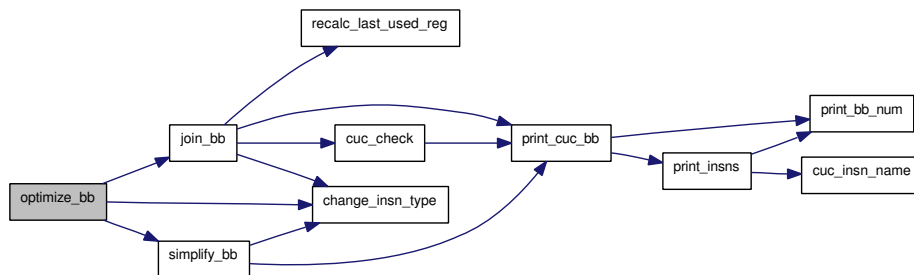
Here is the call graph for this function:

**6.69.3.24 void negate_conditional (cuc_insn * ii)**

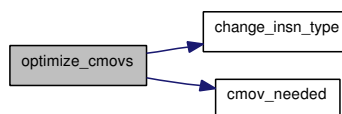
Here is the call graph for this function:

**6.69.3.25 int optimize_bb (cuc_func * func)**

Here is the call graph for this function:

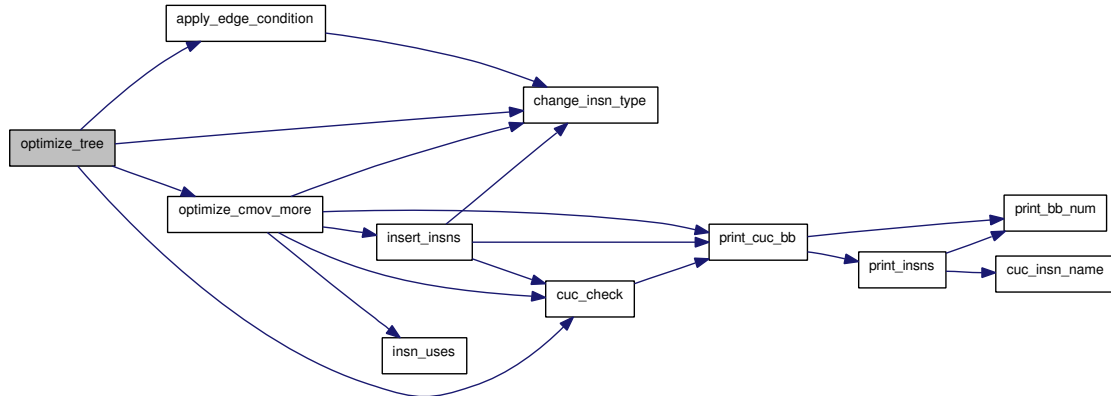
**6.69.3.26 int optimize_cmovs (cuc_func * func)**

Here is the call graph for this function:



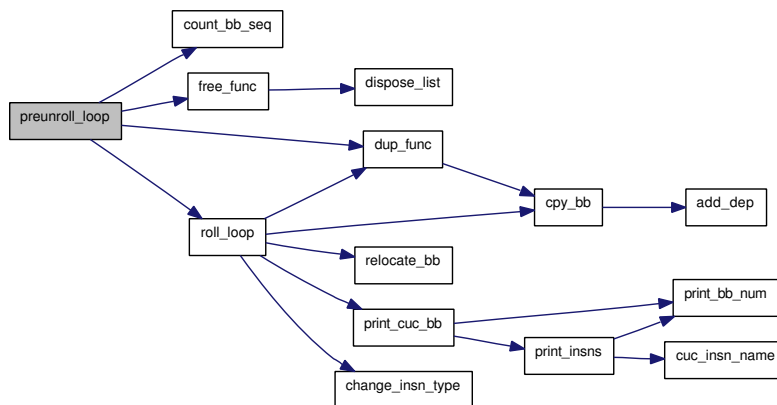
6.69.3.27 int optimize_tree (cuc_func * func)

Here is the call graph for this function:



6.69.3.28 cuc_func* preunroll_loop (cuc_func * func, int b, int preroll, int unroll, char * bb_filename)

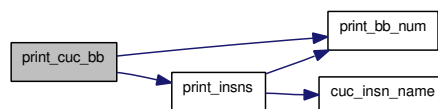
Here is the call graph for this function:



6.69.3.29 void print_bb_num (int num)

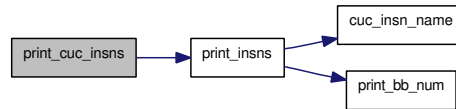
6.69.3.30 void print_cuc_bb (cuc_func * func, char * s)

Here is the call graph for this function:

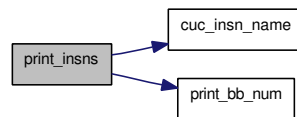


6.69.3.31 void print_cuc_insns (char * s, int verbose)

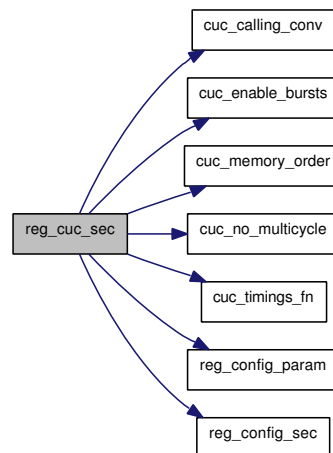
Here is the call graph for this function:

**6.69.3.32 void print_insns (int bb, cuc_insn * insn, int size, int verbose)**

Here is the call graph for this function:

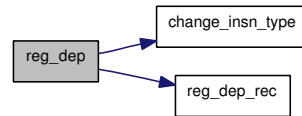
**6.69.3.33 void recalc_cnts (cuc_func * f, char * bb_filename)****6.69.3.34 void reg_cuc_sec ()**

Here is the call graph for this function:

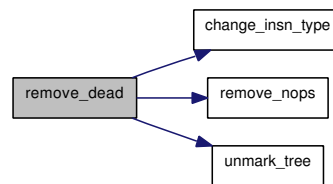


6.69.3.35 void reg_dep (cuc_func * *func*)

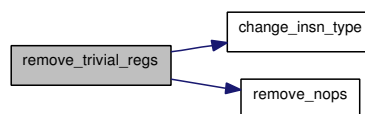
Here is the call graph for this function:

**6.69.3.36 int remove_dead (cuc_func * *func*)**

Here is the call graph for this function:

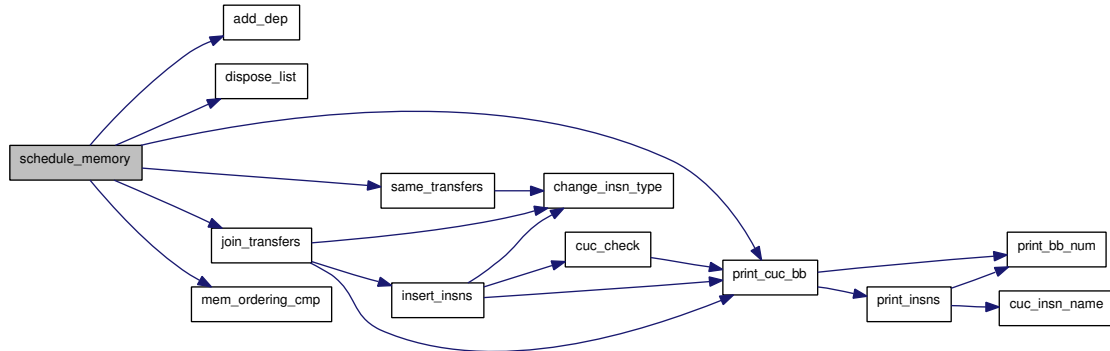
**6.69.3.37 int remove_dead_bb (cuc_func * *func*)****6.69.3.38 int remove_nops (cuc_func * *func*)****6.69.3.39 int remove_trivial_regs (cuc_func * *f*)**

Here is the call graph for this function:



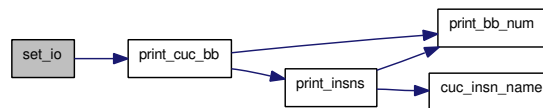
6.69.3.40 int schedule_memory (cuc_func *func, int otype)

Here is the call graph for this function:



6.69.3.41 void set_io (cuc_func *func)

Here is the call graph for this function:



6.69.4 Variable Documentation

6.69.4.1 const int caller_saved[MAX_REGS]

6.69.4.2 int cuc_debug

6.69.4.3 FILE* flog

6.69.4.4 cuc_insn insn[MAX_INSNS]

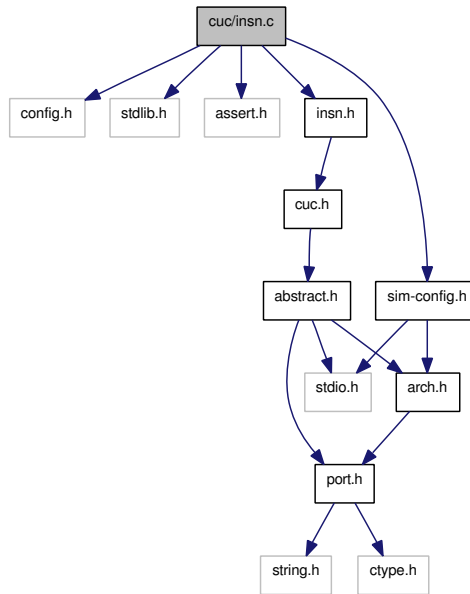
6.69.4.5 int num_insn

6.69.4.6 int reloc[MAX_INSNS]

6.70 cuc/insn.c File Reference

```
#include "config.h"
#include <stdlib.h>
#include <assert.h>
#include "insn.h"
#include "sim-config.h"
```

Include dependency graph for insn.c:



Functions

- void [change_insn_type](#) (`cuc_insn *i`, int index)
- const char * [cuc_insn_name](#) (`cuc_insn *i`)
- void [print_insns](#) (int bb, `cuc_insn *insn`, int ninsn, int verbose)
- void [add_dep](#) (`dep_list **list`, int dep)
- void [dispose_list](#) (`dep_list **list`)
- void [add_data_dep](#) (`cuc_func *f`)
- void [insert_insns](#) (`cuc_func *f`, int ref, int n)
- int [apply_edge_condition](#) (`cuc_insn *i`)
- static int [cmov_needed](#) (`cuc_func *f`, int ref)
- int [optimize_cmovs](#) (`cuc_func *f`)
- static int [insn_uses](#) (`cuc_func *f`, int ref)
- static int [optimize_cmov_more](#) (`cuc_func *f`, int ref)
- int [optimize_tree](#) (`cuc_func *f`)
- int [remove_nops](#) (`cuc_func *f`)
- static void [unmark_tree](#) (`cuc_func *f`, int ref)
- int [remove_dead](#) (`cuc_func *f`)
- int [remove_trivial_regs](#) (`cuc_func *f`)

- void `set_io` (`cuc_func` *f)
- void `add_latches` (`cuc_func` *f)
- int `cse` (`cuc_func` *f)
- static int `count_cmovs` (`cuc_insn` *ii, int match)
- static void `search_csm` (int iter, `cuc_func` *f, `cuc_shared_list` *list)
- void `csm` (`cuc_func` *f)
- void `print_shared` (`cuc_func` *rf, `cuc_shared_item` *shared, int nshared)
- void `csm_gen` (`cuc_func` *f, `cuc_func` *rf, `cuc_shared_item` *shared, int nshared)

Variables

- const `cuc_known_insn` `known` [II_LAST+1]
- static unsigned long `tmp_op`
- static unsigned long `tmp_opt`
- static `cuc_shared_list` * `main_list`
- static int * `iteration`

6.70.1 Function Documentation

6.70.1.1 void add_data_dep (cuc_func *f)

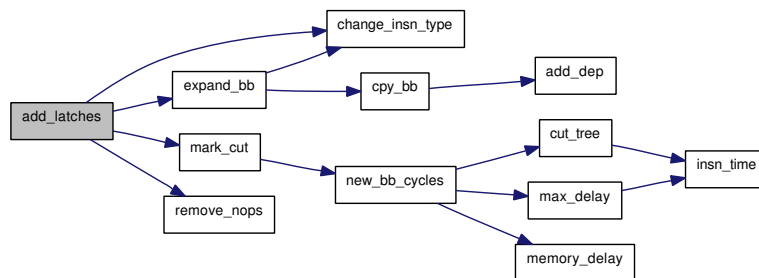
Here is the call graph for this function:



6.70.1.2 void add_dep (dep_list **list, int dep)

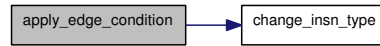
6.70.1.3 void add_latches (cuc_func *f)

Here is the call graph for this function:



6.70.1.4 int apply_edge_condition (cuc_insn * ii)

Here is the call graph for this function:



6.70.1.5 void change_insn_type (cuc_insn * i, int index)

6.70.1.6 static int cmov_needed (cuc_func * f, int ref) [static]

6.70.1.7 static int count_cmovs (cuc_insn * ii, int match) [static]

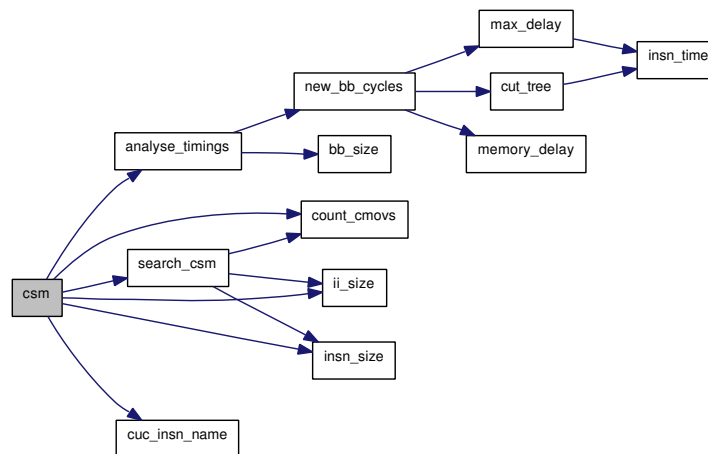
6.70.1.8 int cse (cuc_func * f)

Here is the call graph for this function:



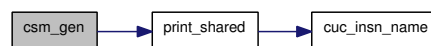
6.70.1.9 void csm (cuc_func * f)

Here is the call graph for this function:



6.70.1.10 void csm_gen (cuc_func * f, cuc_func * rf, cuc_shared_item * shared, int nshared)

Here is the call graph for this function:

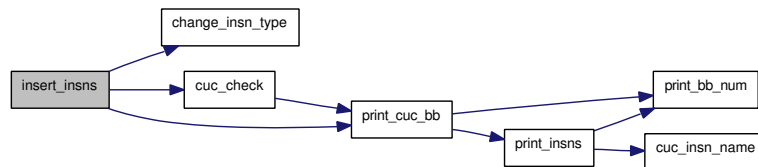


6.70.1.11 `const char* cuc_insn_name (cuc_insn * ii)`

6.70.1.12 `void dispose_list (dep_list ** list)`

6.70.1.13 `void insert_insns (cuc_func * f, int ref, int n)`

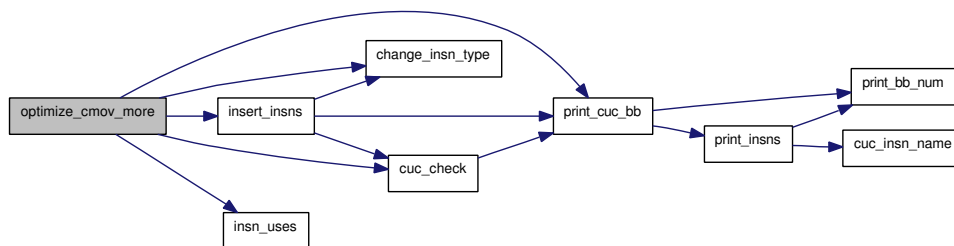
Here is the call graph for this function:



6.70.1.14 `static int insn_uses (cuc_func * f, int ref)` [static]

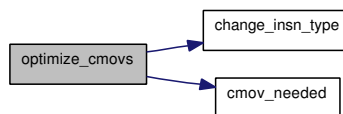
6.70.1.15 `static int optimize_cmov_more (cuc_func * f, int ref)` [static]

Here is the call graph for this function:



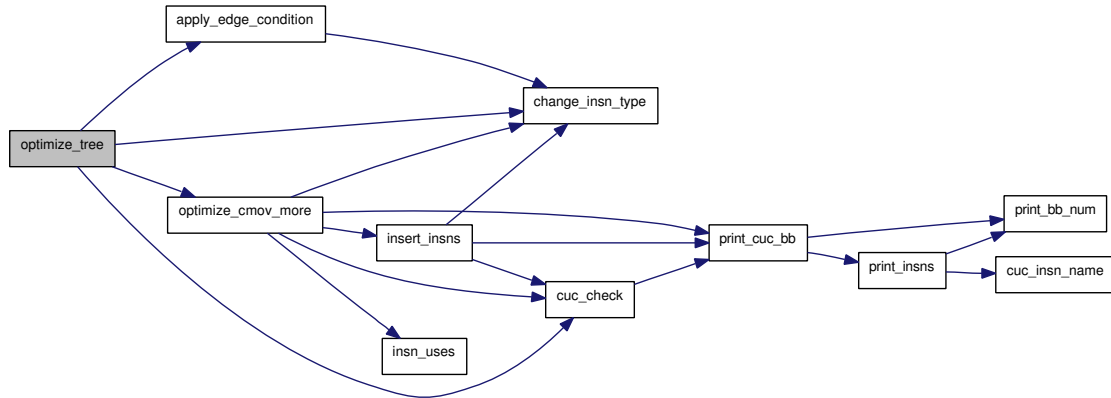
6.70.1.16 `int optimize_cmovs (cuc_func * f)`

Here is the call graph for this function:



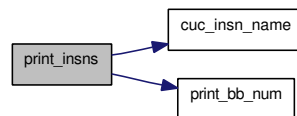
6.70.1.17 int optimize_tree (cuc_func * f)

Here is the call graph for this function:



6.70.1.18 void print_insn (int bb, cuc_insn * insn, int ninsn, int verbose)

Here is the call graph for this function:



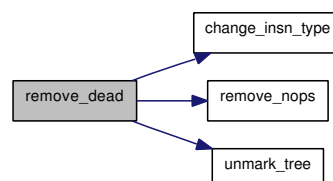
6.70.1.19 void print_shared (cuc_func * rf, cuc_shared_item * shared, int nshared)

Here is the call graph for this function:



6.70.1.20 int remove_dead (cuc_func * f)

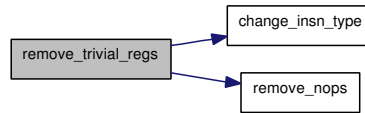
Here is the call graph for this function:



6.70.1.21 `int remove_nops (cuc_func *f)`

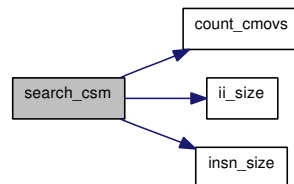
6.70.1.22 `int remove_trivial_regs (cuc_func *f)`

Here is the call graph for this function:



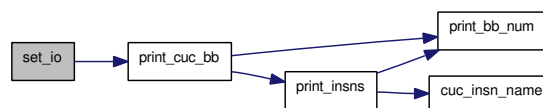
6.70.1.23 `static void search_csm (int iter, cuc_func *f, cuc_shared_list *list)` [static]

Here is the call graph for this function:



6.70.1.24 `void set_io (cuc_func *f)`

Here is the call graph for this function:



6.70.1.25 `static void unmark_tree (cuc_func *f, int ref)` [static]

6.70.2 Variable Documentation

6.70.2.1 `int* iteration` [static]

6.70.2.2 `const cuc_known_insn known[II_LAST+1]`

6.70.2.3 `cuc_shared_list* main_list` [static]

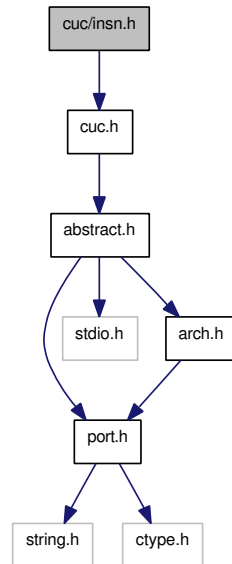
6.70.2.4 `unsigned long tmp_op` [static]

6.70.2.5 `unsigned long tmp_opt` [static]

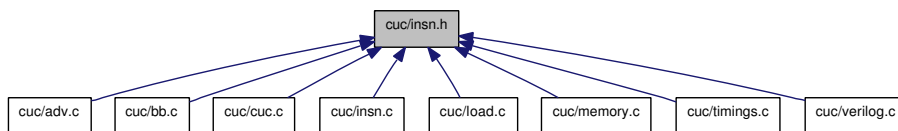
6.71 cuc/insn.h File Reference

```
#include "cuc.h"
```

Include dependency graph for insn.h:



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



Data Structures

- struct [cuc_known_insn](#)
- struct [cuc_timing_table](#)
- struct [cuc_conv](#)

Defines

- #define [II_ADD](#) 0
- #define [II_SUB](#) 1
- #define [II_AND](#) 2
- #define [II_OR](#) 3
- #define [II_XOR](#) 4
- #define [II_MUL](#) 5
- #define [II_SRL](#) 6
- #define [II_SLL](#) 7
- #define [II_SRA](#) 8

- #define [II_LB](#) 9
- #define [II_LH](#) 10
- #define [II_LW](#) 11
- #define [II_SB](#) 12
- #define [II_SH](#) 13
- #define [II_SW](#) 14
- #define [II_SFEQ](#) 15
- #define [II_SFNE](#) 16
- #define [II_SFLE](#) 17
- #define [II_SFLT](#) 18
- #define [II_SFGE](#) 19
- #define [II_SFGT](#) 20
- #define [II_BF](#) 21
- #define [II_LRBB](#) 22
- #define [II_CMOV](#) 23
- #define [II_REG](#) 24
- #define [II_NOP](#) 25
- #define [II_CALL](#) 26
- #define [II_LAST](#) 26
- #define [II_MASK](#) 0x0fff
- #define [II_MEM](#) 0x1000
- #define [II_SIGNED](#) 0x2000
- #define [II_IS_LOAD](#)(x) ((x) == II_LB || (x) == II_LH || (x) == II_LW)
- #define [II_IS_STORE](#)(x) ((x) == II_SB || (x) == II_SH || (x) == II_SW)
- #define [II_MEM_WIDTH](#)(x)

Functions

- double [ii_size](#) (int index, int imm)
- double [insn_time](#) (cuc_insn *ii)
- double [insn_size](#) (cuc_insn *ii)
- void [change_insn_type](#) (cuc_insn *i, int index)
- const char * [cuc_insn_name](#) (cuc_insn *ii)
- void [load_timing_table](#) (char *filename)
- void [print_shared](#) (cuc_func *rf, cuc_shared_item *shared, int nshared)

Variables

- const [cuc_known_insn](#) known [II_LAST+1]

6.71.1 Define Documentation

6.71.1.1 **#define II_ADD 0**

6.71.1.2 **#define II_AND 2**

6.71.1.3 **#define II_BF 21**

6.71.1.4 **#define II_CALL 26**

6.71.1.5 **#define II_CMOV 23**

6.71.1.6 **#define II_IS_LOAD(x) ((x) == II_LB || (x) == II_LH || (x) == II_LW)**

6.71.1.7 **#define II_IS_STORE(x) ((x) == II_SB || (x) == II_SH || (x) == II_SW)**

6.71.1.8 **#define II_LAST 26**

6.71.1.9 **#define II_LB 9**

6.71.1.10 **#define II_LH 10**

6.71.1.11 **#define II_LRBB 22**

6.71.1.12 **#define II_LW 11**

6.71.1.13 **#define II_MASK 0x0fff**

6.71.1.14 **#define II_MEM 0x1000**

6.71.1.15 **#define II_MEM_WIDTH(x)**

Value:

```
((x) == II_LB || (x) == II_SB) ? 1 : \  
    ((x) == II_LH || (x) == II_SH) ? 2 : \  
    ((x) == II_LW || (x) == II_SW) ? 4 : -1
```


6.71.1.16 #define II_MUL 5
6.71.1.17 #define II_NOP 25
6.71.1.18 #define II_OR 3
6.71.1.19 #define II_REG 24
6.71.1.20 #define II_SB 12
6.71.1.21 #define II_SFEQ 15
6.71.1.22 #define II_SFGE 19
6.71.1.23 #define II_SFGT 20
6.71.1.24 #define II_SFLE 17
6.71.1.25 #define II_SFLT 18
6.71.1.26 #define II_SFNE 16
6.71.1.27 #define II_SH 13
6.71.1.28 #define II_SIGNED 0x2000
6.71.1.29 #define II_SLL 7
6.71.1.30 #define II_SRA 8
6.71.1.31 #define II_SRL 6
6.71.1.32 #define II_SUB 1
6.71.1.33 #define II_SW 14
6.71.1.34 #define II_XOR 4

6.71.2 Function Documentation

6.71.2.1 void change_insn_type (cuc_insn * *i*, int *index*)
6.71.2.2 const char* cuc_insn_name (cuc_insn * *ii*)
6.71.2.3 double ii_size (int *index*, int *imm*)
6.71.2.4 double insn_size (cuc_insn * *ii*)
6.71.2.5 double insn_time (cuc_insn * *ii*)
6.71.2.6 void load_timing_table (char * *filename*)
6.71.2.7 void print_shared (cuc_func * *rf*, cuc_shared_item * *shared*, int *nshared*)

Here is the call graph for this function: Generated on 11 11:50:04 2008 for Or1ksim: The OpenRISC 1000 Architectural Simulator by Doxygen



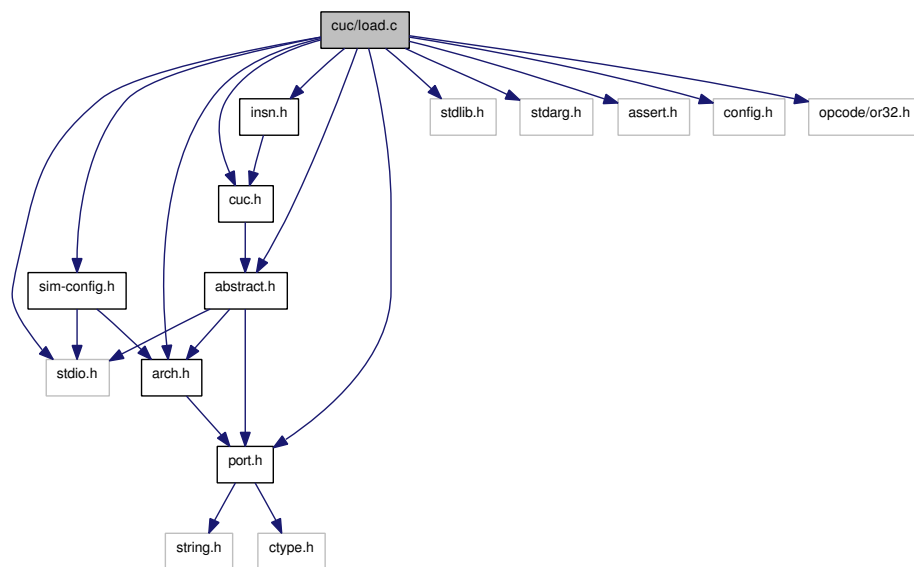
6.71.3 Variable Documentation

6.71.3.1 `const cuc_known_insn known[II_LAST+1]`

6.72 cuc/load.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <assert.h>
#include "config.h"
#include "port.h"
#include "arch.h"
#include "abstract.h"
#include "sim-config.h"
#include "cuc.h"
#include "opcode/or32.h"
#include "insn.h"
```

Include dependency graph for load.c:



Functions

- void [print_cuc_insns](#) (char *s, int verbose)
- void [xchg_insn](#) (int i, int j)
- void [negate_conditional](#) (cuc_insn *ii)
- void [remove_dslots](#) ()
- void [detect_locals](#) ()
- const char * [build_insn](#) (unsigned long data, cuc_insn *insn)
- void [expand_branch](#) ()
- void [expand_memory](#) ()

- void [expand_signed](#) ()
- void [expand_calls](#) ()
- int [cuc_load](#) (char *in_fn)

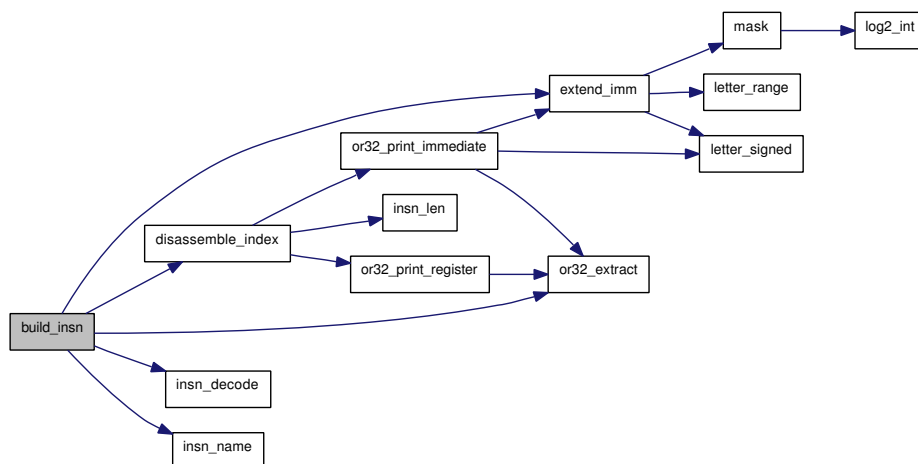
Variables

- static const [cuc_conv](#) conv []
- [cuc_insn](#) insn [MAX_INSNS]
- int [num_insn](#)
- int [reloc](#) [MAX_INSNS]

6.72.1 Function Documentation

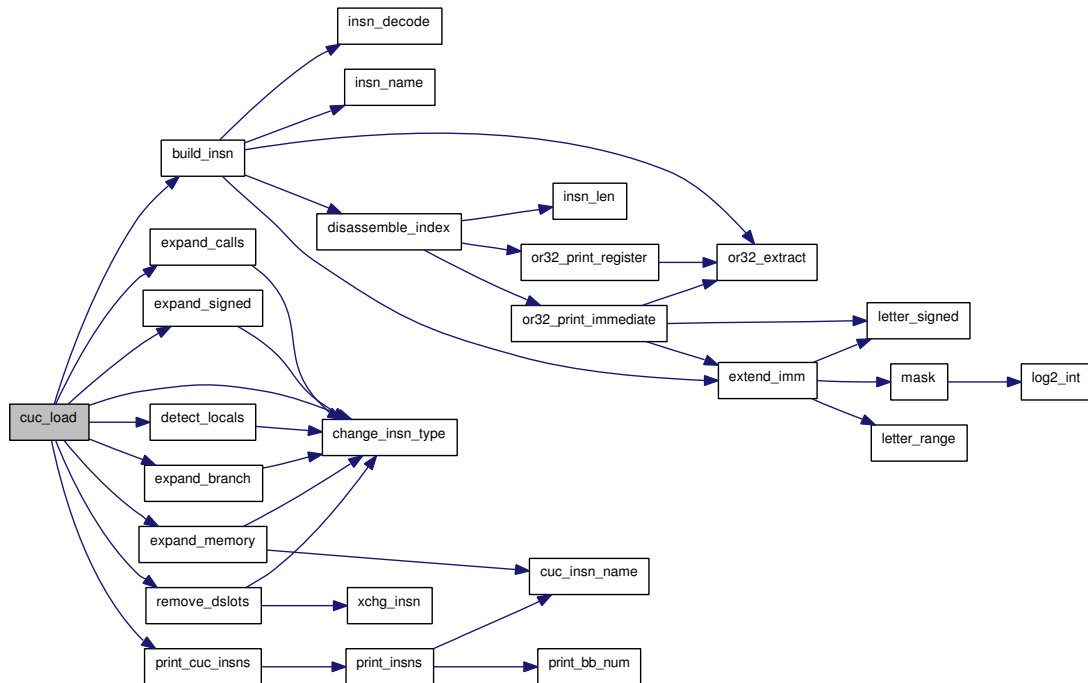
6.72.1.1 const char* [build_insn](#) (unsigned long *data*, [cuc_insn](#) * *insn*)

Here is the call graph for this function:



6.72.1.2 int cuc_load (char * in_fn)

Here is the call graph for this function:



6.72.1.3 void detect_locals ()

Here is the call graph for this function:



6.72.1.4 void expand_branch ()

Here is the call graph for this function:



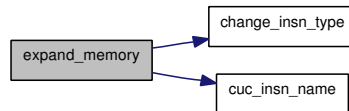
6.72.1.5 void expand_calls ()

Here is the call graph for this function:



6.72.1.6 void expand_memory ()

Here is the call graph for this function:



6.72.1.7 void expand_signed ()

Here is the call graph for this function:



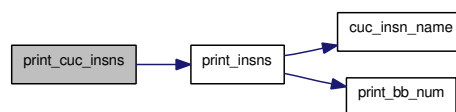
6.72.1.8 void negate_conditional (cuc_insn * ii)

Here is the call graph for this function:



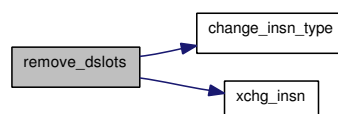
6.72.1.9 void print_cuc_insns (char * s, int verbose)

Here is the call graph for this function:



6.72.1.10 void remove_dslots ()

Here is the call graph for this function:



6.72.1.11 void xchg_insn (int *i*, int *j*)

6.72.2 Variable Documentation

6.72.2.1 const cuc_conv conv[] [static]

6.72.2.2 cuc_insn insn[MAX_INSNS]

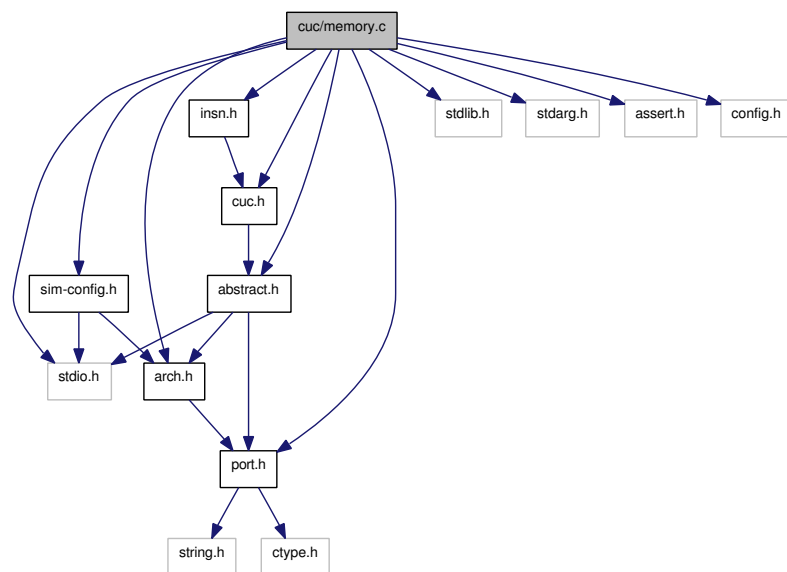
6.72.2.3 int num_insn

6.72.2.4 int reloc[MAX_INSNS]

6.73 cuc/memory.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <assert.h>
#include "config.h"
#include "port.h"
#include "arch.h"
#include "abstract.h"
#include "sim-config.h"
#include "cuc.h"
#include "insn.h"
```

Include dependency graph for memory.c:



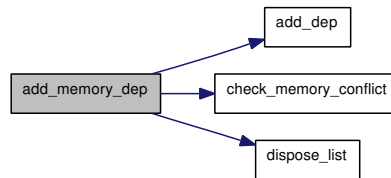
Functions

- void `clean_deps` (`cuc_func *f`)
- static int `check_memory_conflict` (`cuc_func *f`, `cuc_insn *a`, `cuc_insn *b`, int otype)
- void `add_memory_dep` (`cuc_func *f`, int otype)
- static int `same_transfers` (`cuc_func *f`, int otype)
- static int `join_transfers` (`cuc_func *f`, int otype)
- int `mem_ordering_cmp` (`cuc_func *f`, `cuc_insn *a`, `cuc_insn *b`)
- int `schedule_memory` (`cuc_func *f`, int otype)

6.73.1 Function Documentation

6.73.1.1 void add_memory_dep (cuc_func *f, int otype)

Here is the call graph for this function:

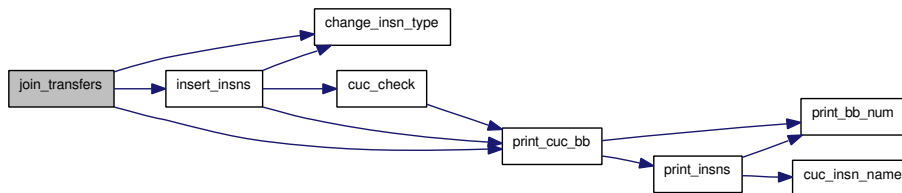


6.73.1.2 static int check_memory_conflict (cuc_func *f, cuc_insn *a, cuc_insn *b, int otype) [static]

6.73.1.3 void clean_deps (cuc_func *f)

6.73.1.4 static int join_transfers (cuc_func *f, int otype) [static]

Here is the call graph for this function:



6.73.1.5 int mem_ordering_cmp (cuc_func *f, cuc_insn *a, cuc_insn *b)

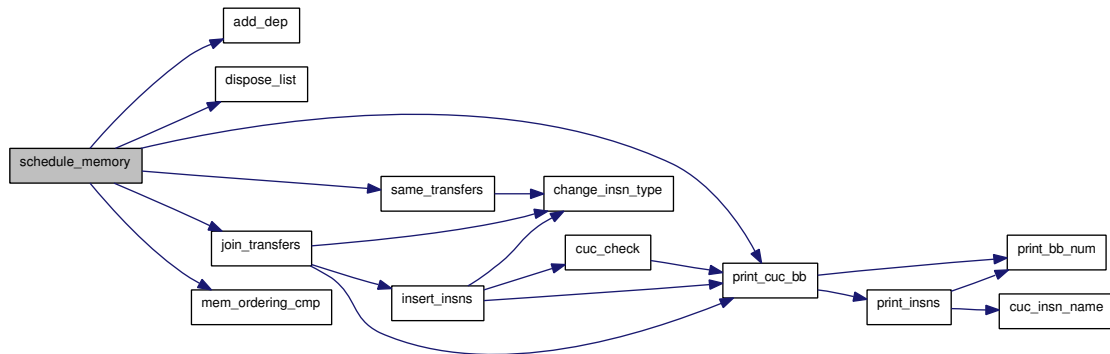
6.73.1.6 static int same_transfers (cuc_func *f, int otype) [static]

Here is the call graph for this function:



6.73.1.7 `int schedule_memory (cuc_func *f, int otype)`

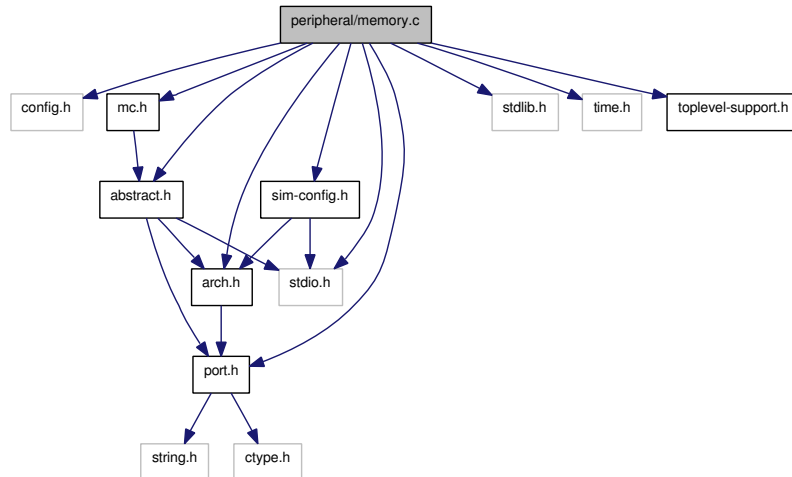
Here is the call graph for this function:



6.74 peripheral/memory.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "arch.h"
#include "sim-config.h"
#include "abstract.h"
#include "mc.h"
#include "toplevel-support.h"
```

Include dependency graph for memory.c:



Data Structures

- struct [mem_config](#)

Functions

- static uint32_t [simmem_read32](#) (oraddr_t addr, void *dat)
- static uint16_t [simmem_read16](#) (oraddr_t addr, void *dat)
- static uint8_t [simmem_read8](#) (oraddr_t addr, void *dat)
- static void [simmem_write32](#) (oraddr_t addr, uint32_t value, void *dat)
- static void [simmem_write16](#) (oraddr_t addr, uint16_t value, void *dat)
- static void [simmem_write8](#) (oraddr_t addr, uint8_t value, void *dat)
- static uint32_t [simmem_read_zero32](#) (oraddr_t addr, void *dat)
- static uint16_t [simmem_read_zero16](#) (oraddr_t addr, void *dat)
- static uint8_t [simmem_read_zero8](#) (oraddr_t addr, void *dat)

- static void `simmem_write_null32` (`oraddr_t` `addr`, `uint32_t` `value`, `void *dat`)
- static void `simmem_write_null16` (`oraddr_t` `addr`, `uint16_t` `value`, `void *dat`)
- static void `simmem_write_null8` (`oraddr_t` `addr`, `uint8_t` `value`, `void *dat`)
- static void `mem_reset` (`void *dat`)
- static void `memory_random_seed` (`union param_val` `val`, `void *dat`)
- static void `memory_pattern` (`union param_val` `val`, `void *dat`)
- static void `memory_type` (`union param_val` `val`, `void *dat`)
- static void `memory_ce` (`union param_val` `val`, `void *dat`)
- static void `memory_mc` (`union param_val` `val`, `void *dat`)
- static void `memory_baseaddr` (`union param_val` `val`, `void *dat`)
- static void `memory_size` (`union param_val` `val`, `void *dat`)
- static void `memory_name` (`union param_val` `val`, `void *dat`)
- static void `memory_log` (`union param_val` `val`, `void *dat`)
- static void `memory_delayr` (`union param_val` `val`, `void *dat`)
- static void `memory_delayw` (`union param_val` `val`, `void *dat`)
- static void * `memory_sec_start` ()
- static void `memory_sec_end` (`void *dat`)
- void `reg_memory_sec` (`void`)

6.74.1 Function Documentation

6.74.1.1 static void `mem_reset` (`void * dat`) [static]

6.74.1.2 static void `memory_baseaddr` (`union param_val val`, `void * dat`) [static]

6.74.1.3 static void `memory_ce` (`union param_val val`, `void * dat`) [static]

6.74.1.4 static void `memory_delayr` (`union param_val val`, `void * dat`) [static]

6.74.1.5 static void `memory_delayw` (`union param_val val`, `void * dat`) [static]

6.74.1.6 static void `memory_log` (`union param_val val`, `void * dat`) [static]

6.74.1.7 static void `memory_mc` (`union param_val val`, `void * dat`) [static]

6.74.1.8 static void `memory_name` (`union param_val val`, `void * dat`) [static]

6.74.1.9 static void `memory_pattern` (`union param_val val`, `void * dat`) [static]

Set the memory pattern

Value must be up to 8 bits. Larger values are truncated with a warning.

Parameters:

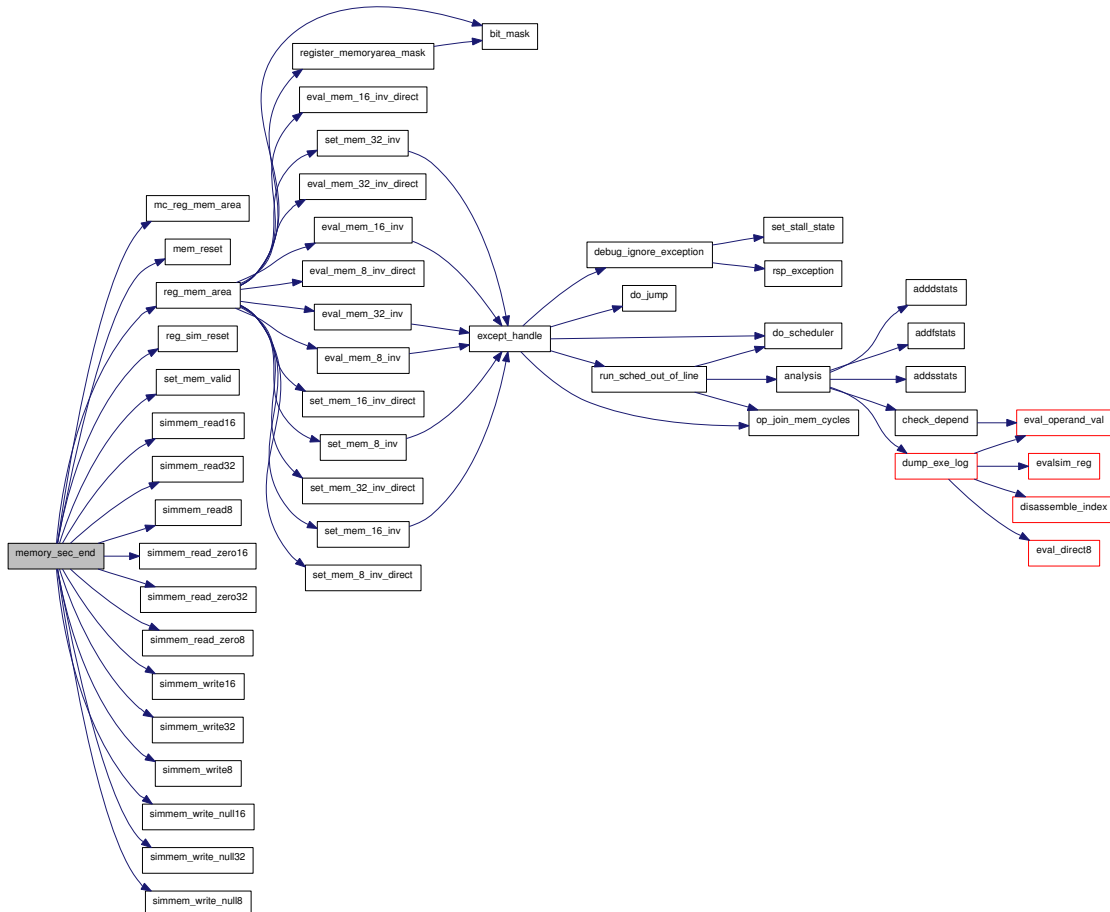
← *val* The value to use

← *dat* The `config` data structure

6.74.1.10 `static void memory_random_seed (union param_val val, void * dat)` [static]

6.74.1.11 `static void memory_sec_end (void * dat)` [static]

Here is the call graph for this function:



6.74.1.12 `static void* memory_sec_start ()` [static]

Initialize a new block of memory configuration

ALL parameters are set explicitly to default values.

Returns:

The new memory configuration data structure

6.74.1.13 `static void memory_size (union param_val val, void * dat)` [static]

6.74.1.14 `static void memory_type (union param_val val, void * dat)` [static]

Set the memory type

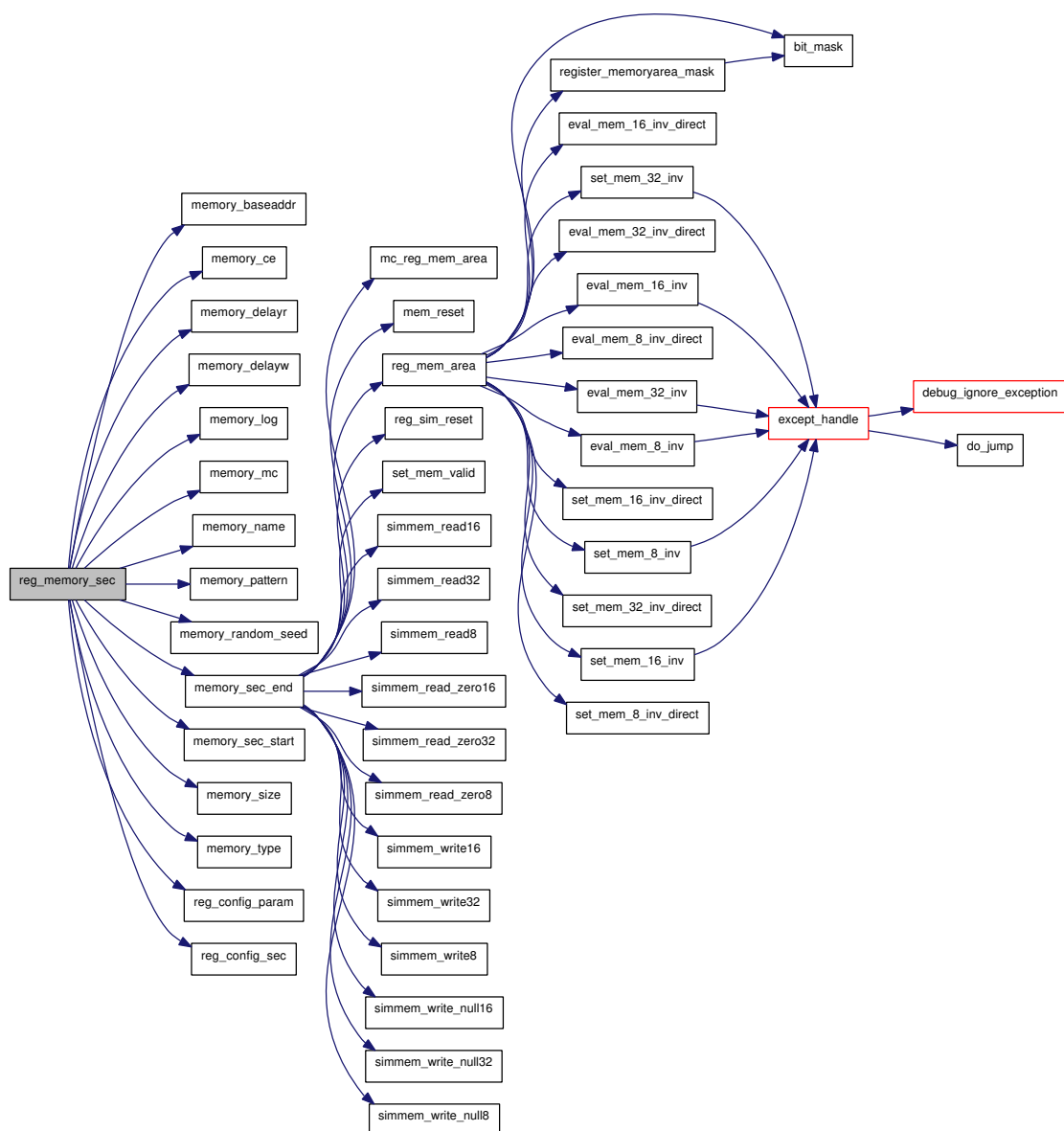
Value must be one of unknown, random, pattern or zero (case insensitive). Unrecognized values are ignored with a warning.

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure

6.74.1.15 void reg_memory_sec (void)

Here is the call graph for this function:

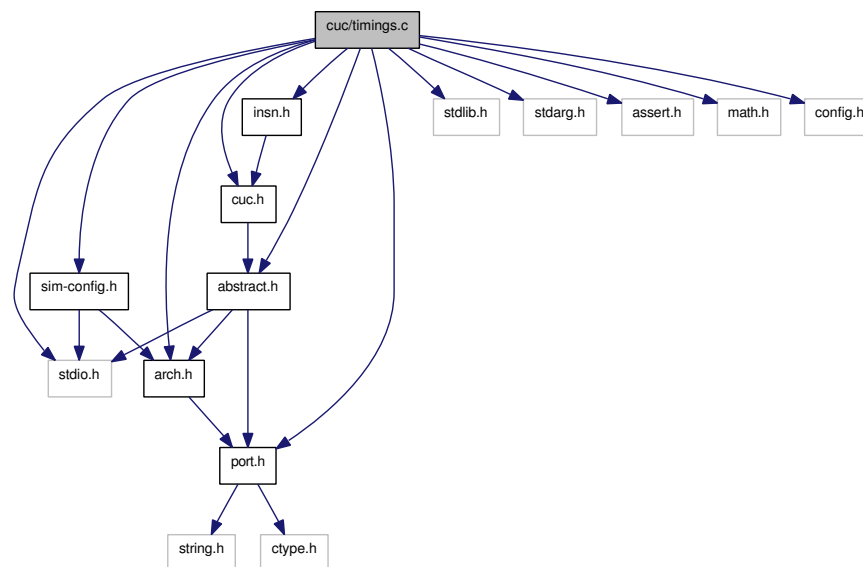


- 6.74.1.16 `static uint16_t simmem_read16 (oraddr_t addr, void * dat) [static]`
- 6.74.1.17 `static uint32_t simmem_read32 (oraddr_t addr, void * dat) [static]`
- 6.74.1.18 `static uint8_t simmem_read8 (oraddr_t addr, void * dat) [static]`
- 6.74.1.19 `static uint16_t simmem_read_zero16 (oraddr_t addr, void * dat) [static]`
- 6.74.1.20 `static uint32_t simmem_read_zero32 (oraddr_t addr, void * dat) [static]`
- 6.74.1.21 `static uint8_t simmem_read_zero8 (oraddr_t addr, void * dat) [static]`
- 6.74.1.22 `static void simmem_write16 (oraddr_t addr, uint16_t value, void * dat) [static]`
- 6.74.1.23 `static void simmem_write32 (oraddr_t addr, uint32_t value, void * dat) [static]`
- 6.74.1.24 `static void simmem_write8 (oraddr_t addr, uint8_t value, void * dat) [static]`
- 6.74.1.25 `static void simmem_write_null16 (oraddr_t addr, uint16_t value, void * dat)
[static]`
- 6.74.1.26 `static void simmem_write_null32 (oraddr_t addr, uint32_t value, void * dat)
[static]`
- 6.74.1.27 `static void simmem_write_null8 (oraddr_t addr, uint8_t value, void * dat) [static]`

6.75 cuc/timings.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <assert.h>
#include <math.h>
#include "config.h"
#include "port.h"
#include "arch.h"
#include "abstract.h"
#include "sim-config.h"
#include "cuc.h"
#include "insn.h"
```

Include dependency graph for timings.c:



Functions

- double [insn_time](#) (cuc_insn *ii)
- double [insn_size](#) (cuc_insn *ii)
- double [ii_size](#) (int index, int imm)
- static double [max_delay](#) (cuc_func *f, int b)
- static int [memory_delay](#) (cuc_func *f, int b)
- void [cut_tree](#) (cuc_func *f, int b, double sd)
- static int [new_bb_cycles](#) (cuc_func *f, int b, int cut)
- void [mark_cut](#) (cuc_func *f)

- static double `bb_size` (`cuc_bb *bb`)
- void `recalc_cnts` (`cuc_func *f`, `char *bb_filename`)
- void `analyse_timings` (`cuc_func *f`, `cuc_timings *timings`)
- void `load_timing_table` (`char *filename`)

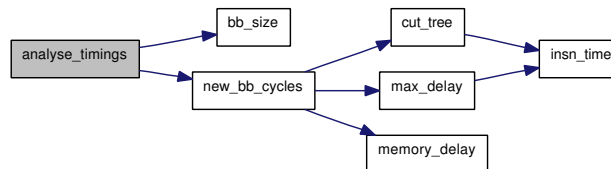
Variables

- static `cuc_timing_table * timing_table`
- static double `max_bb_delay`

6.75.1 Function Documentation

6.75.1.1 void `analyse_timings` (`cuc_func *f`, `cuc_timings *timings`)

Here is the call graph for this function:



6.75.1.2 static double `bb_size` (`cuc_bb *bb`) [static]

6.75.1.3 void `cut_tree` (`cuc_func *f`, `int b`, `double sd`)

Here is the call graph for this function:



6.75.1.4 `double ii_size (int index, int imm)`

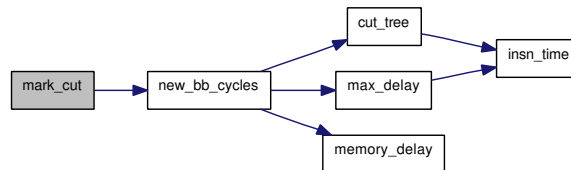
6.75.1.5 `double insn_size (cuc_insn * ii)`

6.75.1.6 `double insn_time (cuc_insn * ii)`

6.75.1.7 `void load_timing_table (char * filename)`

6.75.1.8 `void mark_cut (cuc_func * f)`

Here is the call graph for this function:



6.75.1.9 `static double max_delay (cuc_func * f, int b)` [static]

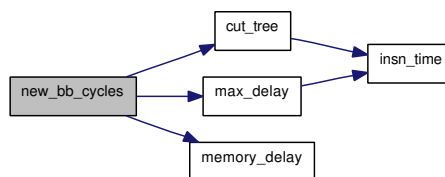
Here is the call graph for this function:



6.75.1.10 `static int memory_delay (cuc_func * f, int b)` [static]

6.75.1.11 `static int new_bb_cycles (cuc_func * f, int b, int cut)` [static]

Here is the call graph for this function:



6.75.1.12 `void recalc_cnts (cuc_func * f, char * bb_filename)`

6.75.2 Variable Documentation

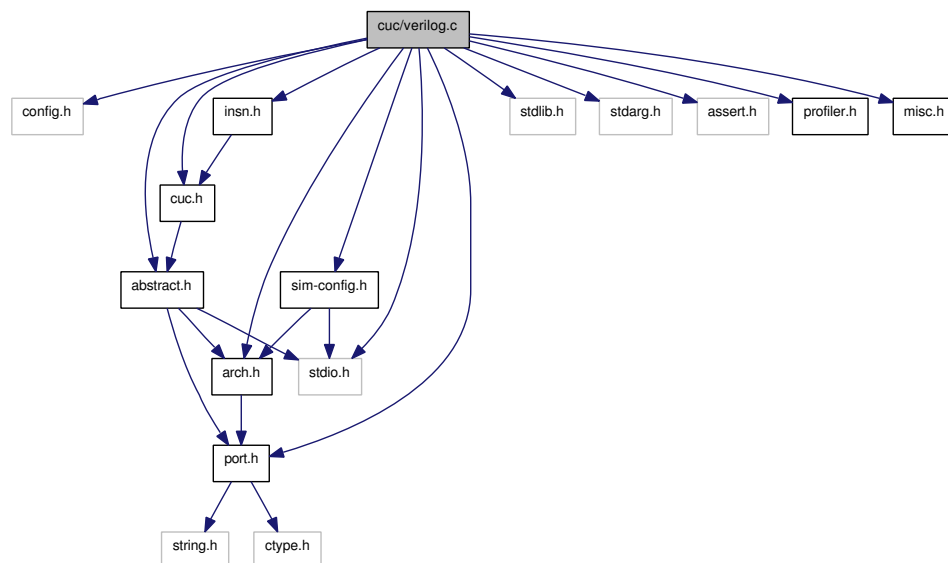
6.75.2.1 `double max_bb_delay` [static]

6.75.2.2 `cuc_timing_table* timing_table` [static]

6.76 cuc/verilog.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <assert.h>
#include "arch.h"
#include "abstract.h"
#include "cuc.h"
#include "insn.h"
#include "profiler.h"
#include "sim-config.h"
#include "misc.h"
```

Include dependency graph for verilog.c:



Defines

- #define [GEN\(x...\)](#) `fprintf (fo, x)`

Functions

- static int [find_lsc_index](#) (`cuc_func *f`, int ref)
- static void [print_deps](#) (`FILE *fo`, `cuc_func *f`, int b, `dep_list *t`, int registered)
- static char * [print_op_v](#) (`cuc_func *f`, char *s, int ref, int j)

- static void `print_insn_v` (FILE *fo, cuc_func *f, int b, int i)
- static int `branch_index` (cuc_bb *bb)
- static void `print_turn_off_dep` (FILE *fo, cuc_func *f, dep_list *dep)
- static int `func_index` (cuc_func *f, int ref)
- void `output_verilog` (cuc_func *f, char *filename, char *funcname)
- void `generate_main` (int nfuncs, cuc_func **f, char *filename)

6.76.1 Define Documentation

6.76.1.1 `#define GEN(x...) fprintf (fo, x)`

6.76.2 Function Documentation

6.76.2.1 static int `branch_index` (cuc_bb *bb) [static]

6.76.2.2 static int `find_lsc_index` (cuc_func *f, int ref) [static]

6.76.2.3 static int `func_index` (cuc_func *f, int ref) [static]

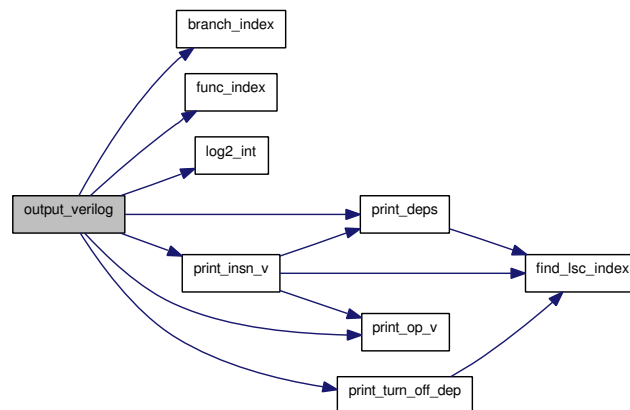
6.76.2.4 void `generate_main` (int nfuncs, cuc_func **f, char *filename)

Here is the call graph for this function:



6.76.2.5 void `output_verilog` (cuc_func *f, char *filename, char *funcname)

Here is the call graph for this function:



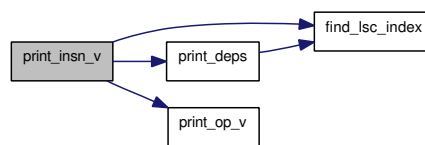
6.76.2.6 `static void print_deps (FILE *fo, cuc_func *f, int b, dep_list *t, int registered)`
[static]

Here is the call graph for this function:



6.76.2.7 `static void print_insn_v (FILE *fo, cuc_func *f, int b, int i)` [static]

Here is the call graph for this function:



6.76.2.8 `static char* print_op_v (cuc_func *f, char *s, int ref, int j)` [static]

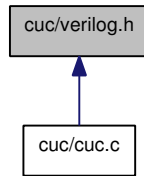
6.76.2.9 `static void print_turn_off_dep (FILE *fo, cuc_func *f, dep_list *dep)` [static]

Here is the call graph for this function:



6.77 cuc/verilog.h File Reference

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



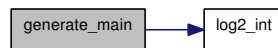
Functions

- void [output_verilog](#) ([cuc_func](#) *func, char *filename, char *funcname)
- void [generate_main](#) (int nfuncs, [cuc_func](#) **f, char *filename)

6.77.1 Function Documentation

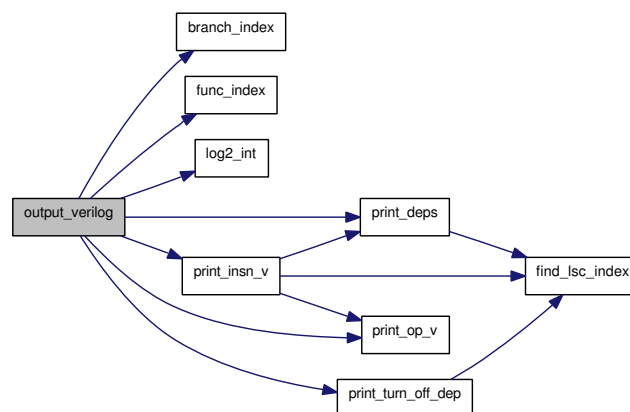
6.77.1.1 void generate_main (int nfuncs, cuc_func **f, char *filename)

Here is the call graph for this function:



6.77.1.2 void output_verilog (cuc_func *func, char *filename, char *funcname)

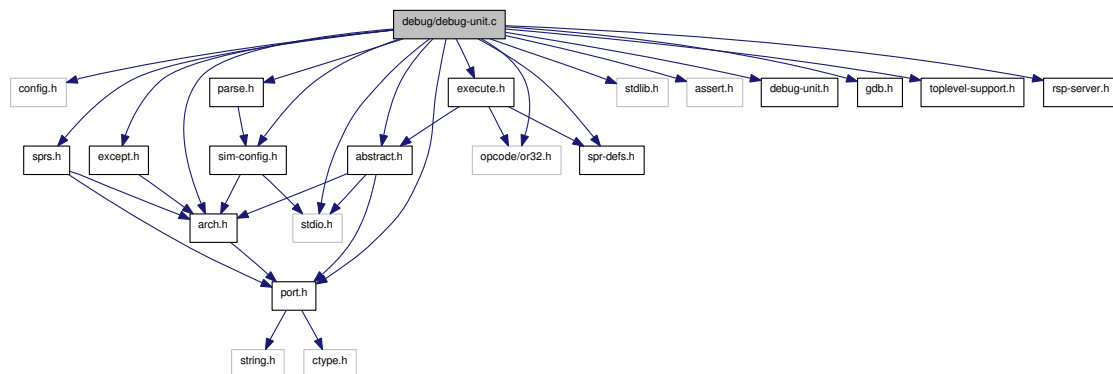
Here is the call graph for this function:



6.78 debug/debug-unit.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include <assert.h>
#include "arch.h"
#include "debug-unit.h"
#include "sim-config.h"
#include "except.h"
#include "abstract.h"
#include "parse.h"
#include "gdb.h"
#include "opcode/or32.h"
#include "spr-defs.h"
#include "execute.h"
#include "sprs.h"
#include "toplevel-support.h"
#include "rsp-server.h"
```

Include dependency graph for debug-unit.c:



Defines

- #define [RISCOP_STALL](#) 0x00000001
- #define [RISCOP_RESET](#) 0x00000002

Enumerations

- enum `development_interface_address_space` { `DEVELOPINT_RISCOP` = 4, `DEVELOPINT_MAX` = 27 }

Functions

- static int `calculate_watchpoints` (enum `debug_unit_action` action, unsigned long udata)
- static int `get_devint_reg` (unsigned int addr, unsigned long *data)
- static int `set_devint_reg` (unsigned int addr, unsigned long data)
- static int `debug_set_mem` (`oraddr_t` address, `uorreg_t` data)
- static int `debug_get_mem` (`oraddr_t` address, `uorreg_t` *data)
- void `du_reset` ()
- void `set_stall_state` (int state)
- int `check_debug_unit` (enum `debug_unit_action` action, unsigned long udata)
- int `debug_get_register` (`oraddr_t` address, `uorreg_t` *data)
- int `debug_set_register` (`oraddr_t` address, `uorreg_t` data)
- int `debug_set_chain` (enum `debug_scan_chain_ids` chain)
- static int `get_devint_reg` (enum `development_interface_address_space` address, unsigned long *data)
- static int `set_devint_reg` (enum `development_interface_address_space` address, unsigned long data)
- int `debug_ignore_exception` (unsigned long except)
- static void `debug_enabled` (union `param_val` val, void *dat)
- static void `debug_gdb_enabled` (union `param_val` val, void *dat)
- static void `debug_rsp_enabled` (union `param_val` val, void *dat)
- static void `debug_server_port` (union `param_val` val, void *dat)
- static void `debug_rsp_port` (union `param_val` val, void *dat)
- static void `debug_vapi_id` (union `param_val` val, void *dat)
- void `reg_debug_sec` ()

Variables

- unsigned long `development` [`DEVELOPINT_MAX`+1]
- static enum `debug_scan_chain_ids` `current_scan_chain` = `JTAG_CHAIN_GLOBAL`
- static int `in_reset` = 0

6.78.1 Define Documentation

6.78.1.1 #define RISCOP_RESET 0x00000002

Reset processor (clears stall)

6.78.1.2 #define RISCOP_STALL 0x00000001

The fields for the RISCOP register in the development interface scan chain (`JTAG_CHAIN_DEVELOPMENT`). Stall processor

6.78.2 Enumeration Type Documentation

6.78.2.1 enum development_interface_address_space

The various addresses in the development interface scan chain (JTAG_CHAIN_DEVELOPMENT). Only documents the ones we actually have

Enumerator:

DEVELOPINT_RISCOPE
DEVELOPINT_MAX

6.78.3 Function Documentation

6.78.3.1 static int calculate_watchpoints (enum debug_unit_action action, unsigned long udata) [static]

Forward declaration of static functions

Check whether we should stall the RISC or cause an exception.

Rewritten by JPB for current architecture.

Parameters:

- ← *action* The action to be checked
- ← *udata* The data to compare against (for some actions)

Returns:

Non-zero if this should generate a breakpoint

6.78.3.2 int check_debug_unit (enum debug_unit_action action, unsigned long udata)

Check for a breakpoint on this action

Note:

This does not include single-stepping - that will be picked up in the main loop AFTER the instruction has executed.

Parameters:

- ← *action* The action to be checked
- ← *udata* The data to compare against (for some actions)

Returns:

Non-zero if there was a breakpoint, 0 otherwise.

Here is the call graph for this function:



Parameters:

- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

6.78.3.9 static void debug_rsp_port (union param_val val, void * dat) [static]

Set the Remote Serial Protocol GDB server port

This is for use with the RSP, which is now the preferred interface. Ensure the value chosen is valid. Note that 0 is permitted, meaning the connection should be to the "or1ksim-rsp" service, rather than a port.

Both this and the legacy port may be specified, but only one may be enabled (see [debug_gdb_enabled\(\)](#) and [debug_rsp_enabled\(\)](#)).

Parameters:

- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

6.78.3.10 static void debug_server_port (union param_val val, void * dat) [static]

Set the legacy GDB server port

This is for use with the OpenRISC Remote JTAG protocol (now deprecated). Ensure the value chosen is valid. Note that 0 is permitted, meaning the connection should be to the "or1ksim" service, rather than a port.

Both this and the RSP port may be specified, but only one may be enabled (see [debug_gdb_enabled\(\)](#) and [debug_rsp_enabled\(\)](#)).

Parameters:

- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

6.78.3.11 int debug_set_chain (enum debug_scan_chain_ids chain)

Set the JTAG chain

Only permit chains we support. Currently TRACE is not implemented.

Parameters:

- ← *chain* Chain to be set as current

Returns:

An error code (including ERR_NONE) if there is no error

6.78.3.12 static int debug_set_mem (oraddr_t address, uint32_t data) [static]

Write to main bus

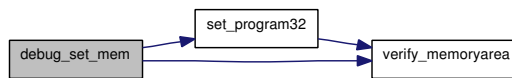
Parameters:

- ← *address* Address to write to
- *data* Data to write

Returns:

An error code (including ERR_NONE) if there is no error

Here is the call graph for this function:



6.78.3.13 int debug_set_register (oraddr_t address, uoreg_t data)

Set a JTAG register

Action depends on which scan chain is currently active.

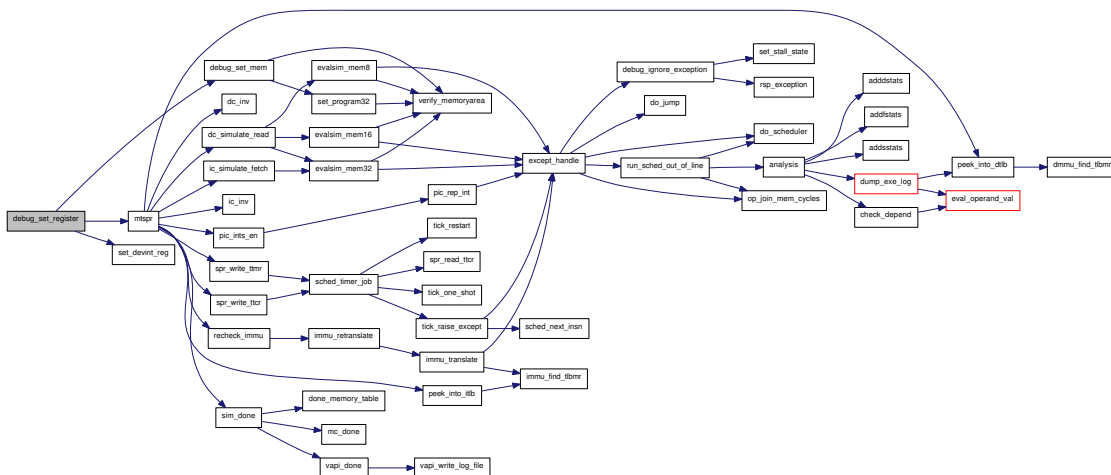
Parameters:

- ← *address* Address on the scan chain
- *data* Data to set

Returns:

An error code (including ERR_NONE) if there is no error

Here is the call graph for this function:



6.78.3.14 `static void debug_vapi_id (union param_val val, void * dat)` [static]

Set the VAPI ID for the debug unit

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure (not used here)

6.78.3.15 `void du_reset ()`

Reset the debug unit

Clear all development interface registers

Here is the call graph for this function:

**6.78.3.16** `static int get_devint_reg (enum development_interface_address_space address, unsigned long * data)` [static]

Get a development interface register

No side effects on get - just return the register

Parameters:

- ← *address* The register to get
- *data* Where to put the result

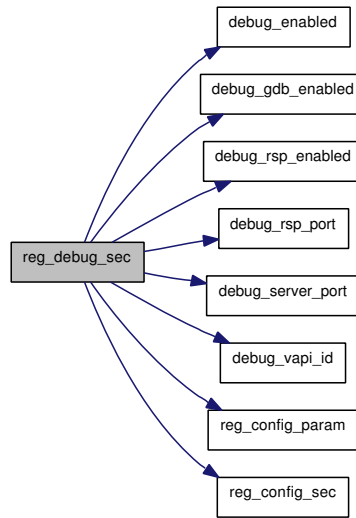
Returns:

An error code (including `ERR_NONE`) if there is no error

6.78.3.17 `static int get_devint_reg (unsigned int addr, unsigned long * data)` [static]**6.78.3.18** `void reg_debug_sec ()`

Register the configuration functions for the debug unit

Here is the call graph for this function:



6.78.3.19 static int set_devint_reg (enum development_interface_address_space address, unsigned long data) [static]

Set a development interface register

Sets the value of the corresponding register. Only RISC_OP has any side-effects. The others just store the value, so it can be read back.

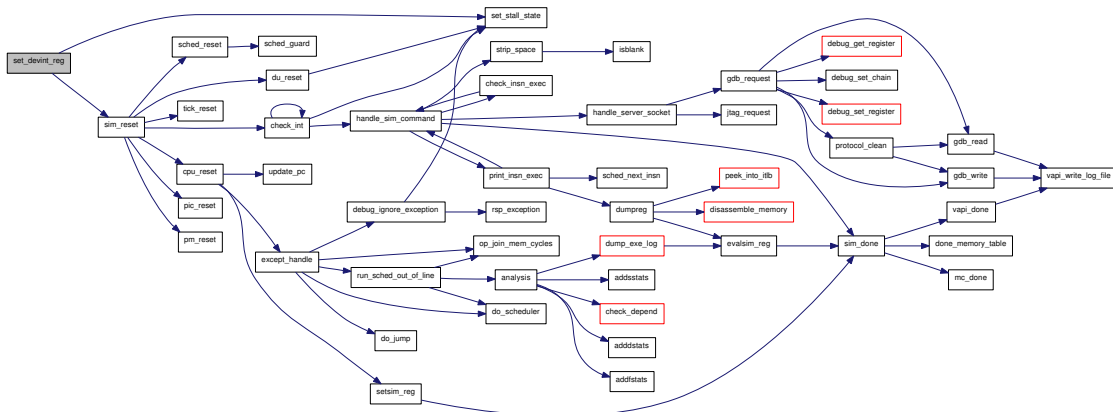
Parameters:

- ← **address** The register to set
- ← **data** The data to set

Returns:

An error code (including ERR_NONE) if there is no error

Here is the call graph for this function:



6.78.3.20 `static int set_devint_reg (unsigned int addr, unsigned long data)` [static]

6.78.3.21 `void set_stall_state (int state)`

Set the stall state of the processor

Parameters:

← *state* If non-zero stall the processor.

6.78.4 Variable Documentation

6.78.4.1 `enum debug_scan_chain_ids current_scan_chain = JTAG_CHAIN_GLOBAL`
[static]

The current scan chain being accessed

6.78.4.2 `unsigned long development[DEVELOPINT_MAX+1]`

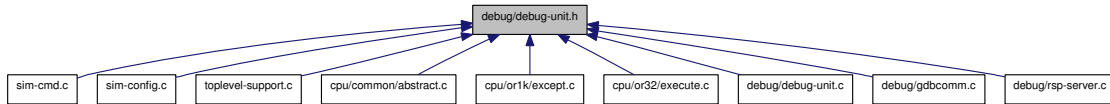
Data structure holding debug registers and their bits

6.78.4.3 `int in_reset = 0` [static]

External STALL signal to debug interface

6.79 debug/debug-unit.h File Reference

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



Enumerations

- enum `debug_unit_action` {
`DebugInstructionFetch = 1, DebugLoadAddress = 2, DebugStoreAddress = 3, DebugLoadData = 4, DebugStoreData = 5` }
- enum `debug_scan_chain_ids` {
`JTAG_CHAIN_GLOBAL = 0, JTAG_CHAIN_DEBUG_UNIT = 1, JTAG_CHAIN_TRACE = 3, JTAG_CHAIN_DEVELOPMENT = 4, JTAG_CHAIN_WISHBONE = 5` }

Functions

- void `du_reset` ()
- void `set_stall_state` (int state)
- int `check_debug_unit` (enum `debug_unit_action` action, unsigned long udata)
- int `debug_get_register` (oraddr_t address, uorreg_t *data)
- int `debug_set_register` (oraddr_t address, uorreg_t data)
- int `debug_set_chain` (enum `debug_scan_chain_ids` chain)
- int `debug_ignore_exception` (unsigned long except)
- void `reg_debug_sec` ()

6.79.1 Enumeration Type Documentation

6.79.1.1 enum debug_scan_chain_ids

Enumeration of the various JTAG scan chains. Only those actually implemented are specified.

Enumerator:

JTAG_CHAIN_GLOBAL
JTAG_CHAIN_DEBUG_UNIT
JTAG_CHAIN_TRACE
JTAG_CHAIN_DEVELOPMENT
JTAG_CHAIN_WISHBONE

6.79.1.2 enum debug_unit_action

The possible debug unit actions

Enumerator:

DebugInstructionFetch

DebugLoadAddress

DebugStoreAddress

DebugLoadData

DebugStoreData

6.79.2 Function Documentation

6.79.2.1 int check_debug_unit (enum debug_unit_action *action*, unsigned long *udata*)

Check for a breakpoint on this action

Note:

This does not include single-stepping - that will be picked up in the main loop AFTER the instruction has executed.

Parameters:

← *action* The action to be checked

← *udata* The data to compare against (for some actions)

Returns:

Non-zero if there was a breakpoint, 0 otherwise.

Here is the call graph for this function:



6.79.2.2 int debug_get_register (oraddr_t *address*, uorreg_t * *data*)

Get a JTAG register

Action depends on which scan chain is currently active.

Parameters:

← *address* Address on the scan chain

→ *data* Where to put the result of the read

Returns:

An error code (including ERR_NONE) if there is no error

Parameters:

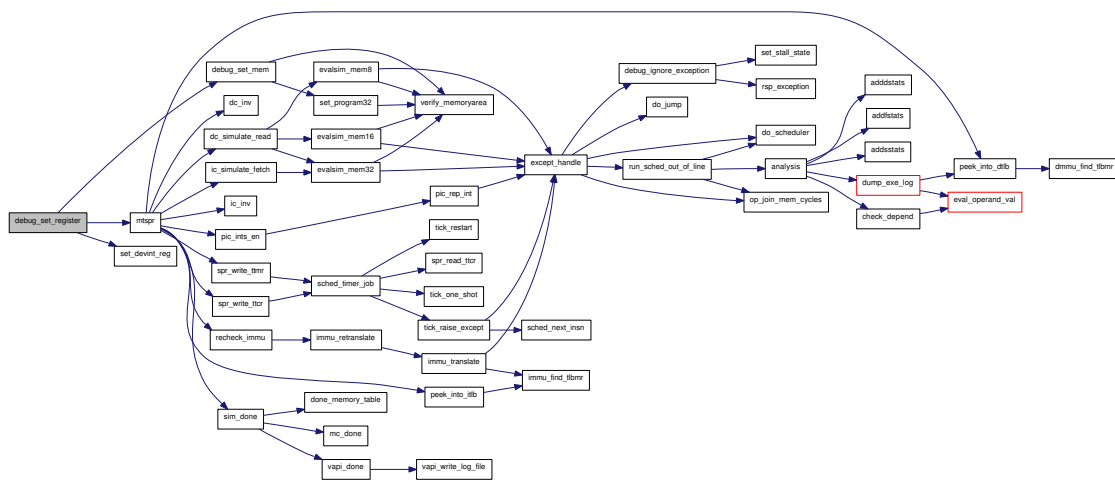
← *address* Address on the scan chain

→ *data* Data to set

Returns:

An error code (including ERR_NONE) if there is no error

Here is the call graph for this function:

**6.79.2.6 void du_reset ()**

Reset the debug unit

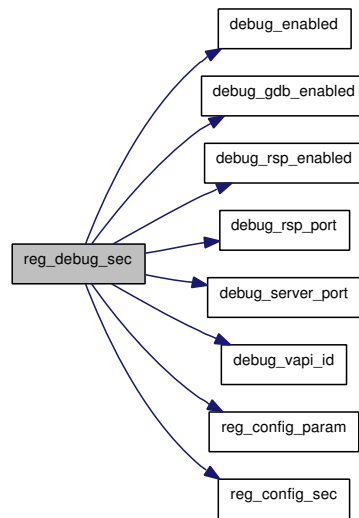
Clear all development interface registers

Here is the call graph for this function:

**6.79.2.7 void reg_debug_sec ()**

Register the configuration functions for the debug unit

Here is the call graph for this function:



6.79.2.8 void set_stall_state (int state)

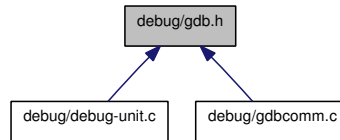
Set the stall state of the processor

Parameters:

← *state* If non-zero stall the processor.

6.80 debug/gdb.h File Reference

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



Data Structures

- struct [jtr_read_message](#)
- struct [jtr_write_message](#)
- struct [jtr_read_block_message](#)
- struct [jtr_write_block_message](#)
- struct [jtr_chain_message](#)
- struct [jtr_failure_response](#)
- struct [jtr_read_response](#)
- struct [jtr_write_response](#)
- struct [jtr_read_block_response](#)
- struct [jtr_write_block_response](#)
- struct [jtr_chain_response](#)

Enumerations

- enum [or1k_jtag_errors](#) {
 - [ERR_NONE](#) = 0, [ERR_CRC](#) = -1, [ERR_MEM](#) = -2, [JTAG_PROXY_INVALID_COMMAND](#) = -3,
 - [JTAG_PROXY_SERVER_TERMINATED](#) = -4, [JTAG_PROXY_NO_CONNECTION](#) = -5, [JTAG_PROXY_PROTOCOL_ERROR](#) = -6, [JTAG_PROXY_COMMAND_NOT_IMPLEMENTED](#) = -7,
 - [JTAG_PROXY_INVALID_CHAIN](#) = -8, [JTAG_PROXY_INVALID_ADDRESS](#) = -9, [JTAG_PROXY_ACCESS_EXCEPTION](#) = -10, [JTAG_PROXY_INVALID_LENGTH](#) = -11,
 - [JTAG_PROXY_OUT_OF_MEMORY](#) = -12 }
- enum [or1k_jtag_proxy_protocol_commands](#) {
 - [OR1K_JTAG_COMMAND_READ](#) = 1, [OR1K_JTAG_COMMAND_WRITE](#) = 2, [OR1K_JTAG_COMMAND_READ_BLOCK](#) = 3, [OR1K_JTAG_COMMAND_WRITE_BLOCK](#) = 4,
 - [OR1K_JTAG_COMMAND_CHAIN](#) = 5 }

6.80.1 Enumeration Type Documentation

6.80.1.1 enum or1k_jtag_errors

Error codes for the OpenRISC 1000 JTAG debugging protocol

Enumerator:

ERR_NONE

ERR_CRC

ERR_MEM
JTAG_PROXY_INVALID_COMMAND
JTAG_PROXY_SERVER_TERMINATED
JTAG_PROXY_NO_CONNECTION
JTAG_PROXY_PROTOCOL_ERROR
JTAG_PROXY_COMMAND_NOT_IMPLEMENTED
JTAG_PROXY_INVALID_CHAIN
JTAG_PROXY_INVALID_ADDRESS
JTAG_PROXY_ACCESS_EXCEPTION
JTAG_PROXY_INVALID_LENGTH
JTAG_PROXY_OUT_OF_MEMORY

6.80.1.2 enum or1k_jtag_proxy_protocol_commands

The OR1K JTAG proxy protocol commands.

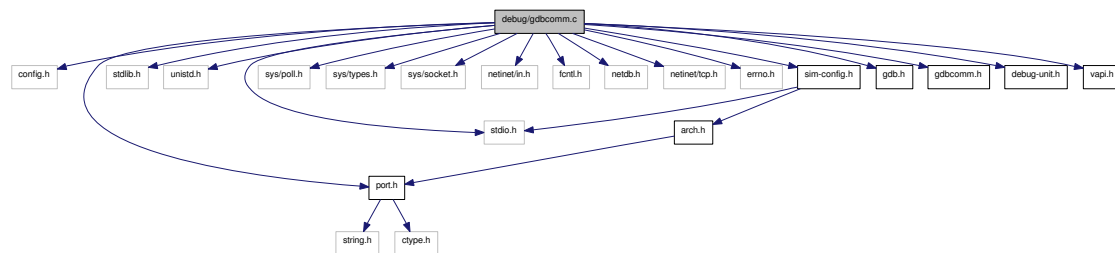
Enumerator:

ORIK_JTAG_COMMAND_READ
ORIK_JTAG_COMMAND_WRITE
ORIK_JTAG_COMMAND_READ_BLOCK
ORIK_JTAG_COMMAND_WRITE_BLOCK
ORIK_JTAG_COMMAND_CHAIN

6.81 debug/gdbcomm.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/poll.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <netdb.h>
#include <netinet/tcp.h>
#include <errno.h>
#include "sim-config.h"
#include "gdb.h"
#include "gdbcomm.h"
#include "debug-unit.h"
#include "vapi.h"
```

Include dependency graph for gdbcomm.c:



Functions

- static void [jtag_request](#) ()
- static void [gdb_request](#) (void)
- static void [protocol_clean](#) (int, int32_t)
- static int [get_server_socket](#) (const char *name, const char *proto, int port)
- static int [gdb_read](#) (void *buf, int len)
- static int [gdb_write](#) (const void *buf, int len)
- void [block_jtag](#) ()
- void [handle_server_socket](#) (enum boolean block)
- void [gdbcomm_init](#) ()

Variables

- static unsigned int `server_ip` = 0
- static unsigned int `server_port` = 0
- static unsigned int `server_fd` = 0
- static unsigned int `gdb_fd` = 0
- static int `tcp_level` = 0

6.81.1 Function Documentation

6.81.1.1 void `block_jtag` ()

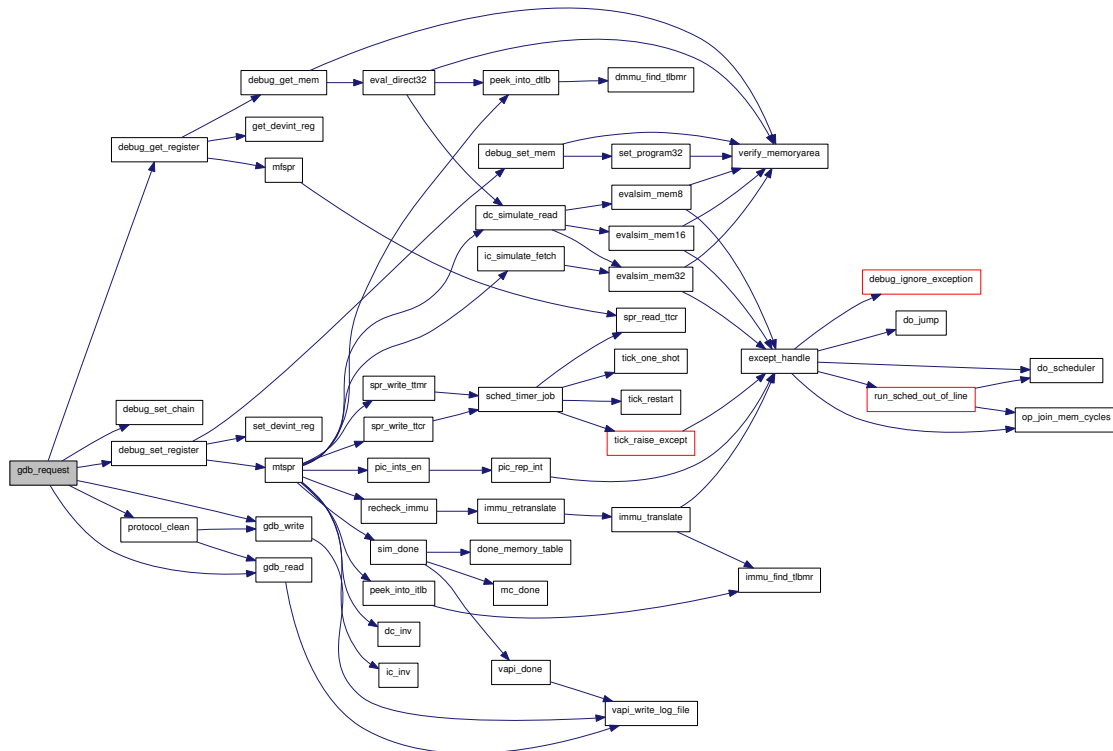
6.81.1.2 static int `gdb_read` (void * *buf*, int *len*) [static]

Here is the call graph for this function:



6.81.1.3 static void `gdb_request` (void) [static]

Here is the call graph for this function:



6.81.1.4 static int gdb_write (const void * buf, int len) [static]

Here is the call graph for this function:



6.81.1.5 void gdbcomm_init ()

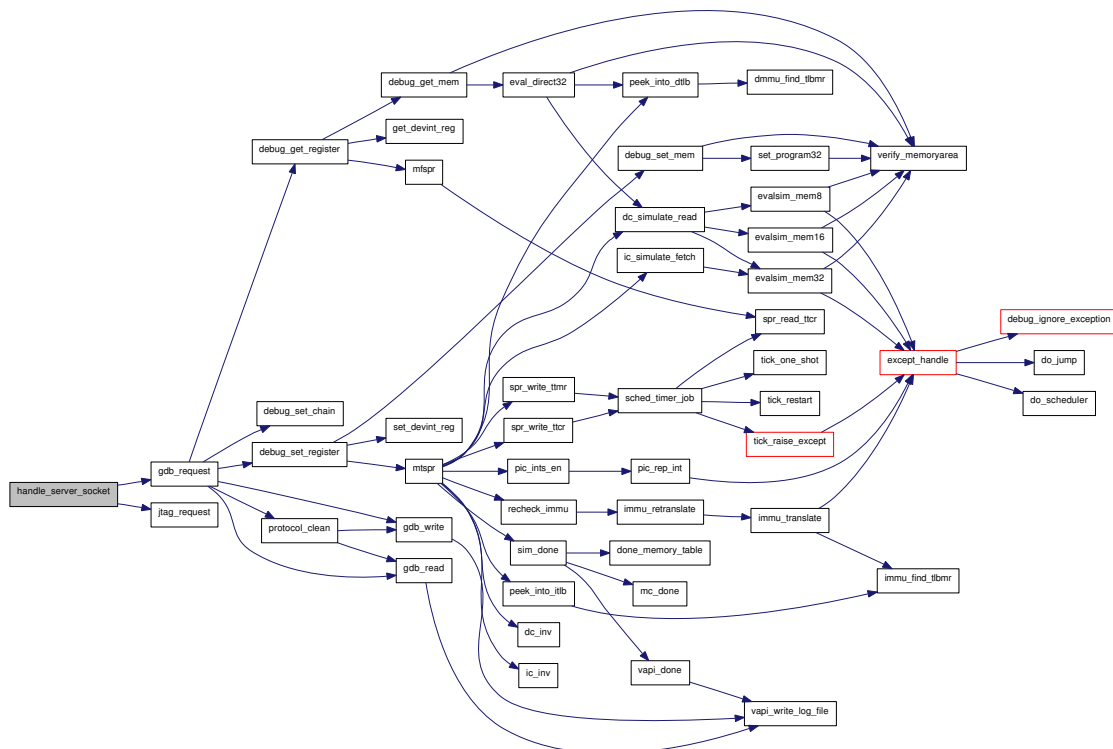
Here is the call graph for this function:



6.81.1.6 static int get_server_socket (const char * name, const char * proto, int port) [static]

6.81.1.7 void handle_server_socket (enum boolean block)

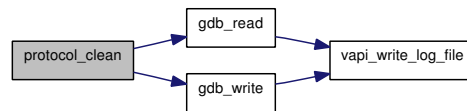
Here is the call graph for this function:



6.81.1.8 `static void jtag_request (void)` [static]

6.81.1.9 `static void protocol_clean (int length, int32_t err)` [static]

Here is the call graph for this function:



6.81.2 Variable Documentation

6.81.2.1 `unsigned int gdb_fd = 0` [static]

6.81.2.2 `unsigned int server_fd = 0` [static]

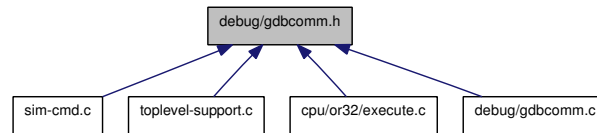
6.81.2.3 `unsigned int server_ip = 0` [static]

6.81.2.4 `unsigned int server_port = 0` [static]

6.81.2.5 `int tcp_level = 0` [static]

6.82 debug/gdbcomm.h File Reference

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



Enumerations

- enum `boolean` { `FALSE` = 0, `TRUE` = 1 }

Functions

- void `handle_server_socket` (enum `boolean`)
- void `block_jtag` ()
- void `gdbcomm_init` ()

6.82.1 Enumeration Type Documentation

6.82.1.1 enum `boolean`

Enumerator:

FALSE

TRUE

6.82.2 Function Documentation

6.82.2.1 void `block_jtag` ()

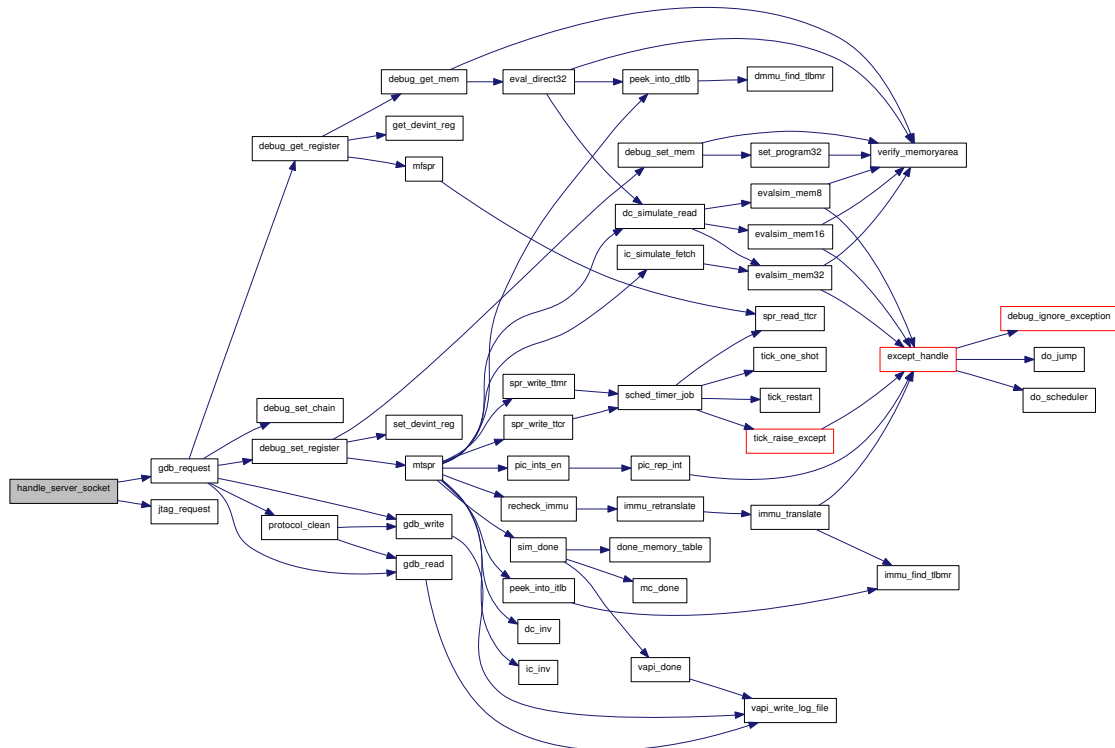
6.82.2.2 void `gdbcomm_init` ()

Here is the call graph for this function:



6.82.2.3 void handle_server_socket (enum boolean)

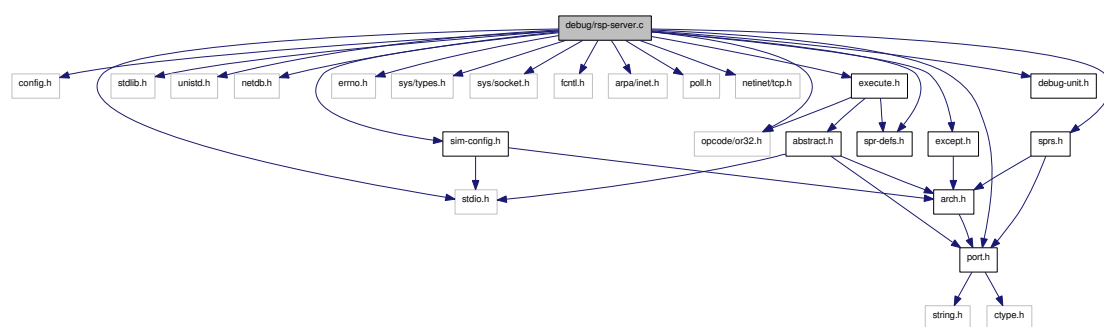
Here is the call graph for this function:



6.83 debug/rsp-server.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <unistd.h>
#include <netdb.h>
#include <stdio.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <poll.h>
#include <netinet/tcp.h>
#include "sim-config.h"
#include "except.h"
#include "opcode/or32.h"
#include "spr-defs.h"
#include "execute.h"
#include "debug-unit.h"
#include "sprs.h"
```

Include dependency graph for rsp-server.c:



Data Structures

- struct [rsp_buf](#)
- struct [mp_entry](#)

Defines

- #define `RSP_TRACE` 1
- #define `OR1KSIM_RSP_SERVICE` "or1ksim-rsp"
- #define `OR1KSIM_RSP_PROTOCOL` "tcp"
- #define `PPC_REGNUM` (MAX_GPRS + 0)
- #define `NPC_REGNUM` (MAX_GPRS + 1)
- #define `SR_REGNUM` (MAX_GPRS + 2)
- #define `NUM_REGS` (MAX_GPRS + 3)
- #define `OR1K_TRAP_INSTR` 0x21000001
- #define `GDB_BUF_MAX` ((NUM_REGS) * 8 + 1)
- #define `MP_HASH_SIZE` 1021

Enumerations

- enum `target_signal` {
`TARGET_SIGNAL_NONE` = 0, `TARGET_SIGNAL_INT` = 2, `TARGET_SIGNAL_ILL` = 4,
`TARGET_SIGNAL_TRAP` = 5,
`TARGET_SIGNAL_FPE` = 8, `TARGET_SIGNAL_BUS` = 10, `TARGET_SIGNAL_SEGV` = 11,
`TARGET_SIGNAL_ALRM` = 14,
`TARGET_SIGNAL_USR2` = 31, `TARGET_SIGNAL_PWR` = 32 }
- enum `mp_type` {
`BP_MEMORY` = 0, `BP_HARDWARE` = 1, `WP_WRITE` = 2, `WP_READ` = 3,
`WP_ACCESS` = 4 }

Functions

- static void `rsp_server_request` ()
- static void `rsp_client_request` ()
- static void `rsp_server_close` ()
- static void `rsp_client_close` ()
- static void `put_packet` (struct `rsp_buf` *buf)
- static void `put_str_packet` (const char *str)
- static struct `rsp_buf` * `get_packet` ()
- static void `put_rsp_char` (char c)
- static int `get_rsp_char` ()
- static int `rsp_unescape` (char *data, int len)
- static void `mp_hash_init` ()
- static void `mp_hash_add` (enum `mp_type` type, unsigned long int addr, unsigned long int instr)
- static struct `mp_entry` * `mp_hash_lookup` (enum `mp_type` type, unsigned long int addr)
- static struct `mp_entry` * `mp_hash_delete` (enum `mp_type` type, unsigned long int addr)
- static int `hex` (int c)
- static void `reg2hex` (unsigned long int val, char *buf)
- static unsigned long int `hex2reg` (char *buf)
- static void `ascii2hex` (char *dest, char *src)
- static void `hex2ascii` (char *dest, char *src)
- static void `set_npc` (unsigned long int addr)
- static void `rsp_report_exception` ()

- static void `rsp_continue` (struct `rsp_buf` *buf)
- static void `rsp_continue_with_signal` (struct `rsp_buf` *buf)
- static void `rsp_continue_generic` (unsigned long int addr, unsigned long int except)
- static void `rsp_read_all_regs` ()
- static void `rsp_write_all_regs` (struct `rsp_buf` *buf)
- static void `rsp_read_mem` (struct `rsp_buf` *buf)
- static void `rsp_write_mem` (struct `rsp_buf` *buf)
- static void `rsp_read_reg` (struct `rsp_buf` *buf)
- static void `rsp_write_reg` (struct `rsp_buf` *buf)
- static void `rsp_query` (struct `rsp_buf` *buf)
- static void `rsp_command` (struct `rsp_buf` *buf)
- static void `rsp_set` (struct `rsp_buf` *buf)
- static void `rsp_restart` ()
- static void `rsp_step` (struct `rsp_buf` *buf)
- static void `rsp_step_with_signal` (struct `rsp_buf` *buf)
- static void `rsp_step_generic` (unsigned long int addr, unsigned long int except)
- static void `rsp_vpkt` (struct `rsp_buf` *buf)
- static void `rsp_write_mem_bin` (struct `rsp_buf` *buf)
- static void `rsp_remove_matchpoint` (struct `rsp_buf` *buf)
- static void `rsp_insert_matchpoint` (struct `rsp_buf` *buf)
- void `rsp_init` ()
- void `handle_rsp` ()
- void `rsp_exception` (unsigned long int except)

Variables

- static const char `hexchars` [] = "0123456789abcdef"
- struct {
 - int `client_waiting`
 - int `proto_num`
 - int `server_fd`
 - int `client_fd`
 - int `sigval`
 - unsigned long int `start_addr`
 - struct `mp_entry` * `mp_hash` [MP_HASH_SIZE]

6.83.1 Define Documentation

6.83.1.1 #define GDB_BUF_MAX ((NUM_REGS) * 8 + 1)

The maximum number of characters in inbound/outbound buffers. The largest packets are the 'G' packet, which must hold the 'G' and all the registers with two hex digits per byte and the 'g' reply, which must hold all the registers, and (in our implementation) an end-of-string (0) character. Adding the EOS allows us to print out the packet as a string. So at least NUMREGBYTES*2 + 1 (for the 'G' or the EOS) are needed for register packets

6.83.1.2 #define MP_HASH_SIZE 1021

Size of the matchpoint hash table. Largest prime < 2¹⁰

6.83.1.3 #define NPC_REGNUM (MAX_GPRS + 1)

Next PC

6.83.1.4 #define NUM_REGS (MAX_GRPS + 3)

Total GDB registers

6.83.1.5 #define OR1K_TRAP_INSTR 0x21000001

Trap instruction for OR32

6.83.1.6 #define OR1KSIM_RSP_PROTOCOL "tcp"

Protocol used by Or1ksim

6.83.1.7 #define OR1KSIM_RSP_SERVICE "or1ksim-rsp"

Name of the Or1ksim RSP service

6.83.1.8 #define PPC_REGNUM (MAX_GPRS + 0)

Previous PC

6.83.1.9 #define RSP_TRACE 1**6.83.1.10 #define SR_REGNUM (MAX_GPRS + 2)**

Supervision Register

6.83.2 Enumeration Type Documentation**6.83.2.1 enum mp_type**

Enumeration of different types of matchpoint. These have explicit values matching the second digit of 'z' and 'Z' packets.

Enumerator:

BP_MEMORY

BP_HARDWARE

WP_WRITE

WP_READ

WP_ACCESS

6.83.2.2 enum target_signal

Definition of GDB target signals. Data taken from the GDB 6.8 source. Only those we use defined here.

Enumerator:

TARGET_SIGNAL_NONE
TARGET_SIGNAL_INT
TARGET_SIGNAL_ILL
TARGET_SIGNAL_TRAP
TARGET_SIGNAL_FPE
TARGET_SIGNAL_BUS
TARGET_SIGNAL_SEGV
TARGET_SIGNAL_ALRM
TARGET_SIGNAL_USR2
TARGET_SIGNAL_PWR

6.83.3 Function Documentation

6.83.3.1 static void ascii2hex (char * dest, char * src) [static]

Convert an ASCII character string to pairs of hex digits

Both source and destination are null terminated.

Parameters:

- *dest* Buffer to store the hex digit pairs (null terminated)
- ← *src* The ASCII string (null terminated)

6.83.3.2 static struct rsp_buf * get_packet () [static, read]

Get a packet from the GDB client

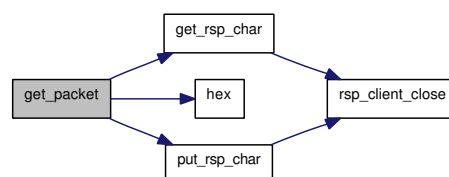
Modeled on the stub version supplied with GDB. The data is in a static buffer. The data should be copied elsewhere if it is to be preserved across a subsequent call to `get_packet()`.

Unlike the reference implementation, we don't deal with sequence numbers. GDB has never used them, and this implementation is only intended for use with GDB 6.8 or later. Sequence numbers were removed from the RSP standard at GDB 5.0.

Returns:

A pointer to the static buffer containing the data

Here is the call graph for this function:



6.83.3.3 `static int get_rsp_char () [static]`

Get a single character from the client socket

This should only be called if the client is open, but we check for safety.

Returns:

The character read, or -1 on failure

Here is the call graph for this function:



6.83.3.4 `void handle_rsp ()`

Look for action on RSP

This function is called when the processor has stalled, which, except for initialization, must be due to an interrupt.

If we have no RSP client, we poll the RSP server for a client requesting to join. We can make no progress until the client is available.

Then if the cause is an interrupt, and the interrupt not been notified to GDB, a packet reporting the cause of the interrupt is sent.

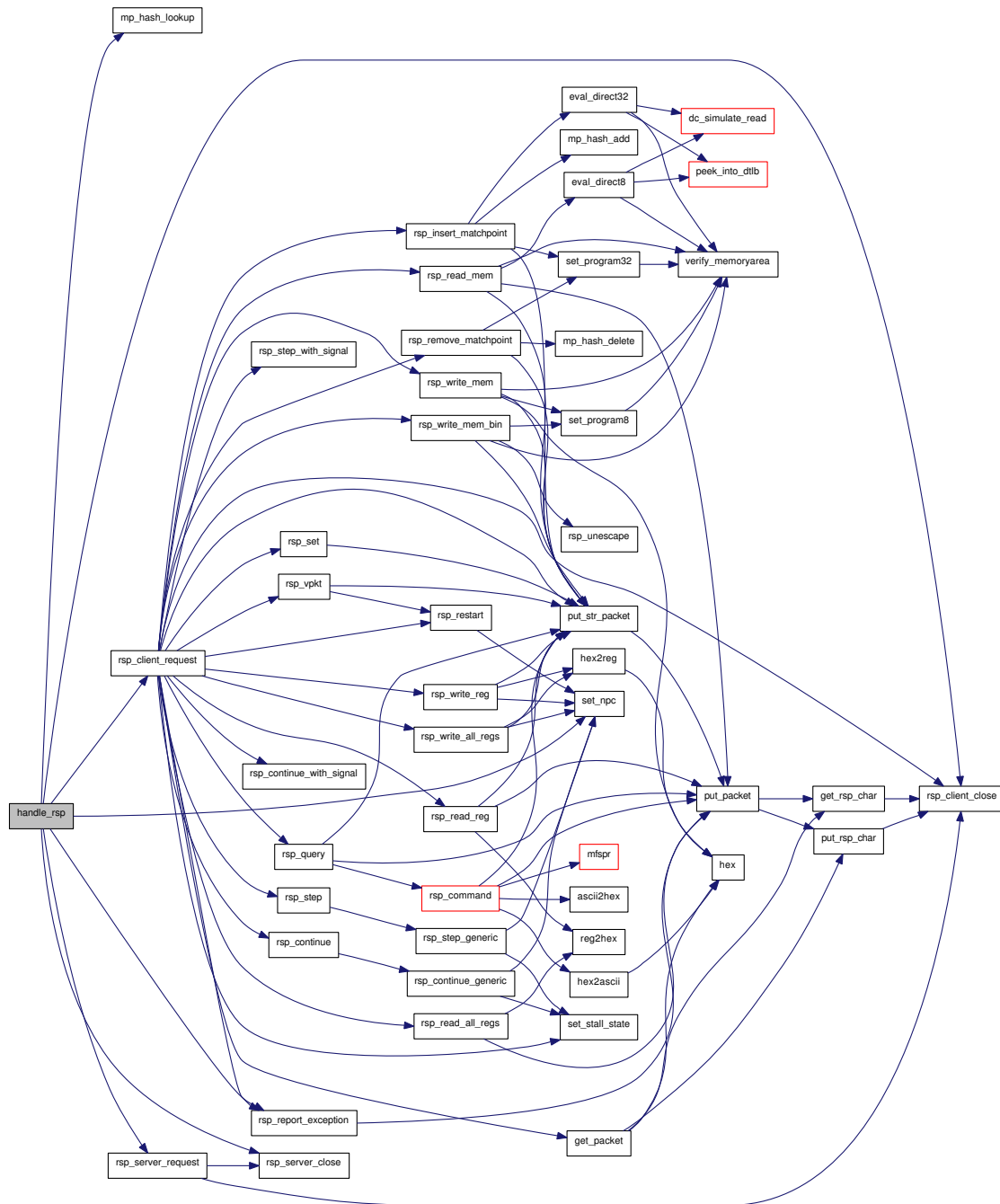
The function then polls the RSP client port (if open) for available input. It then processes the GDB RSP request and return.

If an error occurs when polling the RSP server, other than an interrupt, a warning message is printed out and the RSP server and client (if open) connections are closed.

If an error occurs when polling the RSP client, other than an interrupt, a warning message is printed out and the RSP client connection is closed.

Polling is always blocking (i.e. timeout -1).

Here is the call graph for this function:



6.83.3.5 static int hex (int c) [static]

Utility to give the value of a hex char

Parameters:

← *ch* A character representing a hexadecimal digit. Done as -1, for consistency with other character

routines, which can use -1 as EOF.

Returns:

The value of the hex character, or -1 if the character is invalid.

6.83.3.6 static void hex2ascii (char * *dest*, char * *src*) [static]

Convert pairs of hex digits to an ASCII character string

Both source and destination are null terminated.

Parameters:

→ *dest* The ASCII string (null terminated)

← *src* Buffer holding the hex digit pairs (null terminated)

Here is the call graph for this function:



6.83.3.7 static unsigned long int hex2reg (char * *buf*) [static]

Convert a hex digit string to a register value

The supplied 8 digit hex string is converted to a 32-bit value according the target endianism

Parameters:

← *buf* The buffer with the hex string

Returns:

The value to convert

Here is the call graph for this function:



6.83.3.8 static void mp_hash_add (enum mp_type *type*, unsigned long int *addr*, unsigned long int *instr*) [static]

Add an entry to the matchpoint hash table

Add the entry if it wasn't already there. If it was there do nothing. The match just be on type and addr. The instr need not match, since if this is a duplicate insertion (perhaps due to a lost packet) they will be different.

Parameters:

- ← *type* The type of matchpoint
- ← *addr* The address of the matchpoint [in] instr The instruction to associate with the address

6.83.3.9 `static struct mp_entry * mp_hash_delete (enum mp_type type, unsigned long int addr)`
[static, read]

Delete an entry from the matchpoint hash table

If it is there the entry is deleted from the hash table. If it is not there, no action is taken. The match must be on type AND addr.

The usual fun and games tracking the previous entry, so we can delete things.

Note:

The deletion DOES NOT free the memory associated with the entry, since that is returned. The caller should free the memory when they have used the information.

Parameters:

- ← *type* The type of matchpoint
- ← *addr* The address of the matchpoint

Returns:

The entry deleted, or NULL if the entry was not found

6.83.3.10 `static void mp_hash_init ()` [static]

Initialize the matchpoint hash table

This is an open hash table, so this function clears all the links to NULL.

6.83.3.11 `static struct mp_entry * mp_hash_lookup (enum mp_type type, unsigned long int addr)`
[static, read]

Look up an entry in the matchpoint hash table

The match must be on type AND addr.

Parameters:

- ← *type* The type of matchpoint
- ← *addr* The address of the matchpoint

Returns:

The entry deleted, or NULL if the entry was not found

6.83.3.12 static void put_packet (struct rsp_buf * buf) [static]

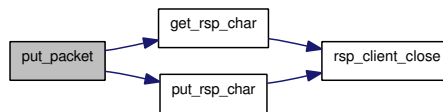
Send a packet to the GDB client

Modeled on the stub version supplied with GDB. Put out the data preceded by a '\$', followed by a '#' and a one byte checksum. '\$', '#', '*' and '}' are escaped by preceding them with '}' and then XORing the character with 0x20.

Parameters:

← *buf* The data to send

Here is the call graph for this function:



6.83.3.13 static void put_rsp_char (char c) [static]

Put a single character out onto the client socket

This should only be called if the client is open, but we check for safety.

Parameters:

← *c* The character to put out

Here is the call graph for this function:

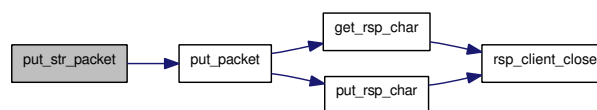


6.83.3.14 static void put_str_packet (const char * str) [static]

Convenience to put a constant string packet

param[in] str The text of the packet

Here is the call graph for this function:



6.83.3.15 `static void reg2hex (unsigned long int val, char * buf)` [static]

Convert a register to a hex digit string

The supplied 32-bit value is converted to an 8 digit hex string according the target endianism. It is null terminated for convenient printing.

Parameters:

← *val* The value to convert

→ *buf* The buffer for the text string

6.83.3.16 `static void rsp_client_close ()` [static]

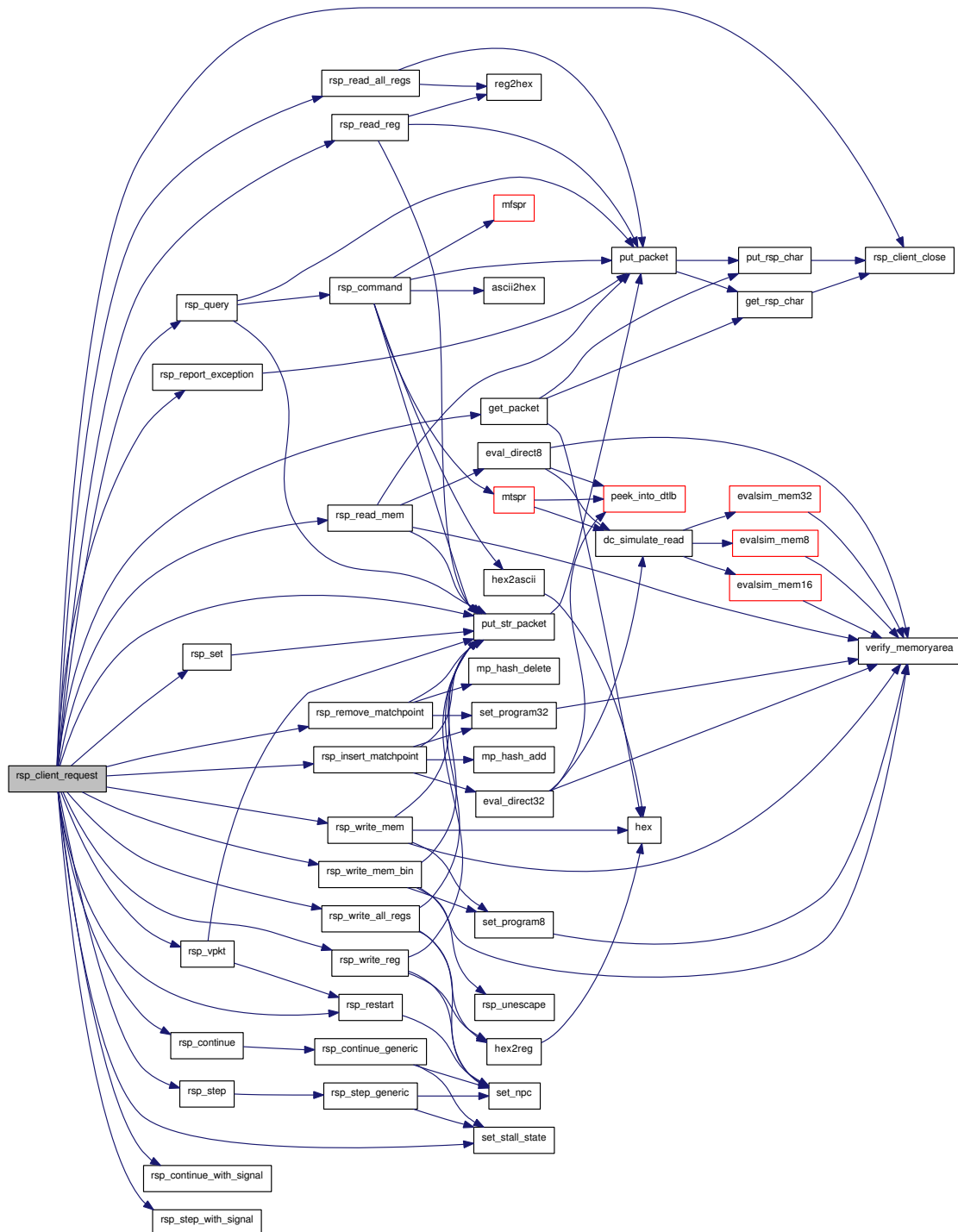
Close the client if it is open

6.83.3.17 `static void rsp_client_request ()` [static]

Deal with a request from the GDB client session

In general, apart from the simplest requests, this function relies on other functions to implement the functionality.

Here is the call graph for this function:



6.83.3.18 static void rsp_command (struct rsp_buf * buf) [static]

Handle a RSP qRcmd request

6.83.3.20 `static void rsp_continue_generic (unsigned long int addr, unsigned long int except)` `[static]`

Generic processing of a continue request

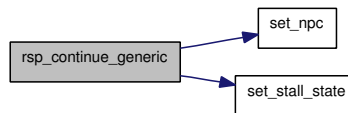
The signal may be EXCEPT_NONE if there is no exception to be handled. Currently the exception is ignored.

The single step flag is cleared in the debug registers and then the processor is unstalled.

Parameters:

- ← *addr* Address from which to step
- ← *except* The exception to use (if any)

Here is the call graph for this function:



6.83.3.21 `static void rsp_continue_with_signal (struct rsp_buf * buf)` `[static]`

Handle a RSP continue with signal request

Currently null. Will use the underlying generic continue function.

Parameters:

- ← *buf* The full continue with signal packet

6.83.3.22 `void rsp_exception (unsigned long int except)`

Note an exception for future processing

The simulator has encountered an exception. Record it here, so that a future call to `handle_exception` will report it back to the client. The signal is supplied in Or1ksim form and recorded in GDB form.

We flag up a warning if an exception is already pending, and ignore the earlier exception.

Parameters:

- ← *except* The exception (Or1ksim form)

6.83.3.23 `void rsp_init ()`

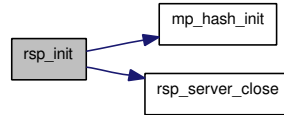
Initialize the Remote Serial Protocol connection

This involves setting up a socket to listen on a socket for attempted connections from a single GDB instance (we couldn't be talking to multiple GDBs at once!).

The service is specified either as a port number in the Or1ksim configuration (parameter `rsp_port` in section `debug`, default 51000) or as a service name in the constant `OR1KSIM_RSP_SERVICE`.

The protocol used for communication is specified in `OR1KSIM_RSP_PROTOCOL`.

Here is the call graph for this function:



6.83.3.24 static void rsp_insert_matchpoint (struct rsp_buf * buf) [static]

Handle a RSP insert breakpoint or matchpoint request

For now only memory breakpoints are implemented, which are implemented by substituting a breakpoint at the specified address. The implementation must cope with the possibility of duplicate packets.

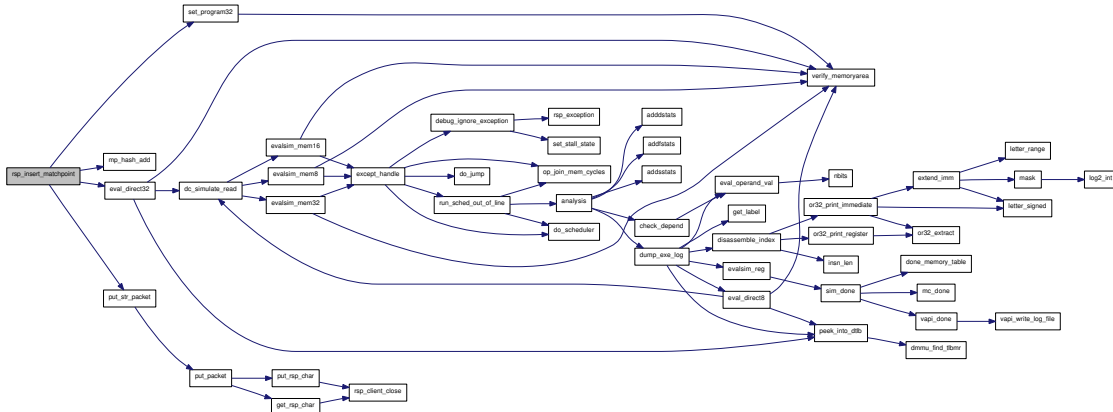
Todo

This doesn't work with icache/immu yet

Parameters:

← *buf* The command received

Here is the call graph for this function:



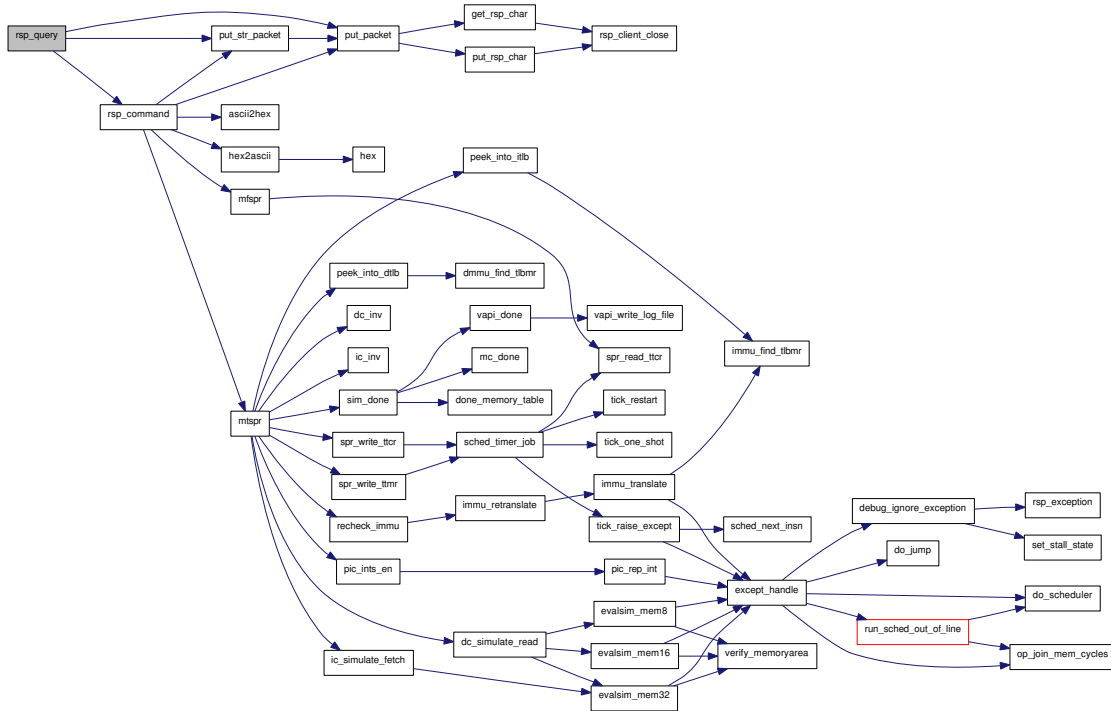
6.83.3.25 static void rsp_query (struct rsp_buf * buf) [static]

Handle a RSP query request

Parameters:

← *buf* The request

Here is the call graph for this function:



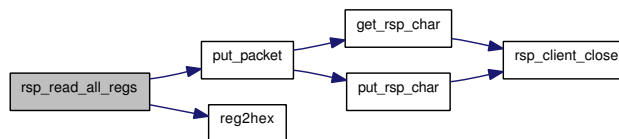
6.83.3.26 static void rsp_read_all_regs () [static]

Handle a RSP read all registers request

The registers follow the GDB sequence for OR1K: GPR0 through GPR31, PPC (i.e. SPR PPC), NPC (i.e. SPR NPC) and SR (i.e. SPR SR). Each register is returned as a sequence of bytes in target endian order.

Each byte is packed as a pair of hex digits.

Here is the call graph for this function:



6.83.3.27 static void rsp_read_mem (struct rsp_buf * buf) [static]

Handle a RSP read memory (symbolic) request

Syntax is:

`m<addr>,<length>`:

The response is the bytes, lowest address first, encoded as pairs of hex digits.

The length given is the number of bytes to be read.

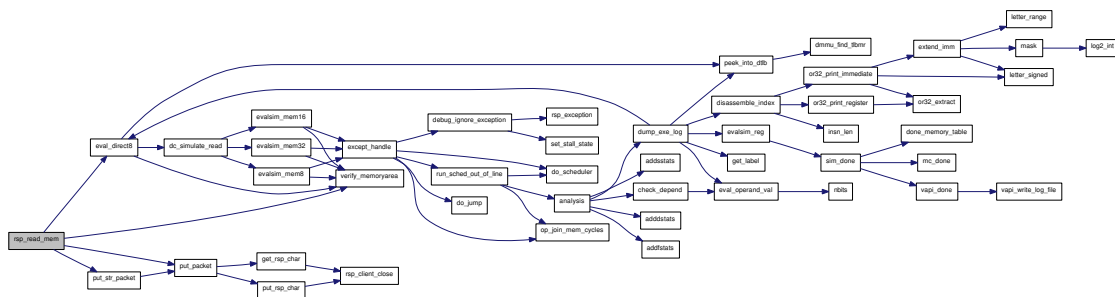
Note:

This function reuses buf, so trashes the original command.

Parameters:

← *buf* The command received

Here is the call graph for this function:



6.83.3.28 static void rsp_read_reg (struct rsp_buf * buf) [static]

Read a single register

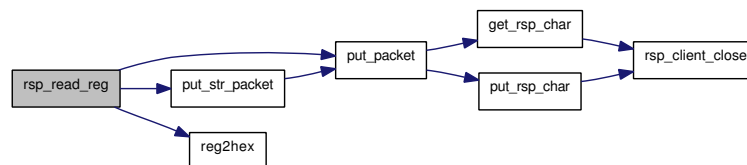
The registers follow the GDB sequence for OR1K: GPR0 through GPR31, PC (i.e. SPR NPC) and SR (i.e. SPR SR). The register is returned as a sequence of bytes in target endian order.

Each byte is packed as a pair of hex digits.

Parameters:

← *buf* The original packet request. Reused for the reply.

Here is the call graph for this function:



6.83.3.29 static void rsp_remove_matchpoint (struct rsp_buf * buf) [static]

Handle a RSP remove breakpoint or matchpoint request

For now only memory breakpoints are implemented, which are implemented by substituting a breakpoint at the specified address. The implementation must cope with the possibility of duplicate packets.

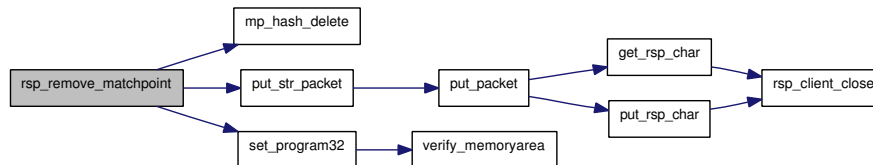
Todo

This doesn't work with icache/immu yet

Parameters:

← *buf* The command received

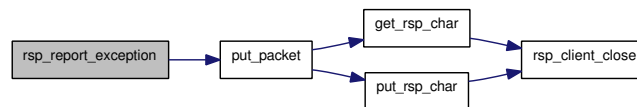
Here is the call graph for this function:

**6.83.3.30 static void rsp_report_exception () [static]**

Send a packet acknowledging an exception has occurred

This is only called if there is a client FD to talk to

Here is the call graph for this function:

**6.83.3.31 static void rsp_restart () [static]**

Handle a RSP restart request

For now we just put the program counter back to the one used with the last vRun request. There is no point in uninstalling the processor, since we'll never get control back.

Here is the call graph for this function:

**6.83.3.32 static void rsp_server_close () [static]**

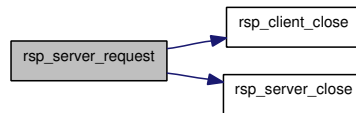
Close the server if it is open

6.83.3.33 static void rsp_server_request () [static]

Handle a request to the server for a new client

We may already have a client. If we do, we will accept an immediately close the new client.

Here is the call graph for this function:



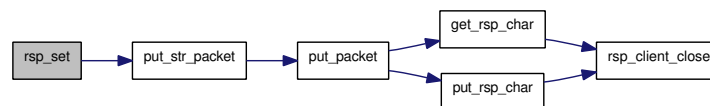
6.83.3.34 static void rsp_set (struct rsp_buf * buf) [static]

Handle a RSP set request

Parameters:

← *buf* The request

Here is the call graph for this function:



6.83.3.35 static void rsp_step (struct rsp_buf * buf) [static]

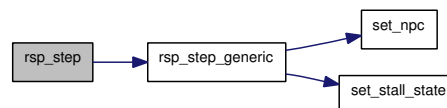
Handle a RSP step request

Parse the command to see if there is an address. Uses the underlying generic step function, with EXCEPT_NONE.

Parameters:

← *buf* The full step packet

Here is the call graph for this function:



6.83.3.36 static void rsp_step_generic (unsigned long int addr, unsigned long int except) [static]

Generic processing of a step request

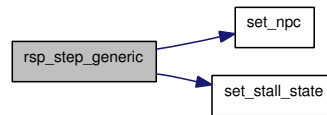
The signal may be EXCEPT_NONE if there is no exception to be handled. Currently the exception is ignored.

The single step flag is set in the debug registers and then the processor is unstalled.

Parameters:

- ← *addr* Address from which to step
- ← *except* The exception to use (if any)

Here is the call graph for this function:



6.83.3.37 static void rsp_step_with_signal (struct rsp_buf * buf) [static]

Handle a RSP step with signal request

Currently null. Will use the underlying generic step function.

Parameters:

- ← *buf* The full step with signal packet

6.83.3.38 static int rsp_unescape (char * data, int len) [static]

"Unescape" RSP binary data

'#', '\$' and '}' are escaped by preceding them by '}' and oring with 0x20.

This function reverses that, modifying the data in place.

Parameters:

- ← *data* The array of bytes to convert [in] len The number of bytes to be converted

Returns:

The number of bytes AFTER conversion

6.83.3.39 static void rsp_vpkt (struct rsp_buf * buf) [static]

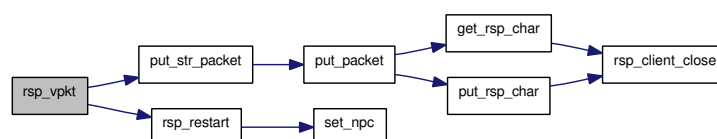
Handle a RSP 'v' packet

These are commands associated with executing the code on the target

Parameters:

- ← *buf* The request

Here is the call graph for this function:



6.83.3.40 static void rsp_write_all_regs (struct rsp_buf * buf) [static]

Handle a RSP write all registers request

The registers follow the GDB sequence for OR1K: GPR0 through GPR31, PPC (i.e. SPR PPC), NPC (i.e. SPR NPC) and SR (i.e. SPR SR). Each register is supplied as a sequence of bytes in target endian order.

Each byte is packed as a pair of hex digits.

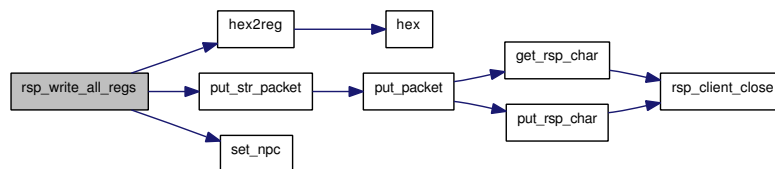
Todo

There is no error checking at present. Non-hex chars will generate a warning message, but there is no other check that the right amount of data is present. The result is always "OK".

Parameters:

← *buf* The original packet request.

Here is the call graph for this function:



6.83.3.41 static void rsp_write_mem (struct rsp_buf * buf) [static]

Handle a RSP write memory (symbolic) request

Syntax is:

m<addr>,<length>:<data>

The data is the bytes, lowest address first, encoded as pairs of hex digits.

The length given is the number of bytes to be written.

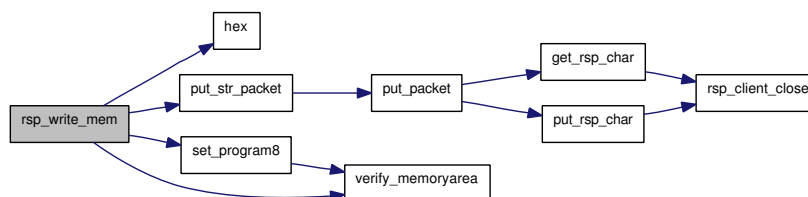
Note:

This function reuses buf, so trashes the original command.

Parameters:

← *buf* The command received

Here is the call graph for this function:



6.83.3.42 static void rsp_write_mem_bin (struct rsp_buf * buf) [static]

Handle a RSP write memory (binary) request

Syntax is:

X<addr>,<length>:

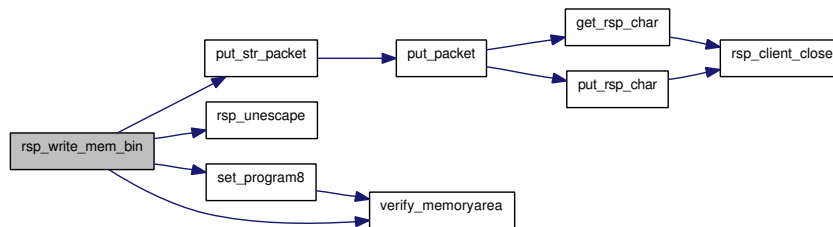
Followed by the specified number of bytes as raw binary. Response should be "OK" if all copied OK, E<nn> if error <nn> has occurred.

The length given is the number of bytes to be written. However the number of data bytes may be greater, since '#', '\$' and '}' are escaped by preceding them by '}' and oring with 0x20.

Parameters:

← *buf* The command received

Here is the call graph for this function:



6.83.3.43 static void rsp_write_reg (struct rsp_buf * buf) [static]

Write a single register

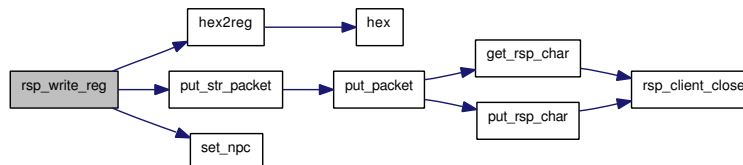
The registers follow the GDB sequence for OR1K: GPR0 through GPR31, PC (i.e. SPR NPC) and SR (i.e. SPR SR). The register is specified as a sequence of bytes in target endian order.

Each byte is packed as a pair of hex digits.

Parameters:

← *buf* The original packet request.

Here is the call graph for this function:



6.83.3.44 static void set_npc (unsigned long int addr) [static]

Set the program counter

This sets the value in the NPC SPR. Not completely trivial, since this is actually cached in `cpu_state.pc`. Any reset of the NPC also involves clearing the delay state and setting the `pcnext` global.

Only actually do this if the requested address is different to the current NPC (avoids clearing the delay pipe).

Parameters:

← *addr* The address to use

6.83.4 Variable Documentation

6.83.4.1 `int client_fd`

FD for talking to GDB

6.83.4.2 `int client_waiting`

Is client waiting a response?

6.83.4.3 `const char hexchars[] = "0123456789abcdef" [static]`

String to map hex digits to chars

6.83.4.4 `struct mp_entry* mp_hash[MP_HASH_SIZE]`

Matchpoint hash table

6.83.4.5 `int proto_num`

Number of the protocol used

6.83.4.6 `struct { ... } rsp [static]`

Central data for the RSP connection

6.83.4.7 `int server_fd`

FD for new connections

6.83.4.8 `int signal`

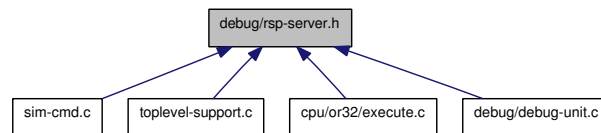
GDB signal for any exception

6.83.4.9 `unsigned long int start_addr`

Start of last run

6.84 debug/rsp-server.h File Reference

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



Functions

- void [rsp_init](#) ()
- void [handle_rsp](#) ()
- void [rsp_exception](#) (unsigned long int except)

6.84.1 Function Documentation

6.84.1.1 void [handle_rsp](#) ()

Look for action on RSP

This function is called when the processor has stalled, which, except for initialization, must be due to an interrupt.

If we have no RSP client, we poll the RSP server for a client requesting to join. We can make no progress until the client is available.

Then if the cause is an interrupt, and the interrupt not been notified to GDB, a packet reporting the cause of the interrupt is sent.

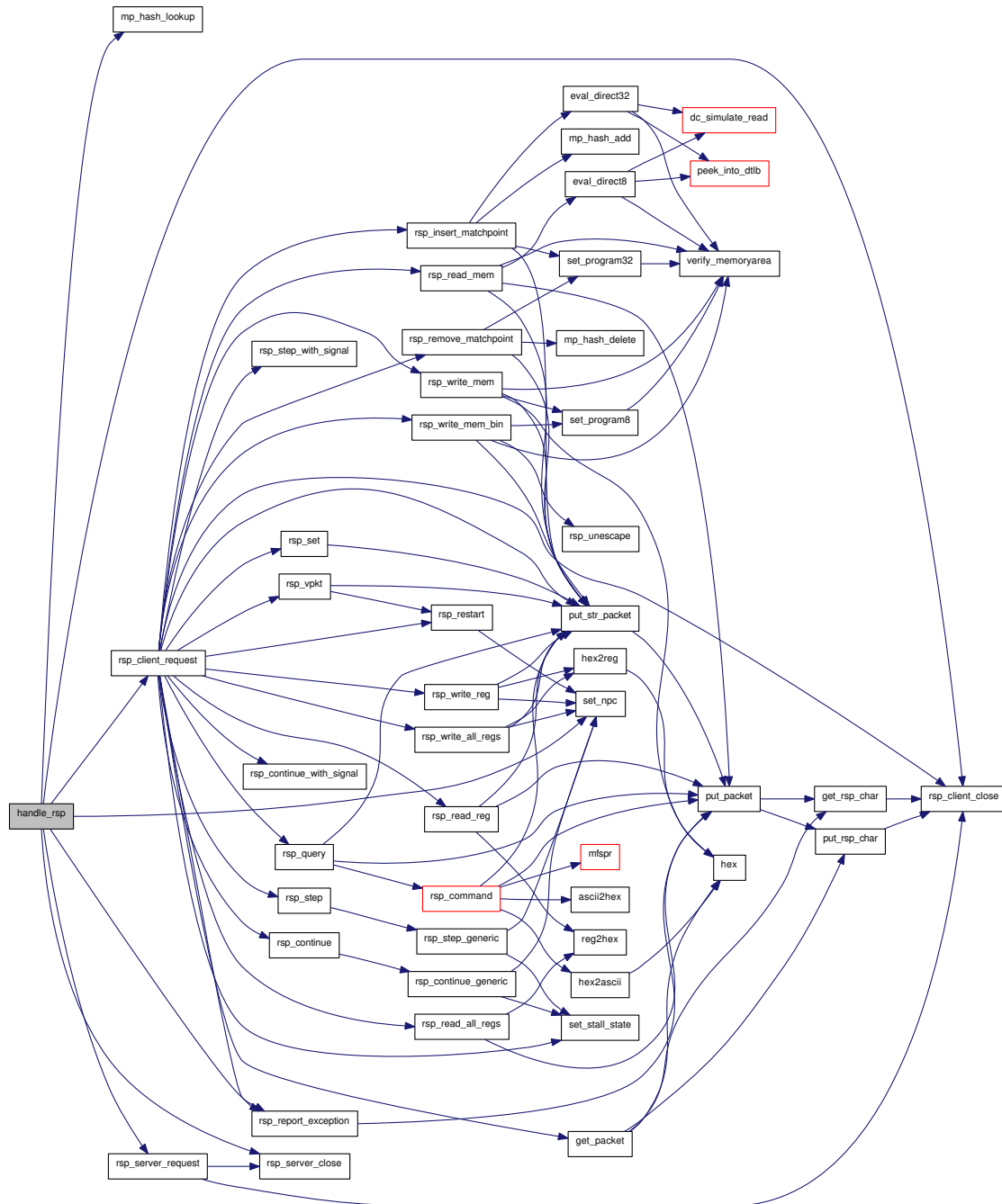
The function then polls the RSP client port (if open) for available input. It then processes the GDB RSP request and return.

If an error occurs when polling the RSP server, other than an interrupt, a warning message is printed out and the RSP server and client (if open) connections are closed.

If an error occurs when polling the RSP client, other than an interrupt, a warning message is printed out and the RSP client connection is closed.

Polling is always blocking (i.e. timeout -1).

Here is the call graph for this function:



6.84.1.2 void rsp_exception (unsigned long int except)

Note an exception for future processing

The simulator has encountered an exception. Record it here, so that a future call to handle_exception will report it back to the client. The signal is supplied in Or1ksim form and recorded in GDB form.

We flag up a warning if an exception is already pending, and ignore the earlier exception.

Parameters:

← *except* The exception (Or1ksim form)

6.84.1.3 void rsp_init ()

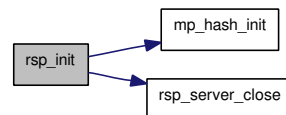
Initialize the Remote Serial Protocol connection

This involves setting up a socket to listen on a socket for attempted connections from a single GDB instance (we couldn't be talking to multiple GDBs at once!).

The service is specified either as a port number in the Or1ksim configuration (parameter `rsp_port` in section `debug`, default 51000) or as a service name in the constant `OR1KSIM_RSP_SERVICE`.

The protocol used for communication is specified in `OR1KSIM_RSP_PROTOCOL`.

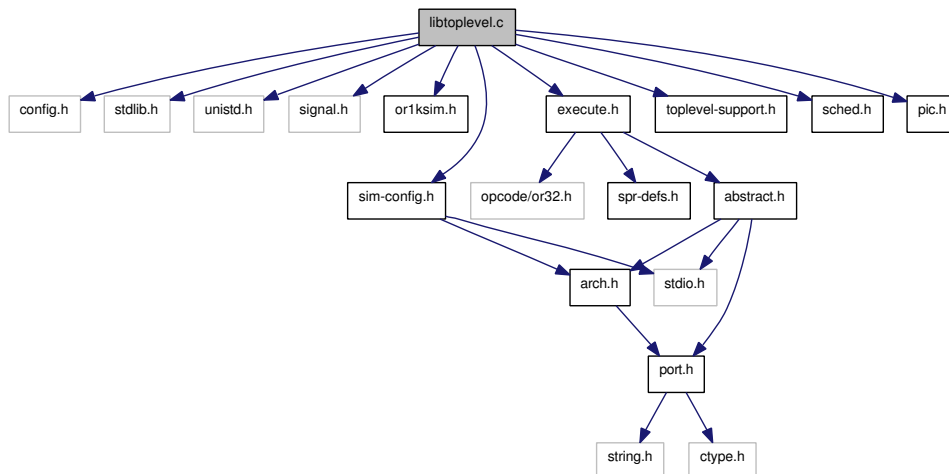
Here is the call graph for this function:



6.85 libtoplevel.c File Reference

```
#include "config.h"
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include "or1ksim.h"
#include "sim-config.h"
#include "toplevel-support.h"
#include "sched.h"
#include "execute.h"
#include "pic.h"
```

Include dependency graph for libtoplevel.c:



Functions

- `int or1ksim_init` (const char *config_file, const char *image_file, void *class_ptr, unsigned long int(*upr)(void *class_ptr, unsigned long int addr, unsigned long int mask), void(*upw)(void *class_ptr, unsigned long int addr, unsigned long int mask, unsigned long int wdata))
- `int or1ksim_run` (double duration)
- `void or1ksim_reset_duration` (double duration)
- `static double internal_or1ksim_time` ()
- `void or1ksim_set_time_point` ()
- `double or1ksim_get_time_period` ()
- `int or1ksim_is_le` ()
- `unsigned long int or1ksim_clock_rate` ()
- `void or1ksim_interrupt` (int i)

6.85.1 Function Documentation

6.85.1.1 static double internal_or1ksim_time () [static]

Return time executed so far

Internal utility to return the time executed so far. Note that this is a re-entrant routine.

Returns:

Time executed so far in seconds

6.85.1.2 unsigned long int or1ksim_clock_rate ()

Return the clock rate

Value is part of the configuration

Returns:

Clock rate in Hz.

6.85.1.3 double or1ksim_get_time_period ()

Return the time since the time point was set

Get the value from the internal parameter

Here is the call graph for this function:



6.85.1.4 int or1ksim_init (const char * *config_file*, const char * *image_file*, void * *class_ptr*, unsigned long int(*) (void * *class_ptr*, unsigned long int *addr*, unsigned long int *mask*) *upr*, void(*) (void * *class_ptr*, unsigned long int *addr*, unsigned long int *mask*, unsigned long int *wdata*) *upw*)

Initialize the simulator.

Allows specification of an (optional) [config](#) file and an image file. Builds up dummy *argc/argv* to pass to the existing argument parser.

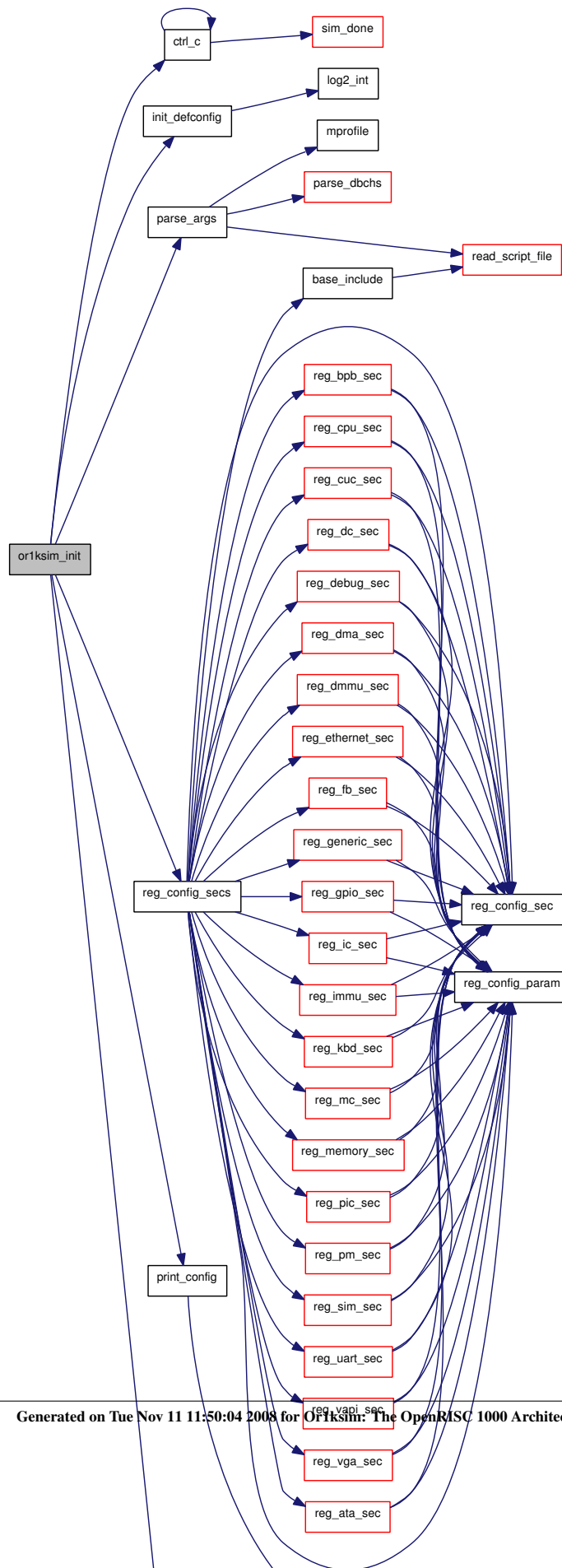
Parameters:

- ← *config_file* Or1ksim configuration file name
- ← *image_file* The program image to execute
- ← *class_ptr* Pointer to a C++ class instance (for use when called by C++)
- ← *upr* Upcall routine for reads
- ← *upw* Upcall routine for writes

Returns:

0 on success and an error code on failure

Here is the call graph for this function:



6.85.1.5 void or1ksim_interrupt (int *i*)

Take an interrupt

Note:

There is no check that the specified interrupt number is reasonable (i.e. ≤ 31).

Parameters:

← *i* The interrupt number

6.85.1.6 int or1ksim_is_le ()

Return the endianism of the model

Note that this is a re-entrant routine.

Returns:

1 if the model is little endian, 0 otherwise.

6.85.1.7 void or1ksim_reset_duration (double *duration*)

Reset the run-time simulation end point

Reset the time for which the simulation should run to the specified duration from NOW (i.e. NOT from when the run started).

Parameters:

← *duration* Time to run for in seconds

6.85.1.8 int or1ksim_run (double *duration*)

Run the simulator

The argument is a time in seconds, which is converted to a number of cycles, if positive. A negative value means "run for ever".

The semantics are that the duration for which the run may occur may be changed mid-run by a call to [or1ksim_reset_duration\(\)](#). This is to allow for the upcalls to generic components adding time, and reducing the time permitted for ISS execution before synchronization of the parent SystemC wrapper.

This is over-ridden if the call was for a negative duration, which means run forever!

Uses a simplified version of the old main program loop. Returns success if the requested number of cycles were run and an error code otherwise.

Parameters:

← *duration* Time to execute for (seconds)

Returns:

OR1KSIM_RC_OK if we run to completion, OR1KSIM_RC_BRKPT if we hit a breakpoint (not clear how this can be set without CLI access)

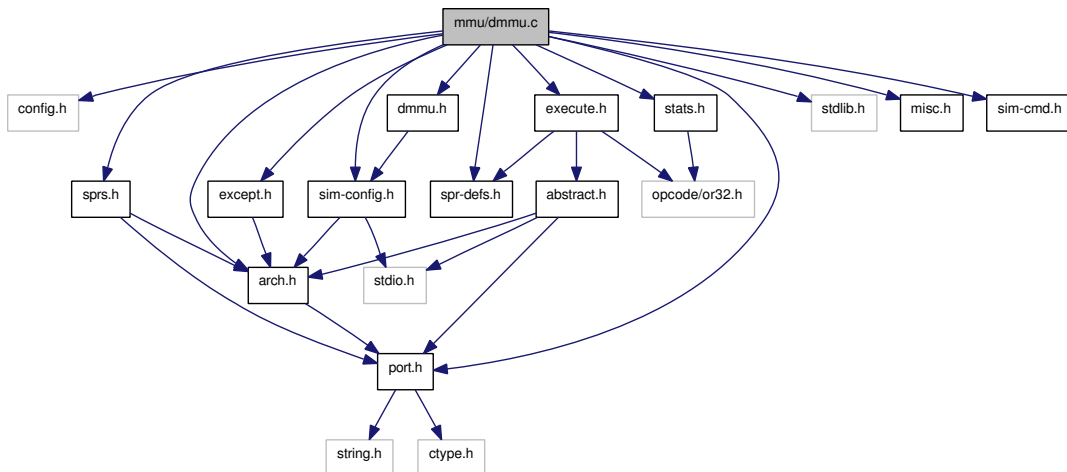
6.86 mainpage File Reference

6.86.1 Detailed Description

6.87 mmu/dmmu.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "dmmu.h"
#include "sim-config.h"
#include "arch.h"
#include "execute.h"
#include "spr-defs.h"
#include "stats.h"
#include "except.h"
#include "sprs.h"
#include "misc.h"
#include "sim-cmd.h"
```

Include dependency graph for dmmu.c:



Functions

- static `uorreg_t * dmmu_find_tlbmr (oraddr_t virtaddr, uorreg_t **dtlbmr_lru, struct dmmu *dmmu)`
- `oraddr_t dmmu_translate (oraddr_t virtaddr, int write_access)`
- `oraddr_t peek_into_dtlb (oraddr_t virtaddr, int write_access, int through_dc)`
- static void `dtlb_status (void *dat)`
- static void `dmmu_enabled (union param_val val, void *dat)`
- static void `dmmu_nsets (union param_val val, void *dat)`
- static void `dmmu_nways (union param_val val, void *dat)`
- static void `dmmu_pagesize (union param_val val, void *dat)`
- static void `dmmu_entsize (union param_val val, void *dat)`
- static void `dmmu_ustates (union param_val val, void *dat)`

- static void [dmmu_missdelay](#) (union [param_val](#) val, void *dat)
- static void [dmmu_hitdelay](#) (union [param_val](#) val, void *dat)
- static void * [dmmu_start_sec](#) ()
- static void [dmmu_end_sec](#) (void *dat)
- void [reg_dmmu_sec](#) (void)

Variables

- struct [dmmu](#) * [dmmu_state](#)

6.87.1 Function Documentation

6.87.1.1 static void [dmmu_enabled](#) (union [param_val](#) val, void * dat) [static]

Enable or disable the DMMU

Set the corresponding field in the UPR

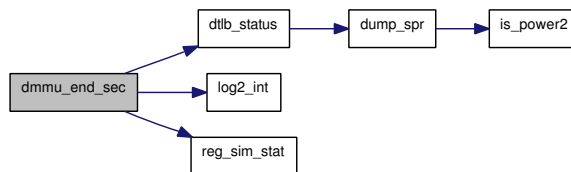
Parameters:

← *val* The value to use

← *dat* The [config](#) data structure

6.87.1.2 static void [dmmu_end_sec](#) (void * dat) [static]

Here is the call graph for this function:



6.87.1.3 static void [dmmu_entsize](#) (union [param_val](#) val, void * dat) [static]

Set the DMMU entry size

Value must be a power of 2. Ignore other values with a warning

Parameters:

← *val* The value to use

← *dat* The [config](#) data structure

Here is the call graph for this function:



6.87.1.4 `static uorreg_t* dmmu_find_tlbmr (oraddr_t virtaddr, uorreg_t** dtlbmr_lru, struct dmmu * dmmu)` [static]

6.87.1.5 `static void dmmu_hitdelay (union param_val val, void * dat)` [static]

6.87.1.6 `static void dmmu_missdelay (union param_val val, void * dat)` [static]

6.87.1.7 `static void dmmu_nsets (union param_val val, void * dat)` [static]

Set the number of DMMU sets

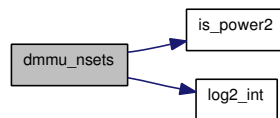
Value must be a power of 2 ≤ 256 . Ignore any other values with a warning. Set the corresponding DMMU configuration flags.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



6.87.1.8 `static void dmmu_nways (union param_val val, void * dat)` [static]

Set the number of DMMU ways

Value must be in the range 1-4. Ignore other values with a warning. Set the corresponding DMMU configuration flags.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.87.1.9 `static void dmmu_pagesize (union param_val val, void * dat)` [static]

Set the DMMU page size

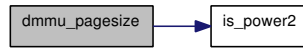
Value must be a power of 2. Ignore other values with a warning

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



6.87.1.10 static void* dmmu_start_sec () [static]

Initialize a new DMMU configuration

ALL parameters are set explicitly to default values. Corresponding SPR flags are set as appropriate.

Returns:

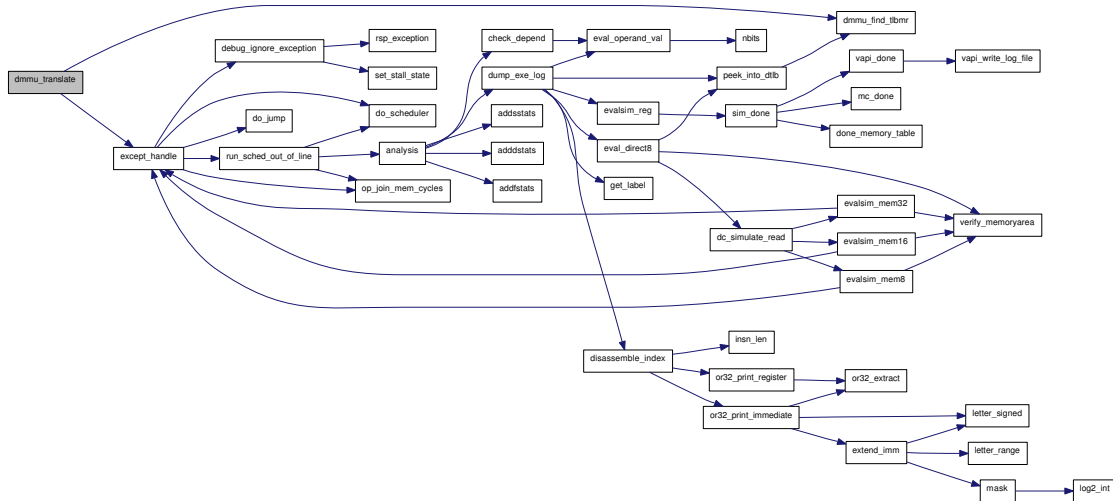
The new memory configuration data structure

Here is the call graph for this function:



6.87.1.11 oraddr_t dmmu_translate (oraddr_t virtaddr, int write_access)

Here is the call graph for this function:



6.87.1.12 static void dmmu_ustates (union param_val val, void * dat) [static]

Set the number of DMMU usage states

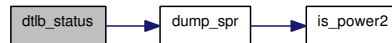
Value must be 2, 3 or 4. Ignore other values with a warning

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure

6.87.1.13 static void dtlb_status (void * dat) [static]

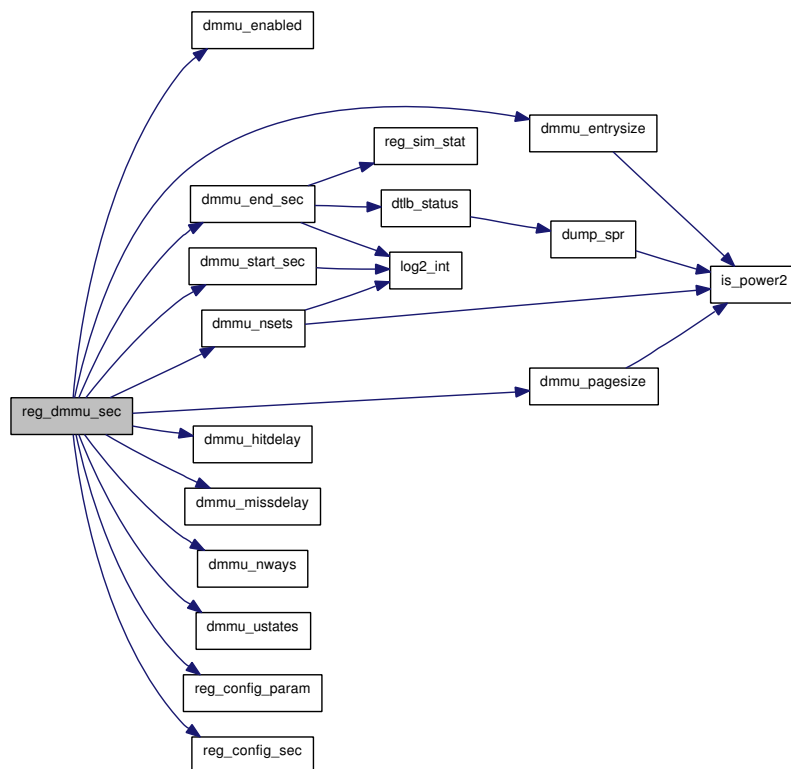
Here is the call graph for this function:

**6.87.1.14 oraddr_t peek_into_dtlb (oraddr_t virtaddr, int write_access, int through_dc)**

Here is the call graph for this function:

**6.87.1.15 void reg_dmmu_sec (void)**

Here is the call graph for this function:



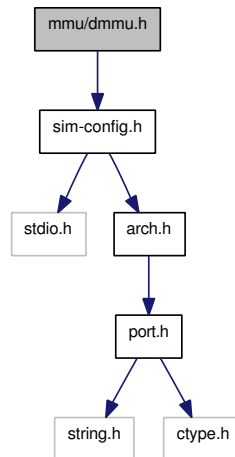
6.87.2 Variable Documentation

6.87.2.1 struct dmmu* dmmu_state

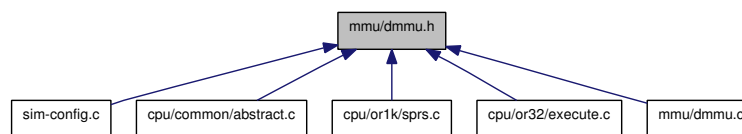
6.88 mmu/dmmu.h File Reference

```
#include "sim-config.h"
```

Include dependency graph for dmmu.h:



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



Data Structures

- struct [dmmu](#)

Defines

- #define [DADDR_PAGE](#)(addr) ((addr) & [dmmu_state](#) → page_mask)

Functions

- [oraddr_t dmmu_translate](#) ([oraddr_t](#) virtaddr, int write_access)
- [oraddr_t dmmu_simulate_tlb](#) ([oraddr_t](#) virtaddr, int write_access)
- [oraddr_t peek_into_dtlb](#) ([oraddr_t](#) virtaddr, int write_access, int through_dc)
- void [reg_dmmu_sec](#) ()

Variables

- struct [dmmu](#) * [dmmu_state](#)

6.88.1 Define Documentation

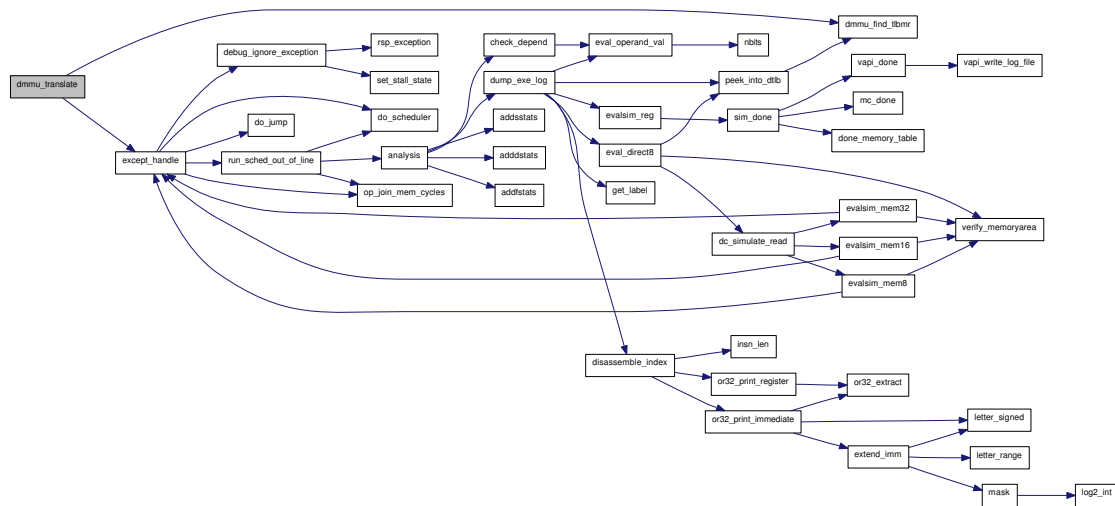
6.88.1.1 `#define DADDR_PAGE(addr) ((addr) & dmmu_state → page_mask)`

6.88.2 Function Documentation

6.88.2.1 `oraddr_t dmmu_simulate_tlb(oraddr_t virtaddr, int write_access)`

6.88.2.2 `oraddr_t dmmu_translate(oraddr_t virtaddr, int write_access)`

Here is the call graph for this function:



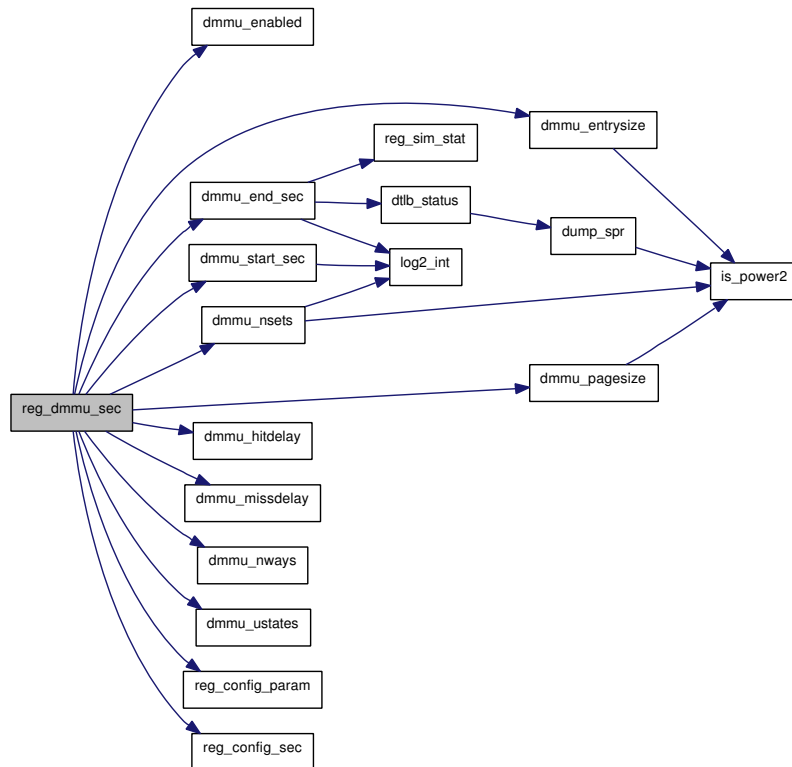
6.88.2.3 `oraddr_t peek_into_dtlb(oraddr_t virtaddr, int write_access, int through_dc)`

Here is the call graph for this function:



6.88.2.4 void reg_dmmu_sec ()

Here is the call graph for this function:



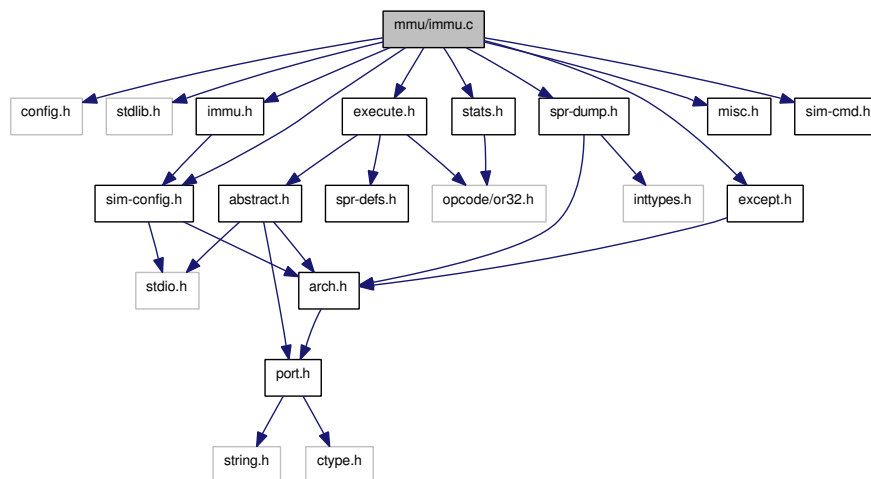
6.88.3 Variable Documentation

6.88.3.1 struct dmmu* dmmu_state

6.89 mmu/immu.c File Reference

```
#include "config.h"
#include <stdlib.h>
#include "immu.h"
#include "sim-config.h"
#include "execute.h"
#include "stats.h"
#include "except.h"
#include "spr-dump.h"
#include "misc.h"
#include "sim-cmd.h"
```

Include dependency graph for immu.c:



Functions

- static `uorreg_t * immu_find_tlbmr (oraddr_t virtaddr, uorreg_t **itlbmr_lru, struct immu *immu)`
- `oraddr_t immu_translate (oraddr_t virtaddr)`
- `oraddr_t peek_into_itlb (oraddr_t virtaddr)`
- static void `itlb_status (void *dat)`
- static void `immu_enabled (union param_val val, void *dat)`
- static void `immu_nsets (union param_val val, void *dat)`
- static void `immu_nways (union param_val val, void *dat)`
- static void `immu_pagesize (union param_val val, void *dat)`
- static void `immu_entsize (union param_val val, void *dat)`
- static void `immu_ustates (union param_val val, void *dat)`
- static void `immu_missdelay (union param_val val, void *dat)`
- static void `immu_hitdelay (union param_val val, void *dat)`
- static void * `immu_start_sec ()`
- static void `immu_end_sec (void *dat)`
- void `reg_immu_sec (void)`

Variables

- struct `immu` * `immu_state`

6.89.1 Function Documentation

6.89.1.1 `static void immu_enabled (union param_val val, void * dat) [static]`

Enable or disable the IMMU

Set the corresponding field in the UPR

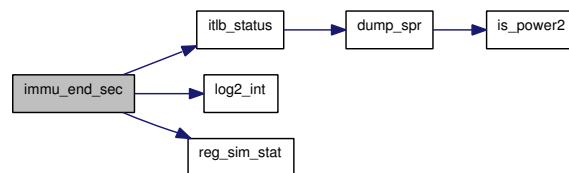
Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.89.1.2 `static void immu_end_sec (void * dat) [static]`

Here is the call graph for this function:



6.89.1.3 `static void immu_entrysize (union param_val val, void * dat) [static]`

Set the IMMU entry size

Value must be a power of 2. Ignore other values with a warning

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



6.89.1.4 `static uorreg_t* immu_find_tlbmr (oraddr_t virtaddr, uorreg_t** tlbmr_lru, struct immu * immu)` [static]

6.89.1.5 `static void immu_hitdelay (union param_val val, void * dat)` [static]

6.89.1.6 `static void immu_missdelay (union param_val val, void * dat)` [static]

6.89.1.7 `static void immu_nsets (union param_val val, void * dat)` [static]

Set the number of DMMU sets

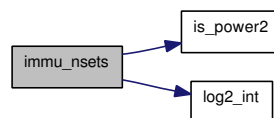
Value must be a power of 2 \leq 256. Ignore any other values with a warning. Set the corresponding IMMU configuration flags.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



6.89.1.8 `static void immu_nways (union param_val val, void * dat)` [static]

Set the number of IMMU ways

Value must be in the range 1-4. Ignore other values with a warning. Set the corresponding IMMU configuration flags.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.89.1.9 `static void immu_pagesize (union param_val val, void * dat)` [static]

Set the IMMU page size

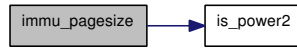
Value must be a power of 2. Ignore other values with a warning

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



6.89.1.10 static void* immu_start_sec () [static]

Initialize a new DMMU configuration

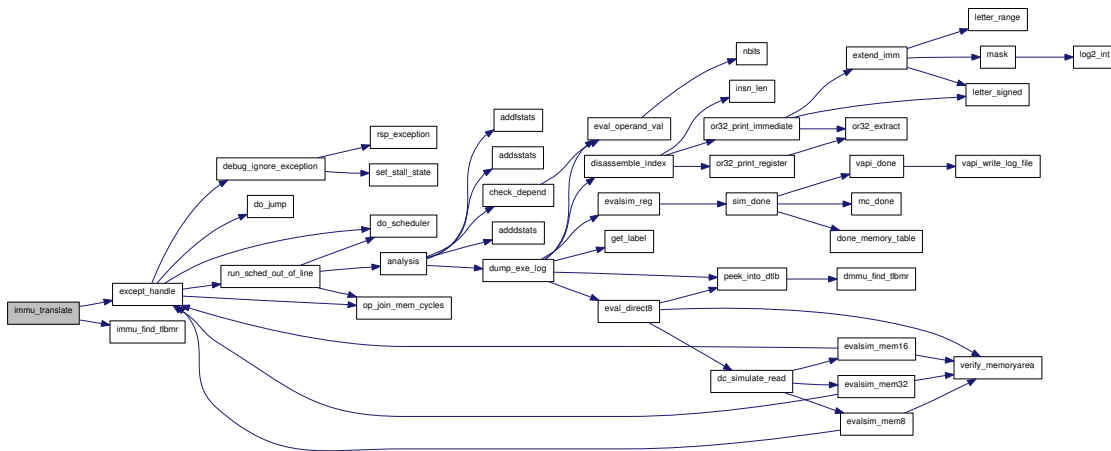
ALL parameters are set explicitly to default values.

Here is the call graph for this function:



6.89.1.11 oraddr_t immu_translate (oraddr_t virtaddr)

Here is the call graph for this function:



6.89.1.12 static void immu_ustates (union param_val val, void * dat) [static]

Set the number of IMMU usage states

Value must be 2, 3 or 4. Ignore other values with a warning

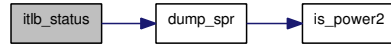
Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.89.1.13 static void itlb_status (void * *dat*) [static]

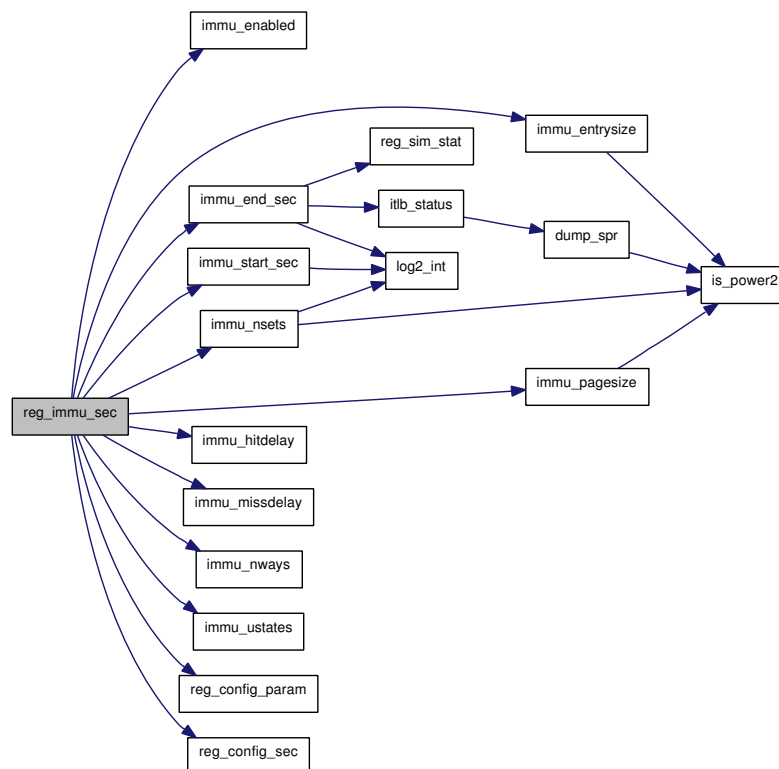
Here is the call graph for this function:

**6.89.1.14** oraddr_t peek_into_itlb (oraddr_t *virtaddr*)

Here is the call graph for this function:

**6.89.1.15** void reg_immu_sec (void)

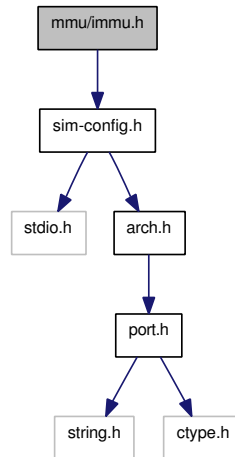
Here is the call graph for this function:

**6.89.2** Variable Documentation**6.89.2.1** struct immu* immu_state

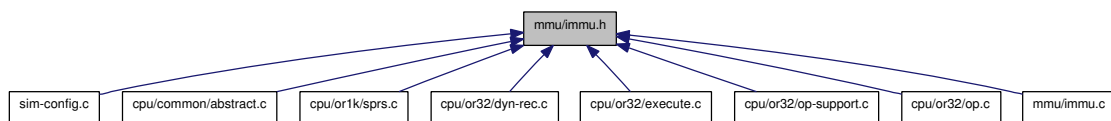
6.90 mmu/immu.h File Reference

```
#include "sim-config.h"
```

Include dependency graph for immu.h:



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



Data Structures

- struct [immu](#)

Defines

- #define [IADDR_PAGE](#)(addr) ((addr) & [immu_state](#) → page_mask)

Functions

- [oraddr_t immu_translate](#) ([oraddr_t](#) virtaddr)
- [oraddr_t immu_simulate_tlb](#) ([oraddr_t](#) virtaddr)
- [oraddr_t peek_into_itlb](#) ([oraddr_t](#) virtaddr)
- void [reg_immu_sec](#) ()

Variables

- struct [immu](#) * [immu_state](#)

6.90.1 Define Documentation

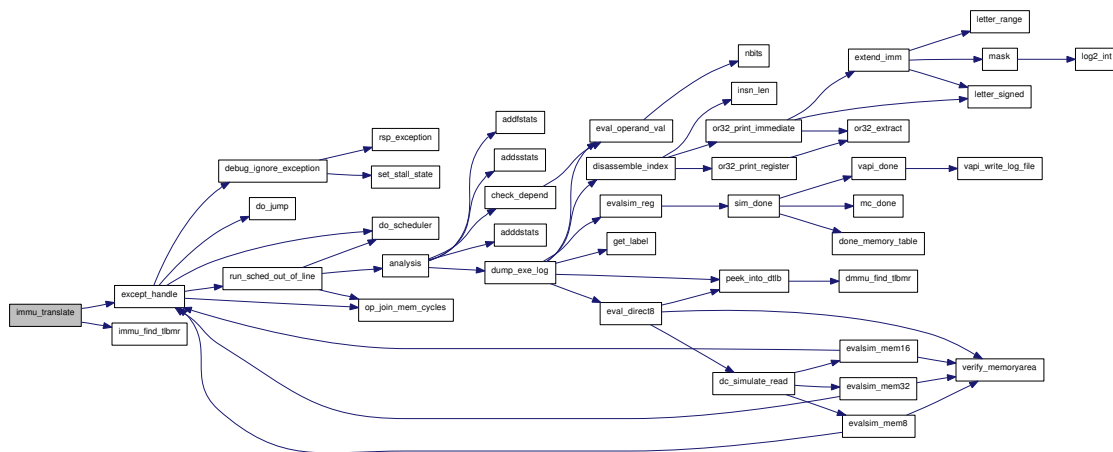
6.90.1.1 `#define IADDR_PAGE(addr) ((addr) & immu_state → page_mask)`

6.90.2 Function Documentation

6.90.2.1 `oraddr_t immu_simulate_tlb (oraddr_t virtaddr)`

6.90.2.2 `oraddr_t immu_translate (oraddr_t virtaddr)`

Here is the call graph for this function:



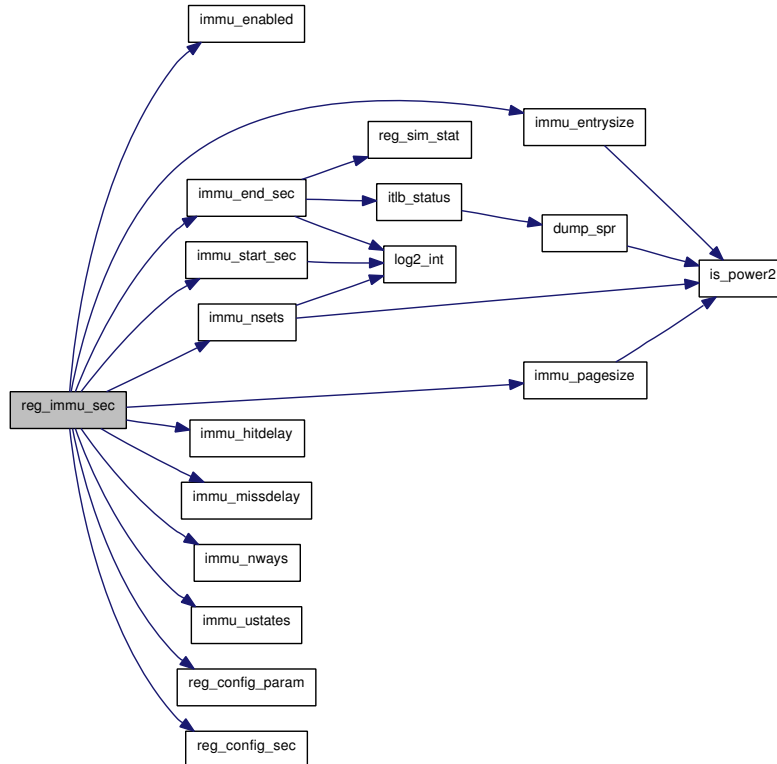
6.90.2.3 `oraddr_t peek_into_itlb (oraddr_t virtaddr)`

Here is the call graph for this function:



6.90.2.4 void reg_immu_sec ()

Here is the call graph for this function:



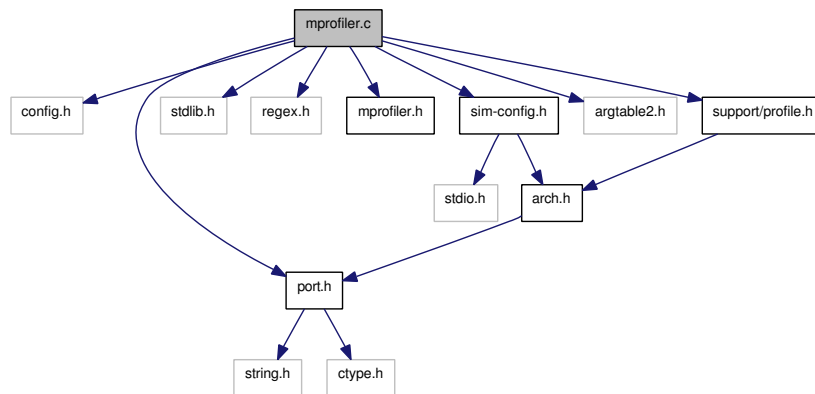
6.90.3 Variable Documentation

6.90.3.1 struct immu* immu_state

6.91 mprofiler.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <regex.h>
#include "mprofiler.h"
#include "sim-config.h"
#include "argtable2.h"
#include "support/profile.h"
```

Include dependency graph for mprofiler.c:



Data Structures

- struct [memory_hash](#)

Defines

- #define [MODE_DETAIL](#) 0
- #define [MODE_PRETTY](#) 1
- #define [MODE_ACCESS](#) 2
- #define [MODE_WIDTH](#) 3
- #define [BUF_SIZE](#) 256
- #define [HASH_SIZE](#) 0x10000
- #define [HASH_FUNC\(x\)](#) ((x) & 0xffff)

Functions

- static void [hash_add](#) ([oraddr_t](#) addr, int index)
- static unsigned long [hash_get](#) ([oraddr_t](#) addr, int index)
- static void [init](#) ()
- static void [read_file](#) (FILE *f, int mode)

- static int `nbits` (unsigned long a)
- static void `printout` (int mode)
- int `main_mprofiler` (int argc, char *argv[], int just_help)

Variables

- static struct `memory_hash` * `hash` [HASH_SIZE]
- static int `group_bits` = 2
- static `oraddr_t` `start_addr` = 0
- static `oraddr_t` `end_addr` = 0xffffffff
- static FILE * `fprof` = 0

6.91.1 Define Documentation

6.91.1.1 `#define BUF_SIZE 256`

6.91.1.2 `#define HASH_FUNC(x) ((x) & 0xffff)`

6.91.1.3 `#define HASH_SIZE 0x10000`

6.91.1.4 `#define MODE_ACCESS 2`

6.91.1.5 `#define MODE_DETAIL 0`

6.91.1.6 `#define MODE_PRETTY 1`

6.91.1.7 `#define MODE_WIDTH 3`

6.91.2 Function Documentation

6.91.2.1 `static void hash_add (oraddr_t addr, int index)` [static]

6.91.2.2 `static unsigned long hash_get (oraddr_t addr, int index)` [static]

6.91.2.3 `static void init ()` [static]

6.91.2.4 `int main_mprofiler (int argc, char * argv[], int just_help)`

Parse the arguments for the profiling utility

Updated by Jeremy Bennett to use argtable2. Also has an option just to print help, for use with the CLI.

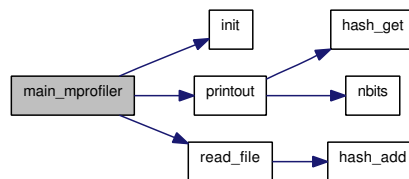
Parameters:

- ← *argc* Number of command args
- ← *argv* Vector of the command args
- ← *just_help* If 1 (true), ignore argc & argv and just print out the help message without parsing args

Returns:

0 on success, 1 on failure

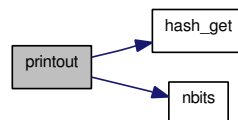
Here is the call graph for this function:



6.91.2.5 `static int nbits (unsigned long a)` [static]

6.91.2.6 `static void printout (int mode)` [static]

Here is the call graph for this function:



6.91.2.7 `static void read_file (FILE *f, int mode)` [static]

Here is the call graph for this function:



6.91.3 Variable Documentation

6.91.3.1 `oraddr_t end_addr = 0xffffffff` [static]

End address

6.91.3.2 `FILE* fprof = 0` [static]

6.91.3.3 `int group_bits = 2` [static]

Groups size – how much addresses should be joined together

6.91.3.4 `struct memory_hash * hash[HASH_SIZE]` [static]

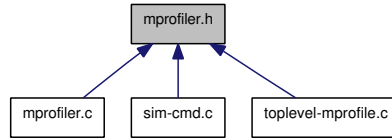
Hash table data structure

6.91.3.5 oraddr_t start_addr = 0 [static]

Start address

6.92 mprofiler.h File Reference

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



Functions

- int [main_mprofiler](#) (int argc, char *argv[], int just_help)

6.92.1 Function Documentation

6.92.1.1 int main_mprofiler (int argc, char * argv[], int just_help)

Parse the arguments for the profiling utility

Updated by Jeremy Bennett to use argtable2. Also has an option just to print help, for use with the CLI.

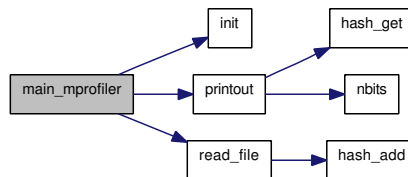
Parameters:

- ← *argc* Number of command args
- ← *argv* Vector of the command args
- ← *just_help* If 1 (true), ignore argc & argv and just print out the help message without parsing args

Returns:

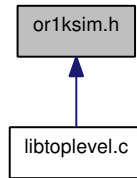
- 0 on success, 1 on failure

Here is the call graph for this function:



6.93 or1ksim.h File Reference

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



Enumerations

- enum `or1ksim_rc` { `ORIKSIM_RC_OK`, `ORIKSIM_RC_BADINIT`, `ORIKSIM_RC_BRKPT` }

Functions

- int `or1ksim_init` (const char *config_file, const char *image_file, void *class_ptr, unsigned long int(*upr)(void *class_ptr, unsigned long intaddr, unsigned long intmask), void(*upw)(void *class_ptr, unsigned long intaddr, unsigned long intmask, unsigned long intwdata))
- int `or1ksim_run` (double duration)
- void `or1ksim_reset_duration` (double duration)
- void `or1ksim_set_time_point` ()
- double `or1ksim_get_time_period` ()
- int `or1ksim_is_le` ()
- unsigned long int `or1ksim_clock_rate` ()
- void `or1ksim_interrupt` (int i)

6.93.1 Enumeration Type Documentation

6.93.1.1 enum or1ksim_rc

Enumerator:

ORIKSIM_RC_OK
ORIKSIM_RC_BADINIT
ORIKSIM_RC_BRKPT

6.93.2 Function Documentation

6.93.2.1 unsigned long int or1ksim_clock_rate ()

Return the clock rate

Value is part of the configuration

Returns:

Clock rate in Hz.

6.93.2.2 double or1ksim_get_time_period ()

Return the time since the time point was set

Get the value from the internal parameter

Here is the call graph for this function:



6.93.2.3 int or1ksim_init (const char * *config_file*, const char * *image_file*, void * *class_ptr*, unsigned long int(*) (void * *class_ptr*, unsigned long intaddr, unsigned long intmask) *upr*, void(*) (void * *class_ptr*, unsigned long intaddr, unsigned long intmask, unsigned long intwdata) *upw*)

6.93.2.4 void or1ksim_interrupt (int *i*)

Take an interrupt

Note:

There is no check that the specified interrupt number is reasonable (i.e. ≤ 31).

Parameters:

← *i* The interrupt number

6.93.2.5 int or1ksim_is_le ()

Return the endianism of the model

Note that this is a re-entrant routine.

Returns:

1 if the model is little endian, 0 otherwise.

6.93.2.6 void or1ksim_reset_duration (double *duration*)

Reset the run-time simulation end point

Reset the time for which the simulation should run to the specified duration from NOW (i.e. NOT from when the run started).

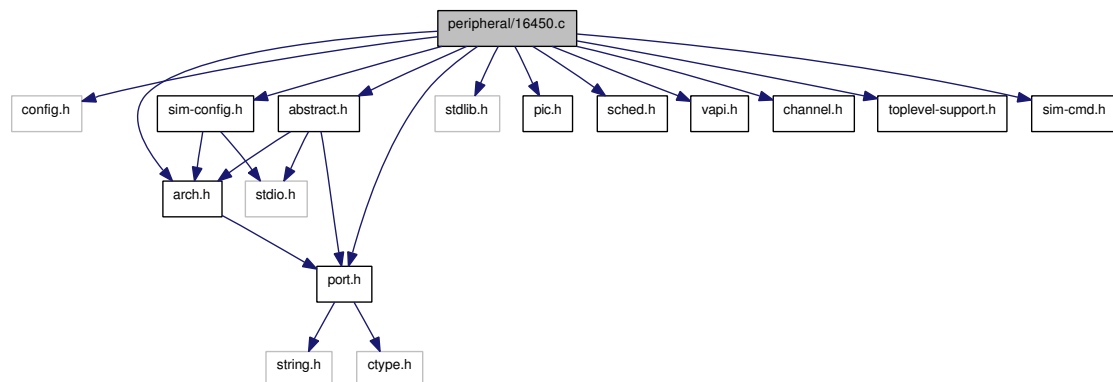
Parameters:

← *duration* Time to run for in seconds

6.94 peripheral/16450.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "sim-config.h"
#include "arch.h"
#include "pic.h"
#include "sched.h"
#include "vapi.h"
#include "channel.h"
#include "abstract.h"
#include "toplevel-support.h"
#include "sim-cmd.h"
```

Include dependency graph for 16450.c:



Data Structures

- struct [dev_16450](#)

Defines

- #define [MIN\(a, b\)](#) ((a) < (b) ? (a) : (b))
- #define [UART_ADDR_SPACE](#) 8
- #define [UART_MAX_FIFO_LEN](#) 16
- #define [MAX_SKEW](#) 1
- #define [UART_VAPI_BUF_LEN](#) 128
- #define [UART_CLOCK_DIVIDER](#) 16
- #define [UART_FGETC_SLOWDOWN](#) 100
- #define [UART_RXBUF](#) 0
- #define [UART_TXBUF](#) 0

- #define `UART_DLL` 0
- #define `UART_DLH` 1
- #define `UART_IER` 1
- #define `UART_IIR` 2
- #define `UART_FCR` 2
- #define `UART_LCR` 3
- #define `UART_MCR` 4
- #define `UART_LSR` 5
- #define `UART_MSR` 6
- #define `UART_SCR` 7
- #define `UART_VALID_LCR` 0xff
- #define `UART_VALID_LSR` 0xff
- #define `UART_VALID_IIR` 0x0f
- #define `UART_VALID_FCR` 0xc0
- #define `UART_VALID_IER` 0x0f
- #define `UART_VALID_MCR` 0x1f
- #define `UART_VALID_MSR` 0xff
- #define `UART_LCR_DLAB` 0x80
- #define `UART_LCR_SBC` 0x40
- #define `UART_LCR_SPAR` 0x20
- #define `UART_LCR_EPAR` 0x10
- #define `UART_LCR_PARITY` 0x08
- #define `UART_LCR_STOP` 0x04
- #define `UART_LCR_WLEN5` 0x00
- #define `UART_LCR_WLEN6` 0x01
- #define `UART_LCR_WLEN7` 0x02
- #define `UART_LCR_WLEN8` 0x03
- #define `UART_LCR_RESET` 0x03
- #define `UART_LSR_RXERR` 0x80
- #define `UART_LSR_TXSRE` 0x40
- #define `UART_LSR_TXBUFE` 0x20
- #define `UART_LSR_BREAK` 0x10
- #define `UART_LSR_FRAME` 0x08
- #define `UART_LSR_PARITY` 0x04
- #define `UART_LSR_OVRRUN` 0x02
- #define `UART_LSR_RDRDY` 0x01
- #define `UART_IIR_NO_INT` 0x01
- #define `UART_IIR_ID` 0x06
- #define `UART_IIR_MSI` 0x00
- #define `UART_IIR_THRI` 0x02
- #define `UART_IIR_RDI` 0x04
- #define `UART_IIR_RLSI` 0x06
- #define `UART_IIR_CTI` 0x0c
- #define `UART_FCR_FIE` 0x01
- #define `UART_FCR_RRXFI` 0x02
- #define `UART_FCR_RTXFI` 0x04
- #define `UART_FIFO_TRIGGER`(x)
- #define `UART_IER_MSI` 0x08
- #define `UART_IER_RLSI` 0x04
- #define `UART_IER_THRI` 0x02

- #define `UART_IER_RDI` 0x01
- #define `UART_MCR_LOOP` 0x10
- #define `UART_MCR_AUX2` 0x08
- #define `UART_MCR_AUX1` 0x04
- #define `UART_MCR_RTS` 0x02
- #define `UART_MCR_DTR` 0x01
- #define `UART_MSR_DCD` 0x80
- #define `UART_MSR_RI` 0x40
- #define `UART_MSR_DSR` 0x20
- #define `UART_MSR_CTS` 0x10
- #define `UART_MSR_DDCD` 0x08
- #define `UART_MSR_TERI` 0x04
- #define `UART_MSR_DDSR` 0x02
- #define `UART_MSR_DCTS` 0x01
- #define `UART_BREAK_COUNT` 1
- #define `UART_CHAR_TIMEOUT` 4

Functions

- static void `uart_recv_break` (void *dat)
- static void `uart_recv_char` (void *dat)
- static void `uart_check_char` (void *dat)
- static void `uart_sched_recv_check` (struct `dev_16450` *uart)
- static void `uart_vapi_cmd` (void *dat)
- static void `uart_clear_int` (struct `dev_16450` *uart, int intr)
- static void `uart_tx_send` (void *dat)
- static unsigned long `char_clks` (int dll, int dlh, int lcr)
- static void `uart_int_msi` (void *dat)
- static void `uart_int_thri` (void *dat)
- static void `uart_int_cti` (void *dat)
- static void `uart_int_rdi` (void *dat)
- static void `uart_int_rlsi` (void *dat)
- static void `uart_check_rlsi` (void *dat)
- static void `uart_check_rdi` (void *dat)
- static void `uart_next_int` (struct `dev_16450` *uart)
- static void `uart_loopback` (struct `dev_16450` *uart)
- static void `send_char` (struct `dev_16450` *uart, int bits_send)
- void `uart_char_clock` (void *dat)
- void `uart_send_break` (void *dat)
- static void `uart_add_char` (struct `dev_16450` *uart, int ch)
- static void `uart_recv_break_start` (void *dat)
- static void `uart_recv_break_stop` (void *dat)
- static void `uart_write_byte` (`oraddr_t` addr, `uint8_t` value, void *dat)
- static `uint8_t` `uart_read_byte` (`oraddr_t` addr, void *dat)
- static void `uart_vapi_read` (unsigned long id, unsigned long data, void *dat)
- void `uart_reset` (void *dat)
- void `uart_status` (void *dat)
- static void `uart_baseaddr` (union `param_val` val, void *dat)
- static void `uart_jitter` (union `param_val` val, void *dat)
- static void `uart_irq` (union `param_val` val, void *dat)

- static void `uart_16550` (union `param_val` val, void *dat)
- static void `uart_channel` (union `param_val` val, void *dat)
- static void `uart_newway` (union `param_val` val, void *dat)
- static void `uart_vapi_id` (union `param_val` val, void *dat)
- static void `uart_enabled` (union `param_val` val, void *dat)
- static void * `uart_sec_start` ()
- static void `uart_sec_end` (void *dat)
- void `reg_uart_sec` (void)

6.94.1 Define Documentation

6.94.1.1 #define MAX_SKEW 1

max. clock skew in subclocks

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

6.94.1.3 #define UART_ADDR_SPACE 8

UART addr space size in bytes

6.94.1.4 #define UART_BREAK_COUNT 1

6.94.1.5 #define UART_CHAR_TIMEOUT 4

6.94.1.6 #define UART_CLOCK_DIVIDER 16

UART clock divider

6.94.1.7 #define UART_DLH 1

6.94.1.8 #define UART_DLL 0

6.94.1.9 #define UART_FCR 2

6.94.1.10 #define UART_FCR_FIE 0x01

6.94.1.11 #define UART_FCR_RRXFI 0x02

6.94.1.12 #define UART_FCR_RTXFI 0x04

6.94.1.13 #define UART_FGETC_SLOWDOWN 100

fgetc() slowdown factor

6.94.1.14 #define UART_FIFO_TRIGGER(x)

Value:

```
/* Trigger values for indexes 0..3 */\n((x) == 0 ? 1\  
:(x) == 1 ? 4\  
:(x) == 2 ? 8\  
:(x) == 3 ? 14 : 0)
```


6.94.1.15 **#define** UART_IER 1

6.94.1.16 **#define** UART_IER_MSI 0x08

6.94.1.17 **#define** UART_IER_RDI 0x01

6.94.1.18 **#define** UART_IER_RLSI 0x04

6.94.1.19 **#define** UART_IER_THRI 0x02

6.94.1.20 **#define** UART_IIR 2

6.94.1.21 **#define** UART_IIR_CTI 0x0c

6.94.1.22 **#define** UART_IIR_ID 0x06

6.94.1.23 **#define** UART_IIR_MSI 0x00

6.94.1.24 **#define** UART_IIR_NO_INT 0x01

6.94.1.25 **#define** UART_IIR_RDI 0x04

6.94.1.26 **#define** UART_IIR_RLSI 0x06

6.94.1.27 **#define** UART_IIR_THRI 0x02

6.94.1.28 **#define** UART_LCR 3

6.94.1.29 **#define** UART_LCR_DLAB 0x80

6.94.1.30 **#define** UART_LCR_EPAR 0x10

6.94.1.31 **#define** UART_LCR_PARITY 0x08

6.94.1.32 **#define** UART_LCR_RESET 0x03

6.94.1.33 **#define** UART_LCR_SBC 0x40

6.94.1.34 **#define** UART_LCR_SPAR 0x20

6.94.1.35 **#define** UART_LCR_STOP 0x04

6.94.1.36 **#define** UART_LCR_WLEN5 0x00

6.94.1.37 **#define** UART_LCR_WLEN6 0x01

6.94.1.38 **#define** UART_LCR_WLEN7 0x02

6.94.1.39 **#define** UART_LCR_WLEN8 0x03

6.94.1.40 **#define** UART_LSR 5

6.94.1.41 **#define** UART_LSR_BREAK 0x10

6.94.1.42 **#define** UART_LSR_FRAME 0x08

Generated on Tue Nov 11 15:53:42 2009 for QEMU. The OpenRISC 1000 Architectural Simulator by Doxygen

6.94.1.43 **#define** UART_LSR_OVRRUN 0x02

6.94.1.44 **#define** UART_LSR_PARITY 0x04

6.94.1.45 **#define** UART_LSR_RDRDY 0x01

6.94.1.50 **#define UART_MCR 4**

6.94.1.51 **#define UART_MCR_AUX1 0x04**

6.94.1.52 **#define UART_MCR_AUX2 0x08**

6.94.1.53 **#define UART_MCR_DTR 0x01**

6.94.1.54 **#define UART_MCR_LOOP 0x10**

6.94.1.55 **#define UART_MCR_RTS 0x02**

6.94.1.56 **#define UART_MSR 6**

6.94.1.57 **#define UART_MSR_CTS 0x10**

6.94.1.58 **#define UART_MSR_DCD 0x80**

6.94.1.59 **#define UART_MSR_DCTS 0x01**

6.94.1.60 **#define UART_MSR_DDCD 0x08**

6.94.1.61 **#define UART_MSR_DDSR 0x02**

6.94.1.62 **#define UART_MSR_DSR 0x20**

6.94.1.63 **#define UART_MSR_RI 0x40**

6.94.1.64 **#define UART_MSR_TERI 0x04**

6.94.1.65 **#define UART_RXBUF 0**

6.94.1.66 **#define UART_SCR 7**

6.94.1.67 **#define UART_TXBUF 0**

6.94.1.68 **#define UART_VALID_FCR 0xc0**

6.94.1.69 **#define UART_VALID_IER 0x0f**

6.94.1.70 **#define UART_VALID_IIR 0x0f**

6.94.1.71 **#define UART_VALID_LCR 0xff**

6.94.1.72 **#define UART_VALID_LSR 0xff**

6.94.1.73 **#define UART_VALID_MCR 0x1f**

6.94.1.74 **#define UART_VALID_MSR 0xff**

6.94.1.75 **#define UART_VAPI_BUF_LEN 128**

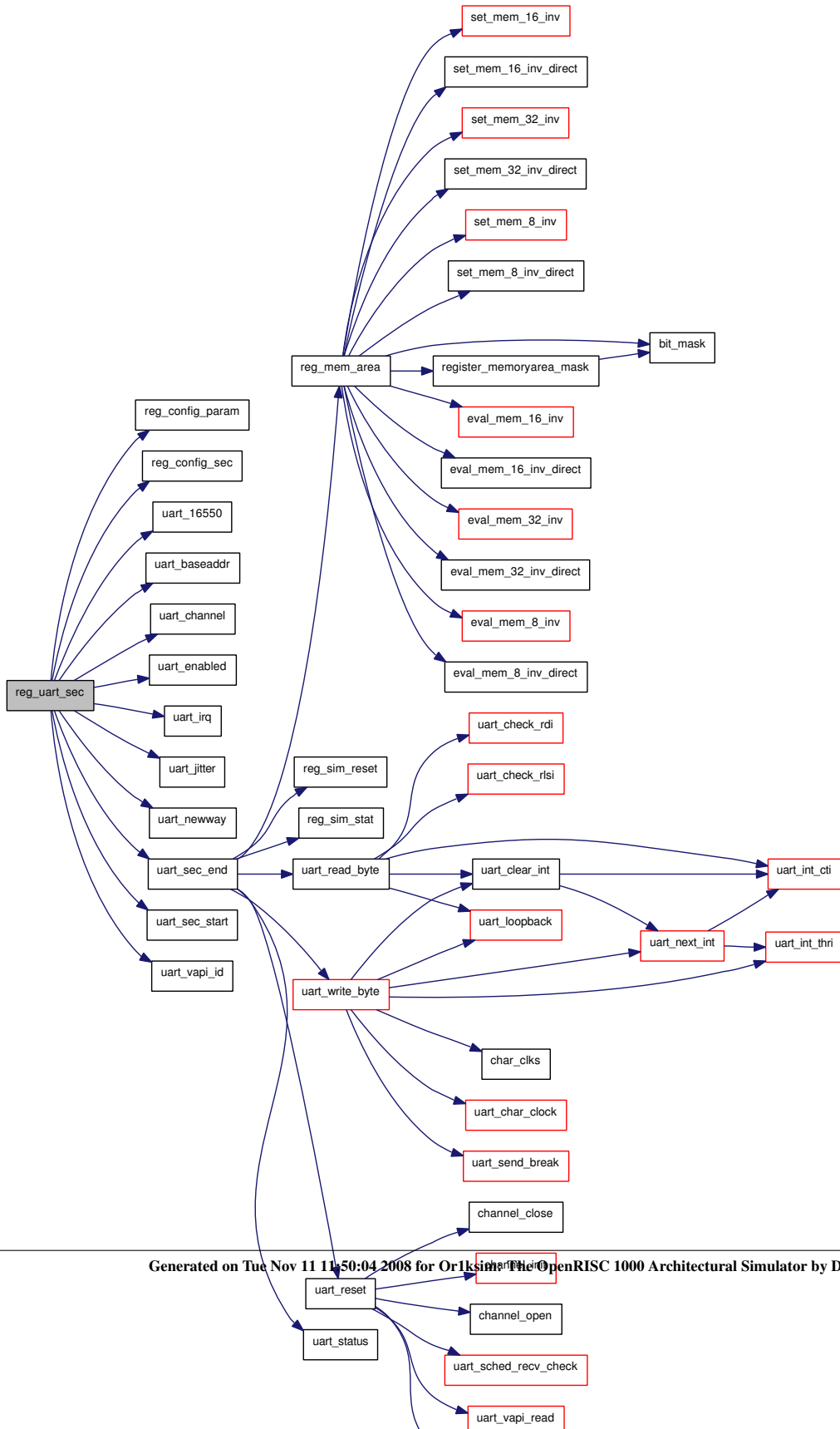
Size of VAPI command buffer

6.94.2 Function Documentation

6.94.2.1 static unsigned long char_clks (int *dll*, int *dlh*, int *lcr*) [static]

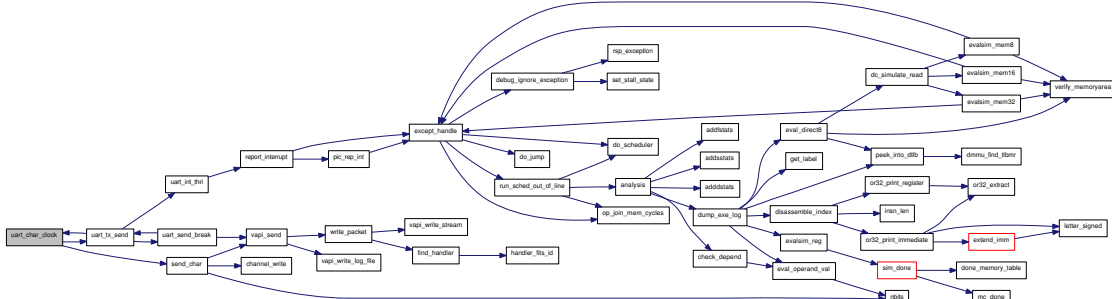
6.94.2.2 void reg_uart_sec (void)

Here is the call graph for this function:



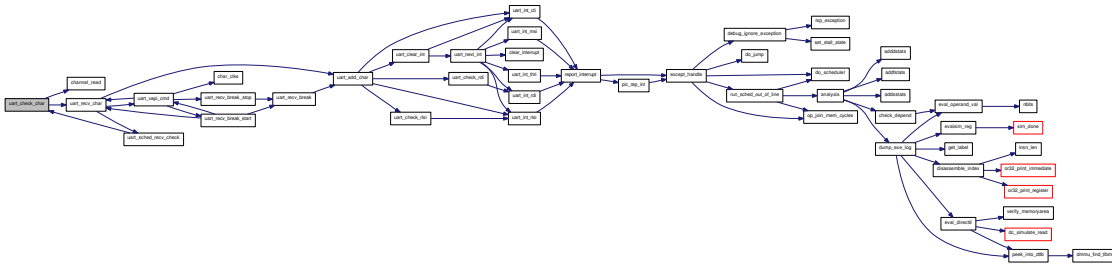
6.94.2.8 void uart_char_clock (void * dat)

Here is the call graph for this function:



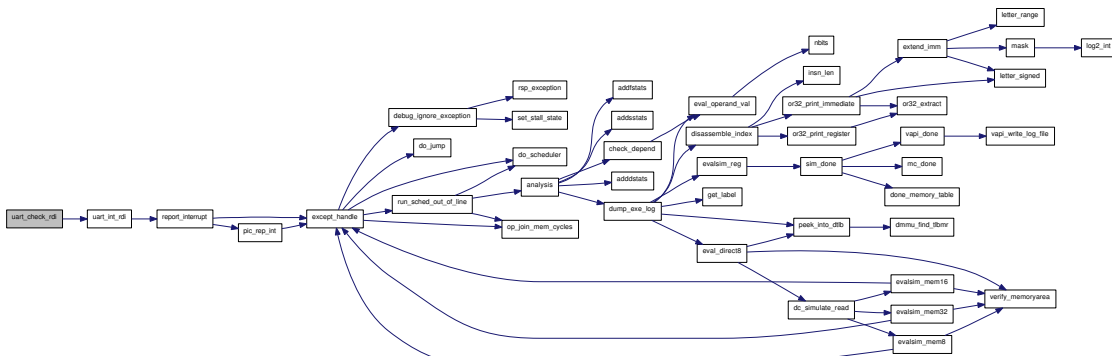
6.94.2.9 static void uart_check_char (void * dat) [static]

Here is the call graph for this function:



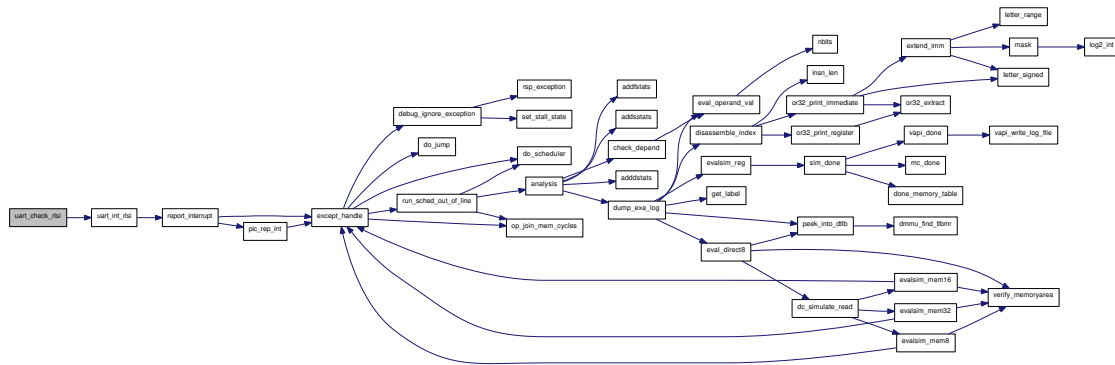
6.94.2.10 static void uart_check_rdi (void * dat) [static]

Here is the call graph for this function:



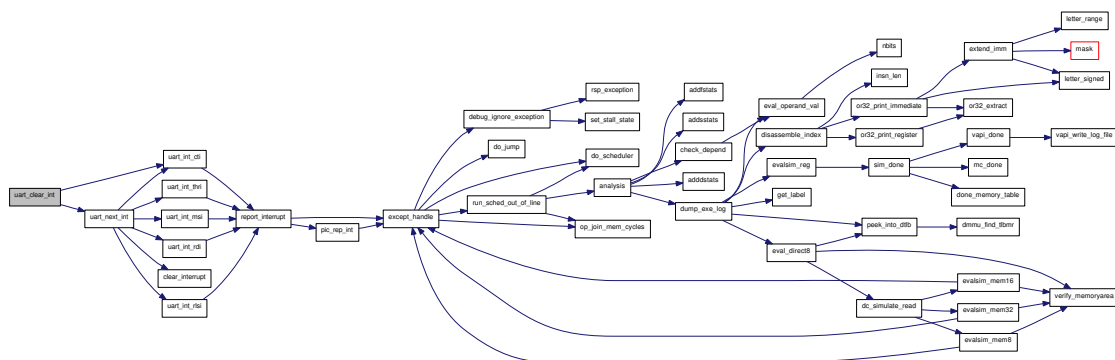
6.94.2.11 static void uart_check_rlsi (void * dat) [static]

Here is the call graph for this function:



6.94.2.12 static void uart_clear_int (struct dev_16450 * uart, int intr) [static]

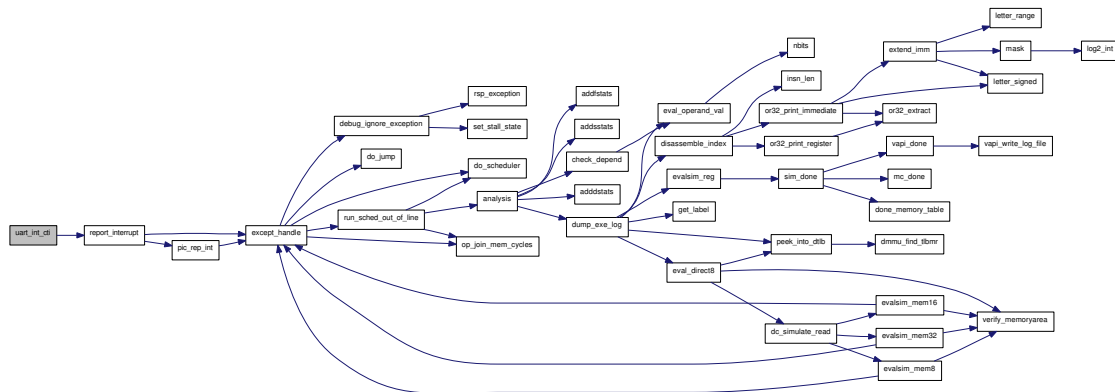
Here is the call graph for this function:



6.94.2.13 `static void uart_enabled (union param_val val, void * dat)` [static]

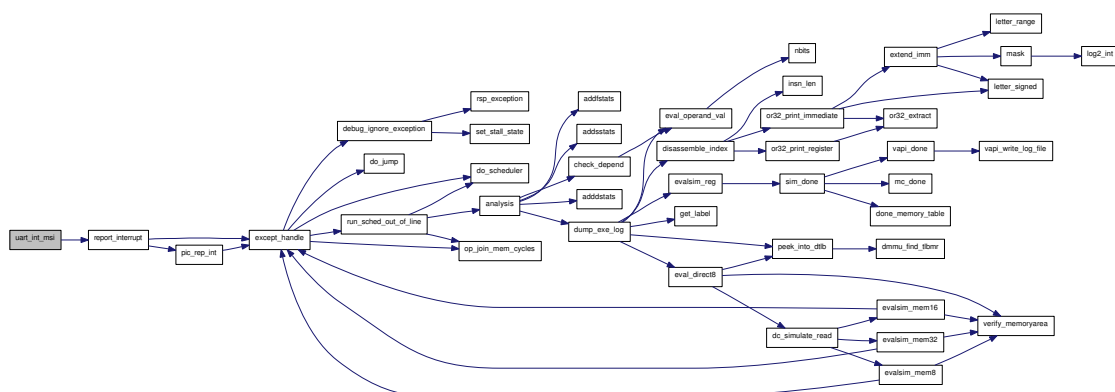
6.94.2.14 `static void uart_int_cti (void * dat)` [static]

Here is the call graph for this function:



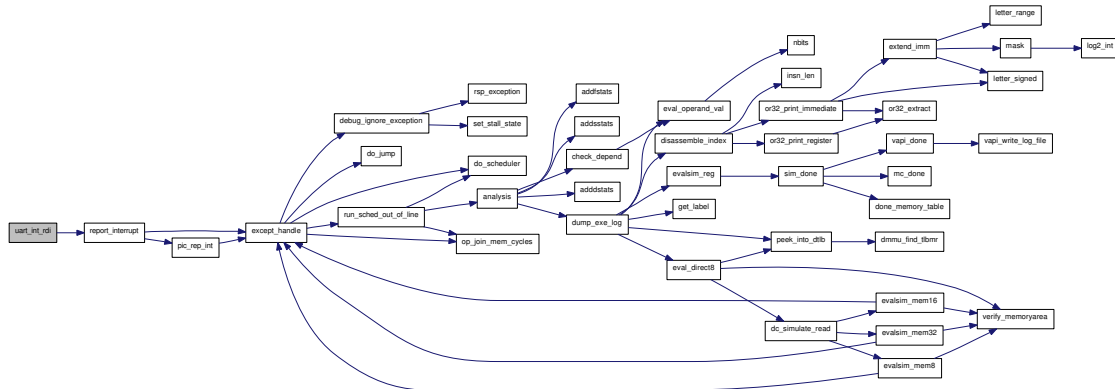
6.94.2.15 `static void uart_int_msi (void * dat)` [static]

Here is the call graph for this function:



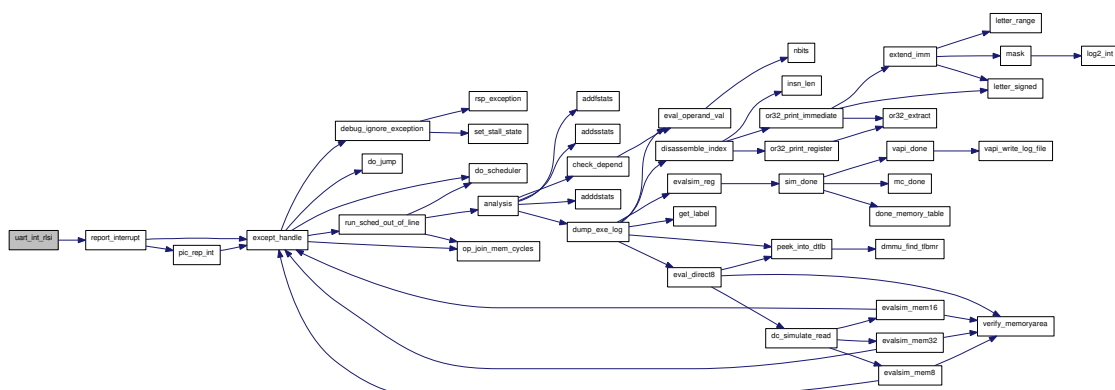
6.94.2.16 static void uart_int_rdi (void * dat) [static]

Here is the call graph for this function:



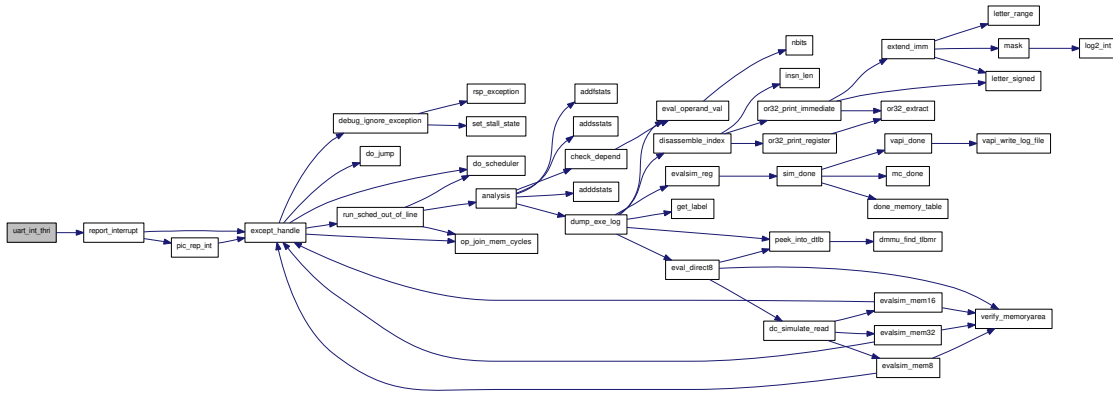
6.94.2.17 static void uart_int_rlsi (void * dat) [static]

Here is the call graph for this function:



6.94.2.18 static void uart_int_thri (void * dat) [static]

Here is the call graph for this function:

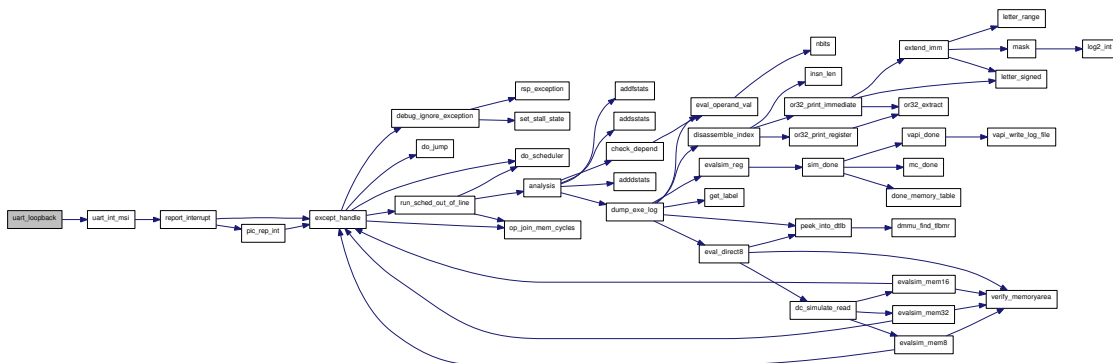


6.94.2.19 static void uart_irq (union param_val val, void * dat) [static]

6.94.2.20 static void uart_jitter (union param_val val, void * dat) [static]

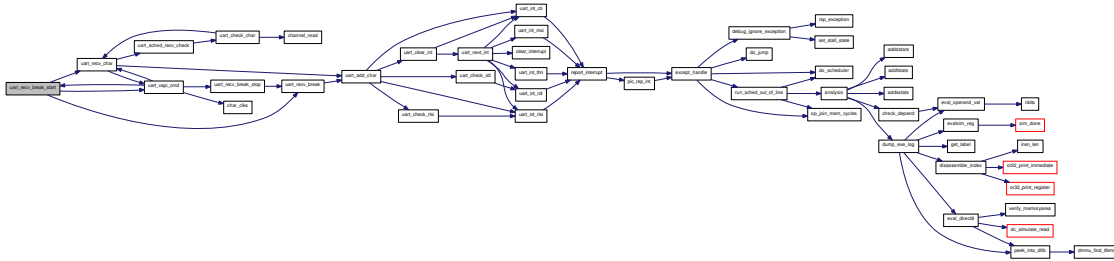
6.94.2.21 static void uart_loopback (struct dev_16450 * uart) [static]

Here is the call graph for this function:



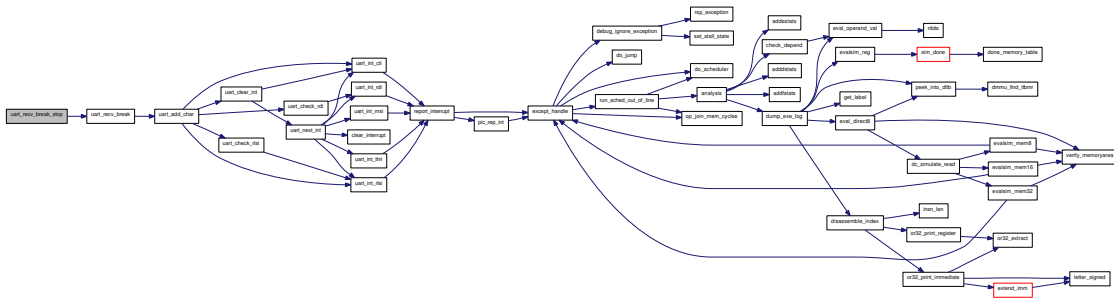
6.94.2.26 static void uart_rcv_break_start (void * dat) [static]

Here is the call graph for this function:



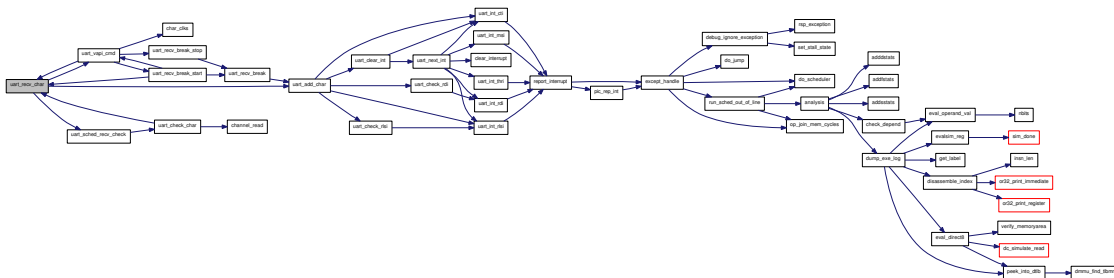
6.94.2.27 static void uart_rcv_break_stop (void * dat) [static]

Here is the call graph for this function:



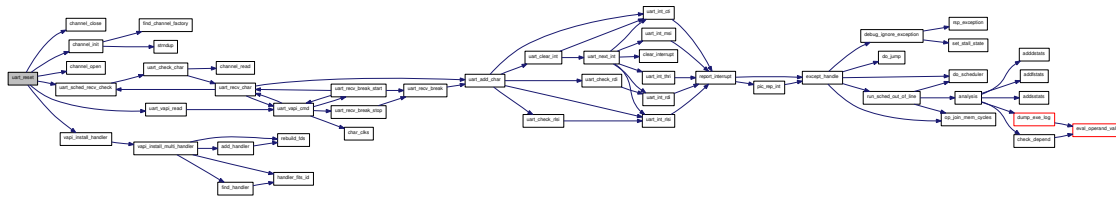
6.94.2.28 static void uart_rcv_char (void * dat) [static]

Here is the call graph for this function:



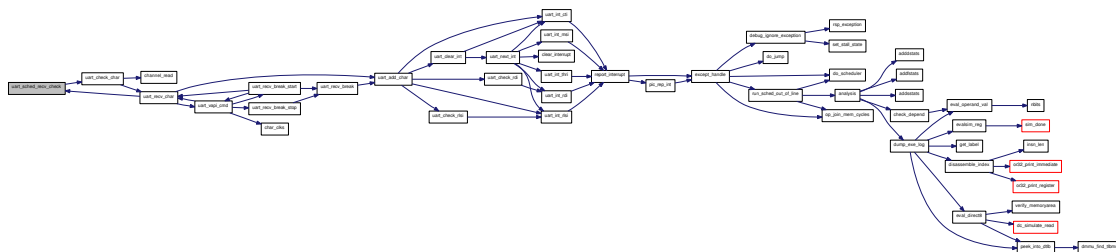
6.94.2.29 void uart_reset (void * dat)

Here is the call graph for this function:



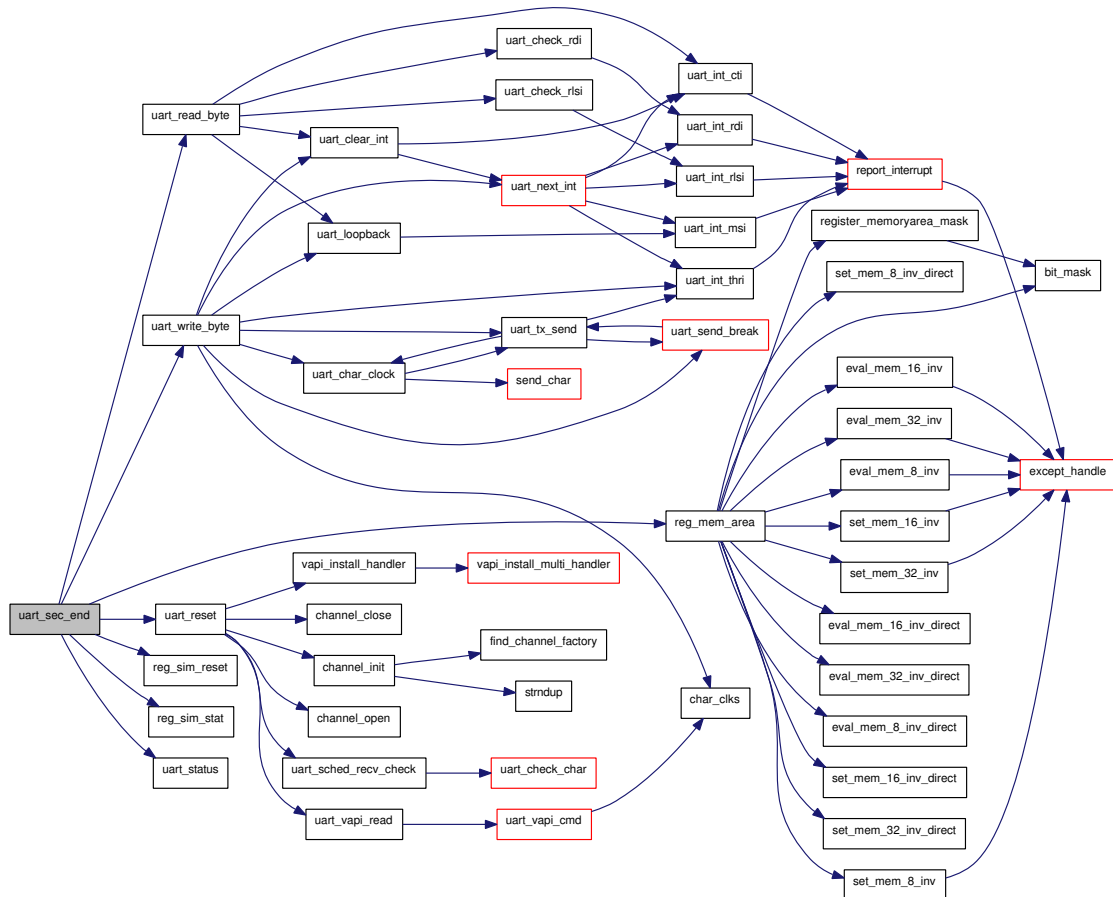
6.94.2.30 static void uart_sched_recv_check (struct dev_16450 * uart) [static]

Here is the call graph for this function:



6.94.2.31 static void uart_sec_end (void * *dat*) [static]

Here is the call graph for this function:



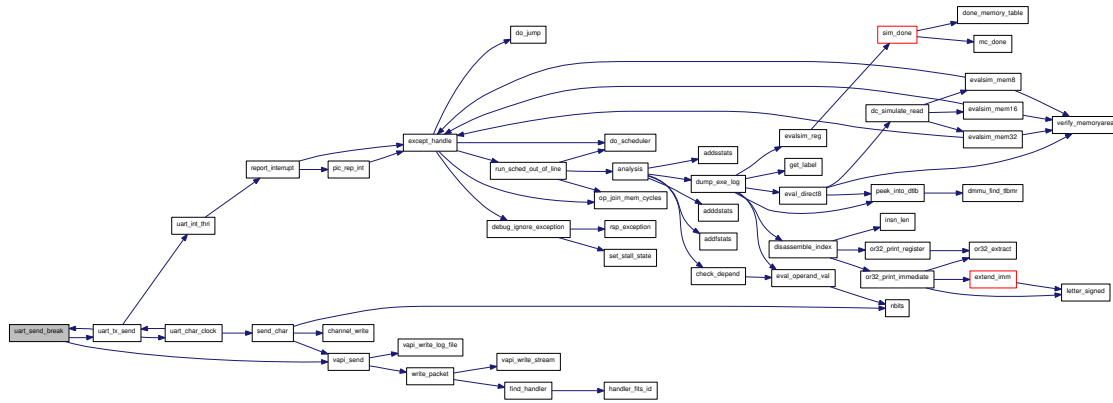
6.94.2.32 static void* uart_sec_start () [static]

Initialize a new UART configuration

ALL parameters are set explicitly to default values.

6.94.2.33 void uart_send_break (void * dat)

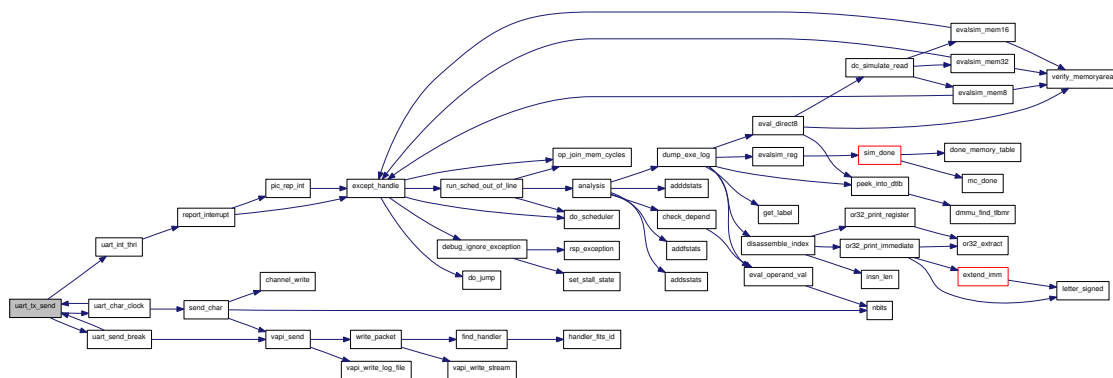
Here is the call graph for this function:



6.94.2.34 void uart_status (void * dat)

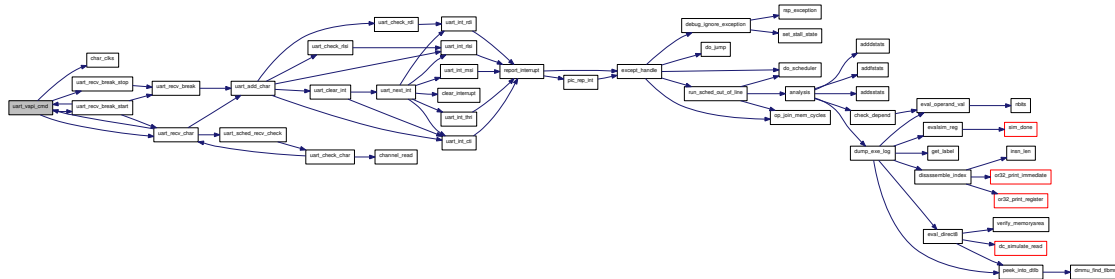
6.94.2.35 void uart_tx_send (void * dat) [static]

Here is the call graph for this function:



6.94.2.36 `static void uart_vapi_cmd (void * dat) [static]`

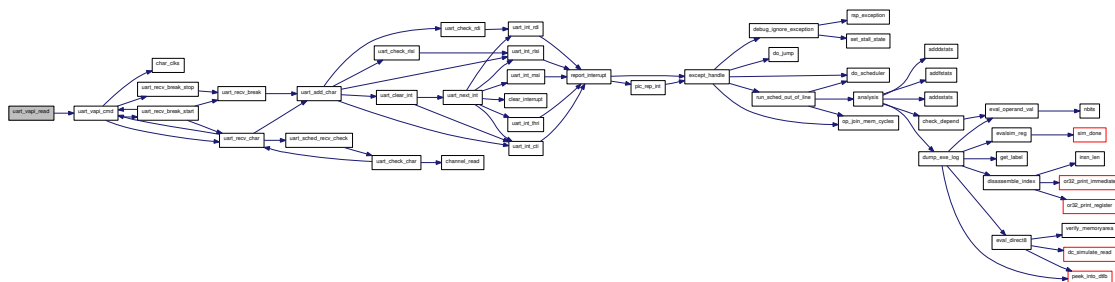
Here is the call graph for this function:



6.94.2.37 `static void uart_vapi_id (union param_val val, void * dat) [static]`

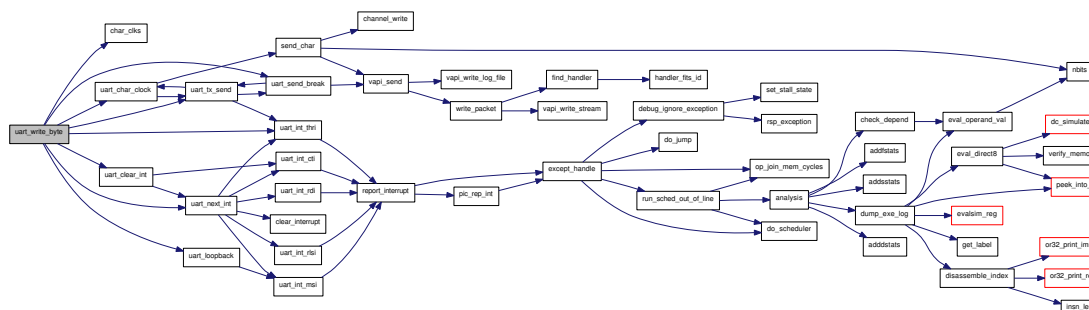
6.94.2.38 `static void uart_vapi_read (unsigned long id, unsigned long data, void * dat) [static]`

Here is the call graph for this function:



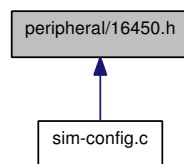
6.94.2.39 `static void uart_write_byte (oraddr_t addr, uint8_t value, void * dat) [static]`

Here is the call graph for this function:



6.95 peripheral/16450.h File Reference

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



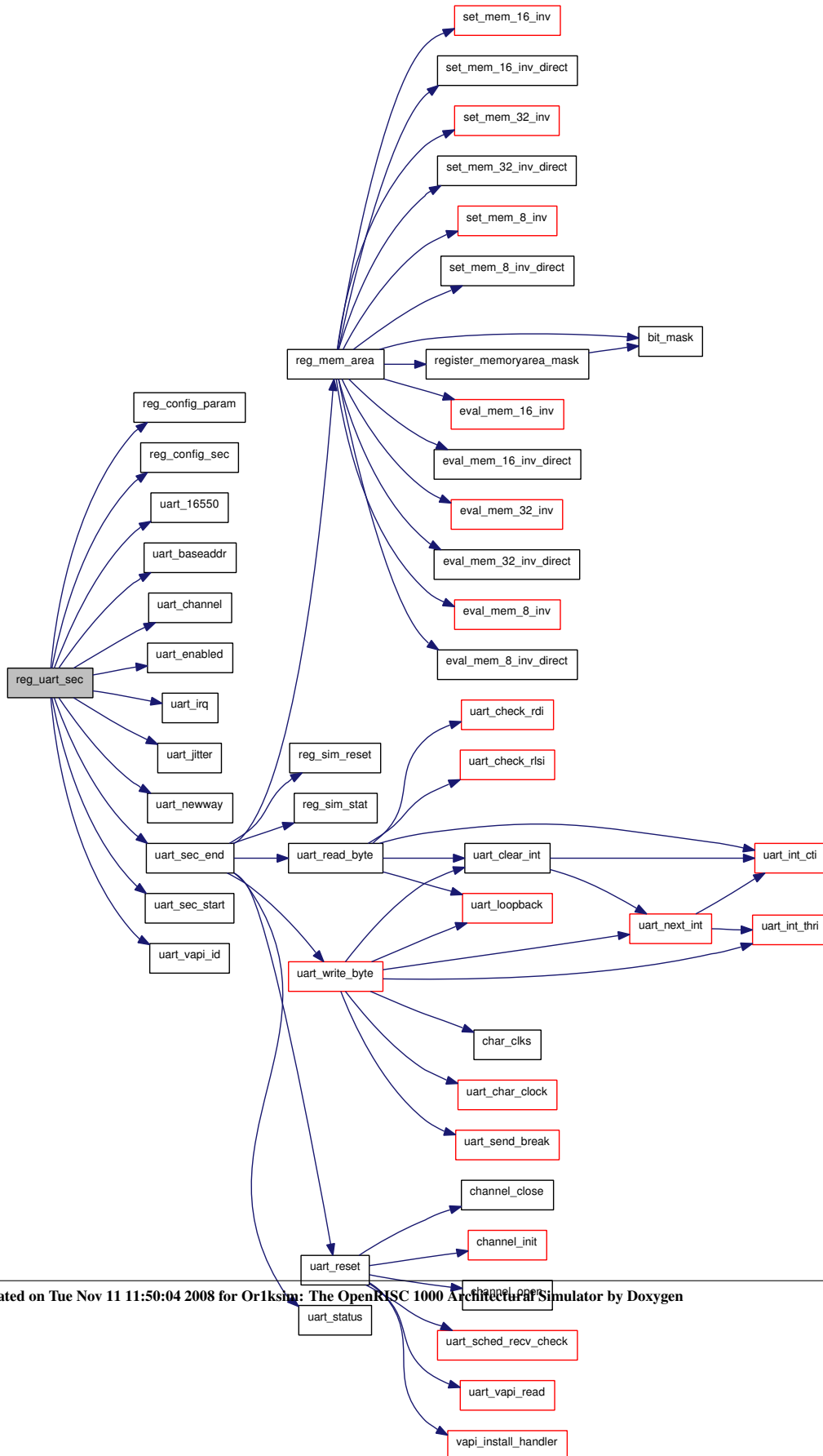
Functions

- void [uart_reset](#) ()
- void [uart_status](#) ()
- void [reg_uart_sec](#) ()

6.95.1 Function Documentation

6.95.1.1 void reg_uart_sec ()

Here is the call graph for this function:

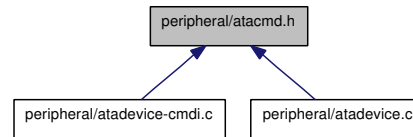


6.95.1.2 void `uart_reset ()`

6.95.1.3 void `uart_status ()`

6.96 peripheral/atacmd.h File Reference

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



Defines

- #define [CFA_ERASE_SECTORS](#) 0xC0
- #define [CFA_REQUEST_EXTENDED_ERROR_CODE](#) 0x03
- #define [CFA_TRANSLATE_SECTOR](#) 0x87
- #define [CFA_WRITE_MULTIPLE_WITHOUT_ERASE](#) 0xCD
- #define [CFA_WRITE_SECTORS_WITHOUT_ERASE](#) 0x38
- #define [CHECK_POWER_MODE](#) 0xE5
- #define [DEVICE_RESET](#) 0x08
- #define [DOWNLOAD_MICROCODE](#) 0x92
- #define [EXECUTE_DEVICE_DIAGNOSTICS](#) 0x90
- #define [FLUSH_CACHE](#) 0xE7
- #define [GET_MEDIA_STATUS](#) 0xDA
- #define [IDENTIFY_DEVICE](#) 0xEC
- #define [IDENTIFY_PACKET_DEVICE](#) 0xA1
- #define [IDLE](#) 0xE3
- #define [IDLE_IMMEDIATE](#) 0xE1
- #define [INITIALIZE_DEVICE_PARAMETERS](#) 0x91
- #define [MEDIA_EJECT](#) 0xED
- #define [MEDIA_LOCK](#) 0xDE
- #define [MEDIA_UNLOCK](#) 0xDF
- #define [NOP](#) 0x00
- #define [PACKET](#) 0xA0
- #define [READ_BUFFER](#) 0xE4
- #define [READ_DMA](#) 0xC8
- #define [READ_DMA_QUEUED](#) 0xC7
- #define [READ_MULTIPLE](#) 0xC4
- #define [READ_NATIVE_MAX_ADDRESS](#) 0xF8
- #define [READ_SECTOR](#) 0x20
- #define [READ_SECTORS](#) 0x20
- #define [READ_VERIFY_SECTOR](#) 0x40
- #define [READ_VERIFY_SECTORS](#) 0x40
- #define [SECURITY_DISABLE_PASSWORD](#) 0xF6
- #define [SECURITY_ERASE_PREPARE](#) 0xF3
- #define [SECURITY_ERASE_UNIT](#) 0xF4
- #define [SECURITY_FREEZE_LOCK](#) 0xF5
- #define [SECURITY_SET_PASSWORD](#) 0xF1
- #define [SECURITY_UNLOCK](#) 0xF2
- #define [SEEK](#) 0x70

- #define SERVICE 0xA2
- #define SET_FEATURES 0xEF
- #define SET_MAX 0xF9
- #define SET_MULTIPLE_MODE 0xC6
- #define SLEEP 0xE6
- #define SMART 0xB0
- #define STANDBY 0xE2
- #define STANDBY_IMMEDIATE 0xE0
- #define WRITE_BUFFER 0xE8
- #define WRITE_DMA 0xCA
- #define WRITE_DMA_QUEUED 0xCC
- #define WRITE_MULTIPLE 0xC5
- #define WRITE_SECTOR 0x30
- #define WRITE_SECTORS 0x30
- #define CFA_ENABLE_8BIT_PIO_TRANSFER_MODE 0x01
- #define ENABLE_WRITE_CACHE 0x02
- #define SET_TRANSFER_MODE_SECTOR_COUNT_REG 0x03
- #define ENABLE_ADVANCED_POWER_MANAGEMENT 0x05
- #define ENABLE_POWERUP_IN_STANDBY_FEATURE_SET 0x06
- #define POWERUP_IN_STANDBY_FEATURE_SET_SPINUP 0x07
- #define CFA_ENABLE_POWER_MODE1 0x0A
- #define DISABLE_MEDIA_STATUS_NOTIFICATION 0x31
- #define DISABLE_READ_LOOKAHEAD 0x55
- #define ENABLE_RELEASE_INTERRUPT 0x5D
- #define ENABLE_SERVICE_INTERRUPT 0x5E
- #define DISABLE_REVERTING_TO_POWERON_DEFAULTS 0x66
- #define CFA_DISABLE_8BIT_PIO_TRANSFER_MODE 0x81
- #define DISABLE_WRITE_CACHE 0x82
- #define DISABLE_ADVANCED_POWER_MANAGEMENT 0x85
- #define DISABLE_POWERUP_IN_STANDBY_FEATURE_SET 0x86
- #define CFA_DISABLE_POWER_MODE1 0x8A
- #define ENABLE_MEDIA_STATUS_NOTIFICATION 0x95
- #define ENABLE_READ_LOOKAHEAD_FEATURE 0xAA
- #define ENABLE_REVERTING_TO_POWERON_DEFAULTS 0xCC
- #define DISABLE_RELEASE_INTERRUPT 0xDD
- #define DISABLE_SERVICE_INTERRUPT 0xDE
- #define SET_MAX_ADDRESS 0x00
- #define SET_MAX_SET_PASSWORD 0x01
- #define SET_MAX_LOCK 0x02
- #define SET_MAX_UNLOCK 0x03
- #define SET_MAX_FREEZE_LOCK 0x04
- #define SMART_READ_DATA 0xD0
- #define SMART_ATTRIBITE_AUTOSAVE 0xD1
- #define SMART_SAVE_ATTRIBUTE_VALUES 0xD3
- #define SMART_EXECUTE_OFFLINE_IMMEDIATE 0xD4
- #define SMART_READ_LOG 0xD5
- #define SMART_WRITE_LOG 0xD6
- #define SMART_ENABLE_OPERATIONS 0xD8
- #define SMART_DISABLE_OPERATIONS 0xD9
- #define SMART_RETURN_STATUS 0xDA

6.96.1 Define Documentation

6.96.1.1 #define CFA_DISABLE_8BIT_PIO_TRANSFER_MODE 0x81

6.96.1.2 #define CFA_DISABLE_POWER_MODE1 0x8A

6.96.1.3 #define CFA_ENABLE_8BIT_PIO_TRANSFER_MODE 0x01

6.96.1.4 #define CFA_ENABLE_POWER_MODE1 0x0A

6.96.1.5 #define CFA_ERASE_SECTORS 0xC0

6.96.1.6 #define CFA_REQUEST_EXTENDED_ERROR_CODE 0x03

6.96.1.7 #define CFA_TRANSLATE_SECTOR 0x87

6.96.1.8 #define CFA_WRITE_MULTIPLE_WITHOUT_ERASE 0xCD

6.96.1.9 #define CFA_WRITE_SECTORS_WITHOUT_ERASE 0x38

6.96.1.10 #define CHECK_POWER_MODE 0xE5

6.96.1.11 #define DEVICE_RESET 0x08

6.96.1.12 #define DISABLE_ADVANCED_POWER_MANAGEMENT 0x85

6.96.1.13 #define DISABLE_MEDIA_STATUS_NOTIFICATION 0x31

6.96.1.14 #define DISABLE_POWERUP_IN_STANDBY_FEATURE_SET 0x86

6.96.1.15 #define DISABLE_READ_LOOKAHEAD 0x55

6.96.1.16 #define DISABLE_RELEASE_INTERRUPT 0xDD

6.96.1.17 #define DISABLE_REVERTING_TO_POWERON_DEFAULTS 0x66

6.96.1.18 #define DISABLE_SERVICE_INTERRUPT 0xDE

6.96.1.19 #define DISABLE_WRITE_CACHE 0x82

6.96.1.20 #define DOWNLOAD_MICROCODE 0x92

6.96.1.21 #define ENABLE_ADVANCED_POWER_MANAGEMENT 0x05

6.96.1.22 #define ENABLE_MEDIA_STATUS_NOTIFICATION 0x95

6.96.1.23 #define ENABLE_POWERUP_IN_STANDBY_FEATURE_SET 0x06

6.96.1.24 #define ENABLE_READ_LOOKAHEAD_FEATURE 0xAA

6.96.1.25 #define ENABLE_RELEASE_INTERRUPT 0x5D

6.96.1.26 #define ENABLE_REVERTING_TO_POWERON_DEFAULTS 0xCC

6.96.1.27 #define ENABLE_SERVICE_INTERRUPT 0x5F

6.96.1.28 #define ENABLE_WRITE_CACHE 0x02

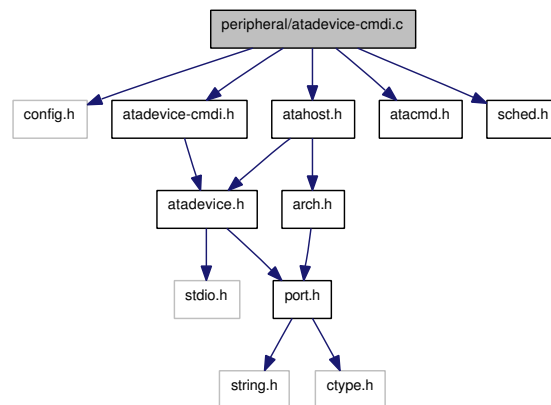
6.96.1.29 #define EXECUTE_DEVICE_DIAGNOSTICS 0x90

6.96.1.30 #define FLUSH_CACHE 0xE7

6.97 peripheral/atadevice-cmdi.c File Reference

```
#include "config.h"
#include "atadevice-cmdi.h"
#include "atahost.h"
#include "atacmd.h"
#include "sched.h"
```

Include dependency graph for atadevice-cmdi.c:



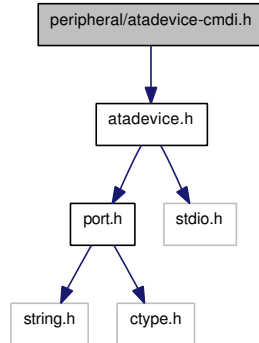
Functions

- static void [ata_cmd_complete](#) (void *dat)
- static void [ata_set_sect](#) (struct [ata_device](#) *dev, uint32_t sect)
- static uint32_t [ata_calc_lba](#) (struct [ata_device](#) *dev)
- static void [ata_read_sect](#) (struct [ata_device](#) *dev)
- static void [ata_write_sect](#) (struct [ata_device](#) *dev)
- static void [ata_set_device_signature](#) (struct [ata_device](#) *device, int packet)
- void [ata_execute_device_diagnostics_cmd](#) (struct [ata_device](#) *device)
- static void [ata_device_reset_cmd](#) (struct [ata_device](#) *device)
- static void [ata_identify_device_cmd](#) (struct [ata_device](#) *device)
- static void [ata_initialize_device_parameters_cmd](#) (struct [ata_device](#) *device)
- static void [ata_read_sectors_cmd](#) (struct [ata_device](#) *device)
- static void [ata_read_native_max_addr](#) (struct [ata_device](#) *dev)
- static void [ata_write_sectors](#) (struct [ata_device](#) *dev)
- static void [ata_set_features](#) (struct [ata_device](#) *dev)
- int [ata_device_execute_cmd](#) (struct [ata_device](#) *device)

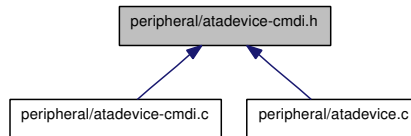
6.98 peripheral/atadevice-cmdi.h File Reference

```
#include "atadevice.h"
```

Include dependency graph for atadevice-cmdi.h:



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



Defines

- #define [BYTES_PER_SECTOR](#) 512
- #define [MIN_MWDMA_CYCLE_TIME](#) 100
- #define [RECOMMENDED_MWDMA_CYCLE_TIME](#) 100
- #define [MIN_PIO_CYCLE_TIME_NO_IORDY](#) 100
- #define [MIN_PIO_CYCLE_TIME_IORDY](#) 100
- #define [SUPPORT_NOP_CMD](#) 0
- #define [SUPPORT_READ_BUFFER_CMD](#) 0
- #define [SUPPORT_WRITE_BUFFER_CMD](#) 0
- #define [SUPPORT_HOST_PROTECTED_AREA](#) 0
- #define [SUPPORT_DEVICE_RESET_CMD](#) 1
- #define [SUPPORT_SERVICE_INTERRUPT](#) 0
- #define [SUPPORT_RELEASE_INTERRUPT](#) 0
- #define [SUPPORT_LOOKAHEAD](#) 0
- #define [SUPPORT_WRITE_CACHE](#) 0
- #define [SUPPORT_POWER_MANAGEMENT](#) 0
- #define [SUPPORT_REMOVABLE_MEDIA](#) 0
- #define [SUPPORT_SECURITY_MODE](#) 0
- #define [SUPPORT_SMART](#) 0
- #define [SUPPORT_SET_MAX](#) 0
- #define [SET_FEATURES_REQUIRED_AFTER_POWER_UP](#) 0
- #define [SUPPORT_POWER_UP_IN_STANDBY_MODE](#) 0

- #define [SUPPORT_REMOVABLE_MEDIA_NOTIFICATION](#) 0

- #define [SUPPORT_APM](#) 0

- #define [SUPPORT_CFA](#) 0

- #define [SUPPORT_READ_WRITE_DMA_QUEUED](#) 0

- #define [SUPPORT_DOWNLOAD_MICROCODE](#) 0

- #define [QUEUE_DEPTH](#) 0

Functions

- int [ata_device_execute_cmd](#) (struct [ata_device](#) *device)

- void [ata_execute_device_diagnostics_cmd](#) (struct [ata_device](#) *device)

6.98.1 Define Documentation

- 6.98.1.1 `#define BYTES_PER_SECTOR 512`
- 6.98.1.2 `#define MIN_MWDMA_CYCLE_TIME 100`
- 6.98.1.3 `#define MIN_PIO_CYCLE_TIME_IORDY 100`
- 6.98.1.4 `#define MIN_PIO_CYCLE_TIME_NO_IORDY 100`
- 6.98.1.5 `#define QUEUE_DEPTH 0`
- 6.98.1.6 `#define RECOMMENDED_MWDMA_CYCLE_TIME 100`
- 6.98.1.7 `#define SET_FEATURES_REQUIRED_AFTER_POWER_UP 0`
- 6.98.1.8 `#define SUPPORT_APM 0`
- 6.98.1.9 `#define SUPPORT_CFA 0`
- 6.98.1.10 `#define SUPPORT_DEVICE_RESET_CMD 1`
- 6.98.1.11 `#define SUPPORT_DOWNLOAD_MICROCODE 0`
- 6.98.1.12 `#define SUPPORT_HOST_PROTECTED_AREA 0`
- 6.98.1.13 `#define SUPPORT_LOOKAHEAD 0`
- 6.98.1.14 `#define SUPPORT_NOP_CMD 0`
- 6.98.1.15 `#define SUPPORT_POWER_MANAGEMENT 0`
- 6.98.1.16 `#define SUPPORT_POWER_UP_IN_STANDBY_MODE 0`
- 6.98.1.17 `#define SUPPORT_READ_BUFFER_CMD 0`
- 6.98.1.18 `#define SUPPORT_READ_WRITE_DMA_QUEUED 0`
- 6.98.1.19 `#define SUPPORT_RELEASE_INTERRUPT 0`
- 6.98.1.20 `#define SUPPORT_REMOVABLE_MEDIA 0`
- 6.98.1.21 `#define SUPPORT_REMOVABLE_MEDIA_NOTIFICATION 0`
- 6.98.1.22 `#define SUPPORT_SECURITY_MODE 0`
- 6.98.1.23 `#define SUPPORT_SERVICE_INTERRUPT 0`
- 6.98.1.24 `#define SUPPORT_SET_MAX 0`
- 6.98.1.25 `#define SUPPORT_SMART 0`
- 6.98.1.26 `#define SUPPORT_WRITE_BUFFER_CMD 0`

6.98.1.27 `#define SUPPORT_WRITE_CACHE 0`
Generated on Tue Nov 11 14:50:04 2008 for OPIsim: The OpenRISC 1000 Architectural Simulator by Doxygen

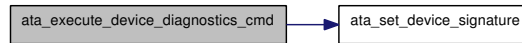
6.98.2 Function Documentation

- 6.98.2.1 `int ata_device_execute_cmd (struct ata_device * device)`

Here is the call graph for this function:

6.98.2.2 void ata_execute_device_diagnostics_cmd (struct ata_device * device)

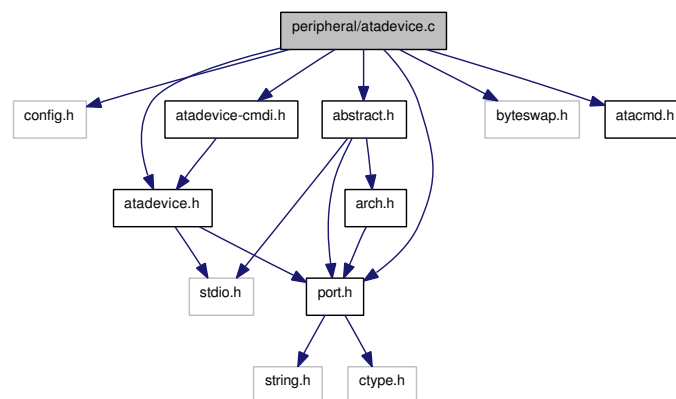
Here is the call graph for this function:



6.99 peripheral/atadevice.c File Reference

```
#include "config.h"
#include "port.h"
#include <byteswap.h>
#include "atadevice.h"
#include "atadevice-cmdi.h"
#include "atacmd.h"
#include "abstract.h"
```

Include dependency graph for atadevice.c:



Functions

- static FILE * [open_file](#) (uint32_t *size, const char *filename)
- static FILE * [open_local](#) (void)
- static void [ata_device_init](#) (struct [ata_device](#) *device, int dev)
- void [ata_devices_init](#) (struct [ata_devices](#) *devices)
- static void [ata_device_hw_reset](#) (struct [ata_device](#) *device, int reset_signal, int daspo, int pdiagi, int daspi)
- void [ata_devices_hw_reset](#) (struct [ata_devices](#) *devices, int reset_signal)
- static void [ata_device_do_control_register](#) (struct [ata_device](#) *device)
- static void [ata_device_do_command_register](#) (struct [ata_device](#) *device)
- short [ata_devices_read](#) (struct [ata_devices](#) *devices, char adr)
- static void [ata_device_write](#) (struct [ata_device](#) *device, char adr, short value)
- void [ata_devices_write](#) (struct [ata_devices](#) *devices, char adr, short value)

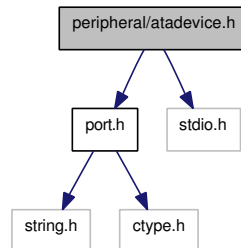
6.99.1.10 static FILE* `open_file` (uint32_t * *size*, const char * *filename*) [static]

6.99.1.11 static FILE* `open_local` (void) [static]

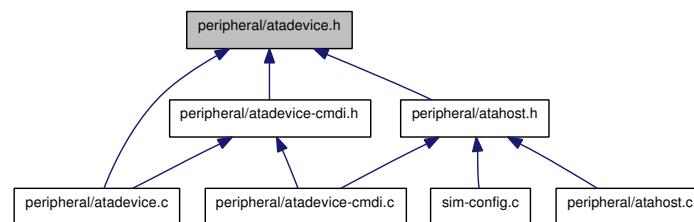
6.100 peripheral/atadevice.h File Reference

```
#include "port.h"
#include <stdio.h>
```

Include dependency graph for atadevice.h:



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



Data Structures

- struct [ata_device](#)
- struct [ata_devices](#)

Defines

- #define [ATA_ASR](#) 0x78
- #define [ATA_CR](#) 0x5c
- #define [ATA_CHR](#) 0x54
- #define [ATA_CLR](#) 0x50
- #define [ATA_DR](#) 0x40
- #define [ATA_DCR](#) 0x78
- #define [ATA_DHR](#) 0x58
- #define [ATA_ERR](#) 0x44
- #define [ATA_FR](#) 0x44
- #define [ATA_SCR](#) 0x48
- #define [ATA_SNR](#) 0x4c
- #define [ATA_SR](#) 0x5c
- #define [ATA_DA](#) 0x7c
- #define [ATA_SR_BSY](#) 0x80
- #define [ATA_SR_DRDY](#) 0x40

- #define [ATA_SR_DF](#) 0x20
- #define [ATA_SR_DSC](#) 0x10
- #define [ATA_SR_DRQ](#) 0x08
- #define [ATA_SR_COR](#) 0x04
- #define [ATA_SR_IDX](#) 0x02
- #define [ATA_SR_ERR](#) 0x01
- #define [ATA_DCR_RST](#) 0x04
- #define [ATA_DCR_IEN](#) 0x02
- #define [ATA_DAR_WTG](#) 0x40
- #define [ATA_DAR_H](#) 0x3c
- #define [ATA_DAR_DS1](#) 0x02
- #define [ATA_DAR_DS0](#) 0x01
- #define [ATA_DHR_LBA](#) 0x40
- #define [ATA_DHR_DEV](#) 0x10
- #define [ATA_DHR_H](#) 0x0f
- #define [ATA_ERR_BBK](#) 0x80
- #define [ATA_ERR_UNC](#) 0x40
- #define [ATA_ERR_IDNF](#) 0x10
- #define [ATA_ERR_ABT](#) 0x04
- #define [ATA_ERR_TON](#) 0x02
- #define [ATA_ERR_AMN](#) 0x01
- #define [TYPE_NO_CONNECT](#) 0
- #define [TYPE_FILE](#) 1
- #define [TYPE_LOCAL](#) 2
- #define [ATA_STATE_IDLE](#) 0x00
- #define [ATA_STATE_SW_RST](#) 0x01
- #define [ATA_STATE_HW_RST](#) 0x02

Functions

- void [ata_devices_init](#) (struct [ata_devices](#) *devices)
- void [ata_devices_hw_reset](#) (struct [ata_devices](#) *devices, int reset_signal)
- short [ata_devices_read](#) (struct [ata_devices](#) *devices, char adr)
- void [ata_devices_write](#) (struct [ata_devices](#) *devices, char adr, short value)

6.100.1 Define Documentation

6.100.1.1 #define ATA_ASR 0x78

6.100.1.2 #define ATA_CHR 0x54

6.100.1.3 #define ATA_CLR 0x50

6.100.1.4 #define ATA_CR 0x5c

6.100.1.5 #define ATA_DA 0x7c

6.100.1.6 #define ATA_DAR_DS0 0x01

6.100.1.7 #define ATA_DAR_DS1 0x02

6.100.1.8 #define ATA_DAR_H 0x3c

6.100.1.9 #define ATA_DAR_WTG 0x40

6.100.1.10 #define ATA_DCR 0x78

6.100.1.11 #define ATA_DCR_IEN 0x02

6.100.1.12 #define ATA_DCR_RST 0x04

6.100.1.13 #define ATA_DHR 0x58

6.100.1.14 #define ATA_DHR_DEV 0x10

6.100.1.15 #define ATA_DHR_H 0x0f

6.100.1.16 #define ATA_DHR_LBA 0x40

6.100.1.17 #define ATA_DR 0x40

6.100.1.18 #define ATA_ERR 0x44

6.100.1.19 #define ATA_ERR_ABT 0x04

6.100.1.20 #define ATA_ERR_AMN 0x01

6.100.1.21 #define ATA_ERR_BBK 0x80

6.100.1.22 #define ATA_ERR_IDNF 0x10

6.100.1.23 #define ATA_ERR_TON 0x02

6.100.1.24 #define ATA_ERR_UNC 0x40

6.100.1.25 #define ATA_FR 0x44

6.100.1.26 #define ATA_SCR 0x48

6.100.1.27 #define ATA_SNR 0x4c

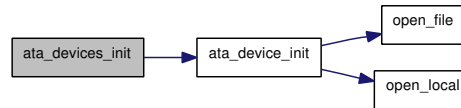
6.100.1.28 #define ATA_SR 0x5c

6.100.1.29 #define ATA_SR_BSY 0x80

6.100.1.30 #define ATA_SR_COR 0x04

6.100.2.2 void ata_devices_init (struct ata_devices * devices)

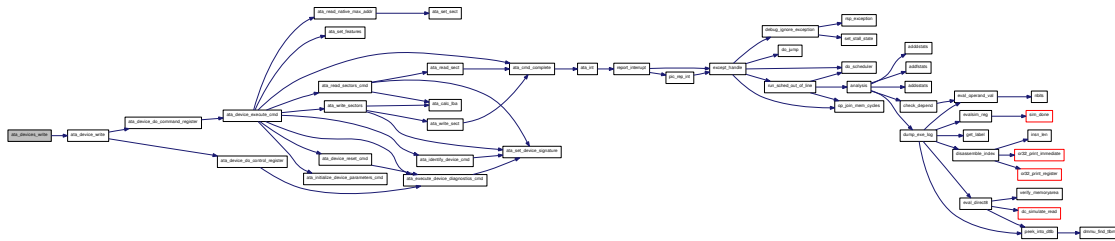
Here is the call graph for this function:



6.100.2.3 short ata_devices_read (struct ata_devices * devices, char adr)

6.100.2.4 void ata_devices_write (struct ata_devices * devices, char adr, short value)

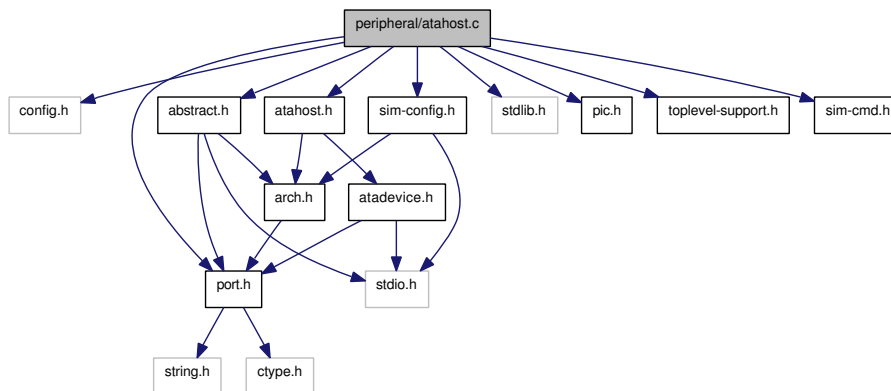
Here is the call graph for this function:



6.101 peripheral/atahost.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "atahost.h"
#include "sim-config.h"
#include "abstract.h"
#include "pic.h"
#include "toplevel-support.h"
#include "sim-cmd.h"
```

Include dependency graph for atahost.c:



Defines

- #define [PIO_MODE0_T1](#) 6
- #define [PIO_MODE0_T2](#) 28
- #define [PIO_MODE0_T4](#) 2
- #define [PIO_MODE0_TEOC](#) 23
- #define [DMA_MODE0_TM](#) 4
- #define [DMA_MODE0_TD](#) 21
- #define [DMA_MODE0_TEOC](#) 21

Functions

- static void [ata_reset](#) (void *dat)
- void [ata_int](#) (void *dat)
- static uint32_t [ata_read32](#) (oraddr_t addr, void *dat)
- static void [ata_write32](#) (oraddr_t addr, uint32_t value, void *dat)
- static void [ata_status](#) (void *dat)
- static void [ata_baseaddr](#) (union [param_val](#) val, void *dat)
- static void [ata_irq](#) (union [param_val](#) val, void *dat)

- static void [ata_dev_id](#) (union [param_val](#) val, void *dat)
- static void [ata_rev](#) (union [param_val](#) val, void *dat)
- static void [ata_pio_mode0_t1](#) (union [param_val](#) val, void *dat)
- static void [ata_pio_mode0_t2](#) (union [param_val](#) val, void *dat)
- static void [ata_pio_mode0_t4](#) (union [param_val](#) val, void *dat)
- static void [ata_pio_mode0_tec](#) (union [param_val](#) val, void *dat)
- static void [ata_dma_mode0_tm](#) (union [param_val](#) val, void *dat)
- static void [ata_dma_mode0_td](#) (union [param_val](#) val, void *dat)
- static void [ata_dma_mode0_tec](#) (union [param_val](#) val, void *dat)
- static void [ata_type](#) (union [param_val](#) val, void *dat)
- static void [ata_file](#) (union [param_val](#) val, void *dat)
- static void [ata_size](#) (union [param_val](#) val, void *dat)
- static void [ata_packet](#) (union [param_val](#) val, void *dat)
- static void [ata_enabled](#) (union [param_val](#) val, void *dat)
- static void [ata_heads](#) (union [param_val](#) val, void *dat)
- static void [ata_sectors](#) (union [param_val](#) val, void *dat)
- static void [ata_firmware](#) (union [param_val](#) val, void *dat)
- static void [ata_mwdma](#) (union [param_val](#) val, void *dat)
- static void [ata_pio](#) (union [param_val](#) val, void *dat)
- static void [ata_start_device](#) (union [param_val](#) val, void *dat)
- static void [ata_enddevice](#) (union [param_val](#) val, void *dat)
- static void * [ata_sec_start](#) (void)
- static void [ata_sec_end](#) (void *dat)
- void [reg_ata_sec](#) ()

Variables

- static unsigned int [conf_dev](#)

6.101.1 Define Documentation

6.101.1.1 `#define DMA_MODE0_TD 21`

6.101.1.2 `#define DMA_MODE0_TEOC 21`

6.101.1.3 `#define DMA_MODE0_TM 4`

6.101.1.4 `#define PIO_MODE0_T1 6`

6.101.1.5 `#define PIO_MODE0_T2 28`

6.101.1.6 `#define PIO_MODE0_T4 2`

6.101.1.7 `#define PIO_MODE0_TEOC 23`

6.101.2 Function Documentation

6.101.2.1 `static void ata_baseaddr (union param_val val, void * dat) [static]`

6.101.2.2 `static void ata_dev_id (union param_val val, void * dat) [static]`

6.101.2.3 `static void ata_dma_mode0_td (union param_val val, void * dat) [static]`

6.101.2.4 `static void ata_dma_mode0_teoc (union param_val val, void * dat) [static]`

6.101.2.5 `static void ata_dma_mode0_tm (union param_val val, void * dat) [static]`

6.101.2.6 `static void ata_enabled (union param_val val, void * dat) [static]`

6.101.2.7 `static void ata_enddevice (union param_val val, void * dat) [static]`

6.101.2.8 `static void ata_file (union param_val val, void * dat) [static]`

Set the ATA file

Free any previously allocated value. Only used if device type is 1.

Parameters:

← *val* The value to use

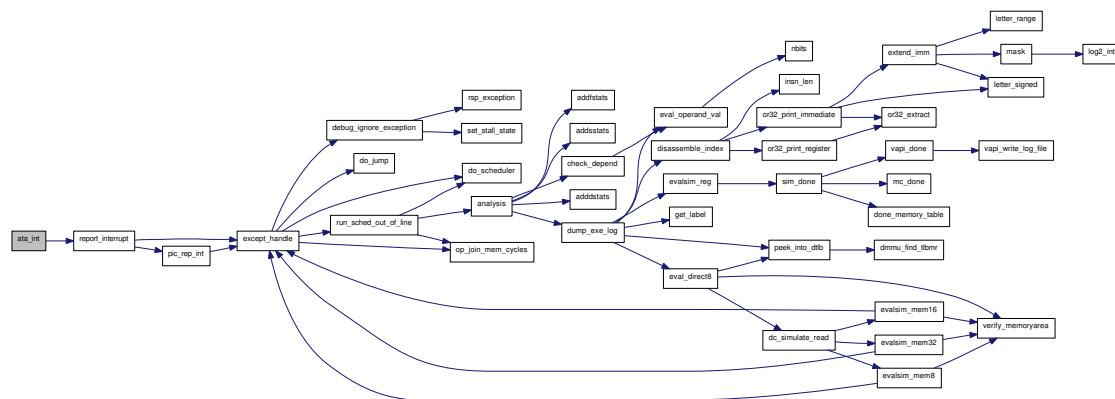
← *dat* The `config` data structure

6.101.2.9 `static void ata_firmware (union param_val val, void * dat)` [static]

6.101.2.10 `static void ata_heads (union param_val val, void * dat)` [static]

6.101.2.11 `void ata_int (void * dat)`

Here is the call graph for this function:



6.101.2.12 `static void ata_irq (union param_val val, void * dat)` [static]

6.101.2.13 `static void ata_mwdma (union param_val val, void * dat)` [static]

Set the ATA multi-word DMA mode

Must be -1, 0, 1 or 2.

Parameters:

← *val* The value to use

← *dat* The [config](#) data structure

6.101.2.14 `static void ata_packet (union param_val val, void * dat)` [static]

6.101.2.15 `static void ata_pio (union param_val val, void * dat)` [static]

Set the ATA programmed I/O mode

Must be 0, 1, 2, 3 or 4.

Parameters:

← *val* The value to use

← *dat* The [config](#) data structure

6.101.2.16 `static void ata_pio_mode0_t1 (union param_val val, void * dat)` [static]

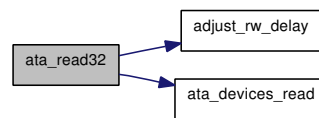
6.101.2.17 `static void ata_pio_mode0_t2 (union param_val val, void * dat)` [static]

6.101.2.18 `static void ata_pio_mode0_t4 (union param_val val, void * dat)` [static]

6.101.2.19 `static void ata_pio_mode0_tec (union param_val val, void * dat)` [static]

6.101.2.20 `static uint32_t ata_read32 (oraddr_t addr, void * dat)` [static]

Here is the call graph for this function:



6.101.2.21 `static void ata_reset (void * dat)` [static]

Here is the call graph for this function:



6.101.2.22 `static void ata_rev (union param_val val, void * dat)` [static]

Set the ATA revision

This must be in the range 0-15, to fit in the relevant field. Anything larger is truncated with a warning.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.101.2.25 `static void ata_sectors (union param_val val, void * dat)` [static]

6.101.2.26 `static void ata_size (union param_val val, void * dat)` [static]

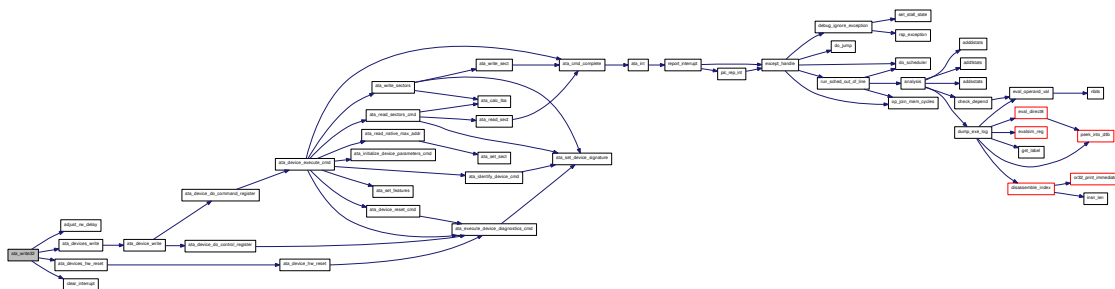
6.101.2.27 `static void ata_start_device (union param_val val, void * dat)` [static]

6.101.2.28 `static void ata_status (void * dat)` [static]

6.101.2.29 `static void ata_type (union param_val val, void * dat)` [static]

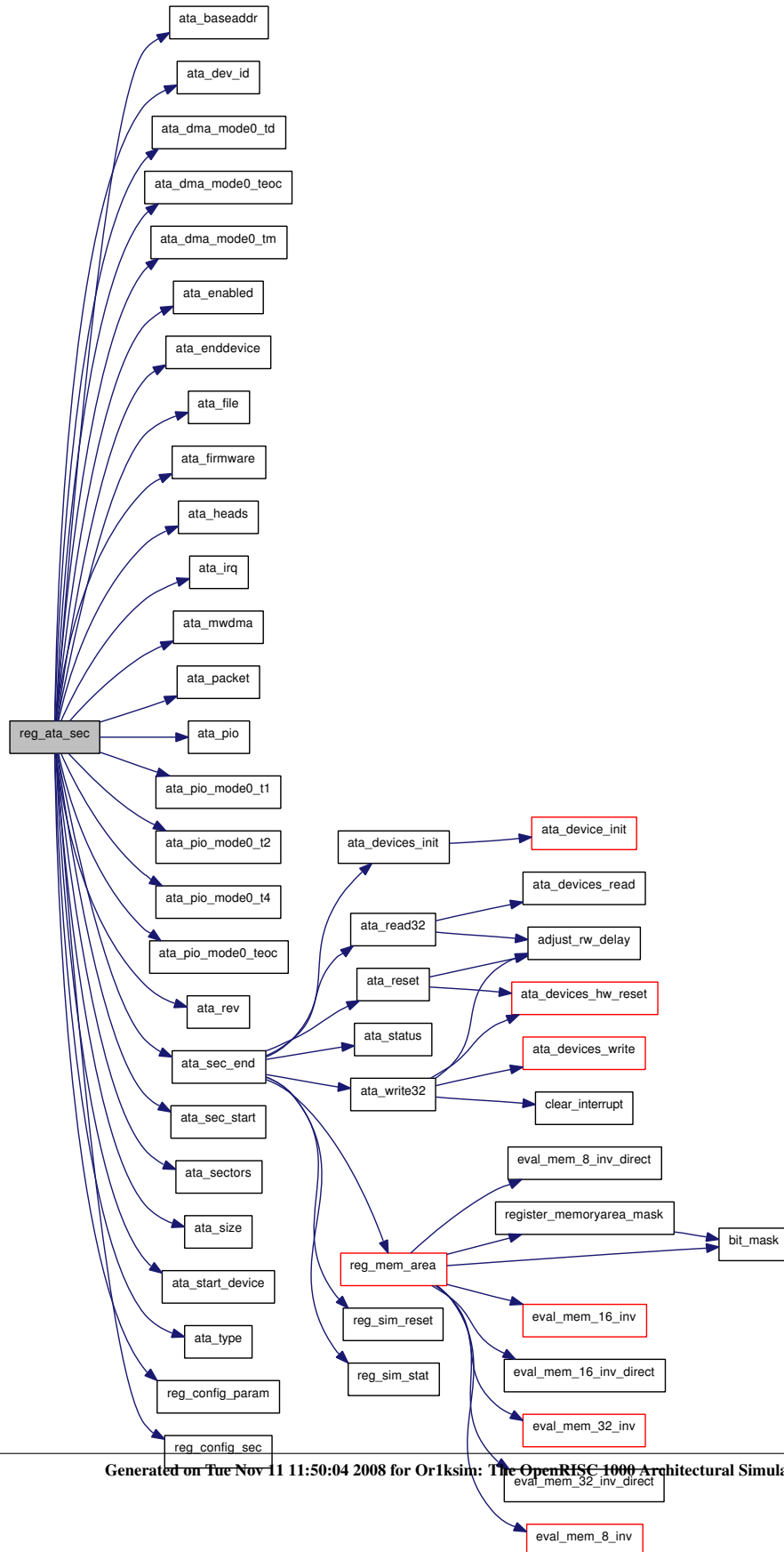
6.101.2.30 `static void ata_write32 (oraddr_t addr, uint32_t value, void * dat)` [static]

Here is the call graph for this function:



6.101.2.31 void reg_ata_sec ()

Here is the call graph for this function:



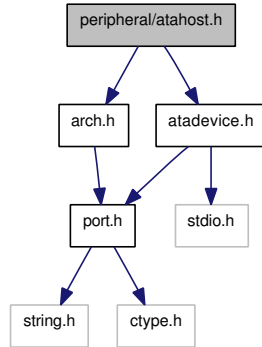
6.101.3 Variable Documentation

6.101.3.1 `unsigned int conf_dev` `[static]`

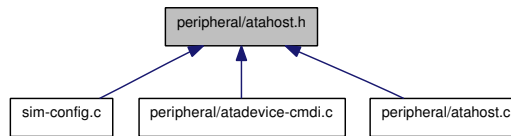
6.102 peripheral/atahost.h File Reference

```
#include "arch.h"
#include "atadevice.h"
```

Include dependency graph for atahost.h:



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



Data Structures

- struct [ata_host](#)

Defines

- #define [ATA_CTRL](#) 0x00
- #define [ATA_STAT](#) 0x04
- #define [ATA_PCTR](#) 0x08
- #define [ATA_PFTR0](#) 0x0c
- #define [ATA_PFTR1](#) 0x10
- #define [ATA_DTR0](#) 0x14
- #define [ATA_DTR1](#) 0x18
- #define [ATA_TXB](#) 0x3c
- #define [ATA_RXB](#) 0x3c
- #define [ATA_DMA_EN](#) (0<<15)
- #define [ATA_DMA_WR](#) (1<<14)
- #define [ATA_DMA_RD](#) (0<<14)
- #define [ATA_BELEC1](#) (1<<9)
- #define [ATA_BELECO](#) (1<<8)
- #define [ATA_IDE_EN](#) (1<<7)
- #define [ATA_FTE1](#) (1<<6)

- #define [ATA_FTE0](#) (1<< 5)
- #define [ATA_PWPP](#) (1<< 4)
- #define [ATA_IORDY_FTE1](#) (1<< 3)
- #define [ATA_IORDY_FTE0](#) (1<< 2)
- #define [ATA_IORDY](#) (1<< 1)
- #define [ATA_RST](#) (1<< 0)
- #define [ATA_DEVID](#) 0xf0000000
- #define [ATA_REVNO](#) 0x0f000000
- #define [ATA_DMA_TIP](#) (1<<15)
- #define [ATA_DRBE](#) (1<<10)
- #define [ATA_DTBF](#) (1<< 9)
- #define [ATA_DMARQ](#) (1<< 8)
- #define [ATA_PIO_TIP](#) (1<< 7)
- #define [ATA_PWPPF](#) (1<< 6)
- #define [ATA_IDEIS](#) (1<< 0)
- #define [ATA_TEOC](#) 24
- #define [ATA_T4](#) 16
- #define [ATA_T2](#) 8
- #define [ATA_TD](#) 8
- #define [ATA_T1](#) 0
- #define [ATA_TM](#) 0
- #define [ATA_ADDR_SPACE](#) 0x80
- #define [is_ata_hostadr](#)(adr) (!(adr & 0x40))
- #define [ata_pio_delay](#)(pioreg) ((((pioreg >> ATA_T1) & 0xff) +1) + (((pioreg >> ATA_T2) & 0xff) +1) + (((pioreg >> ATA_T4) & 0xff) +1) +1)
- #define [ata_dma_delay](#)(dmareg) ((((dmareg >> ATA_TD) & 0xff) +1) + (((pioreg >> ATA_TM) & 0xff) +1) +1)

Functions

- void [ata_int](#) (void *dat)
- void [reg_ata_sec](#) ()

6.102.1 Define Documentation

6.102.1.1 `#define ATA_ADDR_SPACE 0x80`

6.102.1.2 `#define ATA_BELEC0 (1<< 8)`

6.102.1.3 `#define ATA_BELEC1 (1<< 9)`

6.102.1.4 `#define ATA_CTRL 0x00`

6.102.1.5 `#define ATA_DEVID 0xf0000000`

6.102.1.6 `#define ata_dma_delay(dmareg) ((((dmareg >> ATA_TD) & 0xff) +1) + (((pioreg >> ATA_TM) & 0xff) +1) +1)`

6.102.1.7 `#define ATA_DMA_EN (0<<15)`

6.102.1.8 `#define ATA_DMA_RD (0<<14)`

6.102.1.9 `#define ATA_DMA_TIP (1<<15)`

6.102.1.10 `#define ATA_DMA_WR (1<<14)`

6.102.1.11 `#define ATA_DMARQ (1<< 8)`

6.102.1.12 `#define ATA_DRBE (1<<10)`

6.102.1.13 `#define ATA_DTBF (1<< 9)`

6.102.1.14 `#define ATA_DTR0 0x14`

6.102.1.15 `#define ATA_DTR1 0x18`

6.102.1.16 `#define ATA_FTE0 (1<< 5)`

6.102.1.17 `#define ATA_FTE1 (1<< 6)`

6.102.1.18 `#define ATA_IDE_EN (1<< 7)`

6.102.1.19 `#define ATA_IDEIS (1<< 0)`

6.102.1.20 `#define ATA_IORDY (1<< 1)`

6.102.1.21 `#define ATA_IORDY_FTE0 (1<< 2)`

6.102.1.22 `#define ATA_IORDY_FTE1 (1<< 3)`

6.102.1.23 `#define ATA_PCTR 0x08`

6.102.1.24 `#define ATA_PFTR0 0x0c`

6.102.1.25 `#define ATA_PFTR1 0x10`

6.102.1.26 `#define ata_pio_delay(pioreg) ((((pioreg >> ATA_T1) & 0xff) +1) + (((pioreg >> ATA_T2) & 0xff) +1) + (((pioreg >> ATA_T4) & 0xff) +1) +1)`

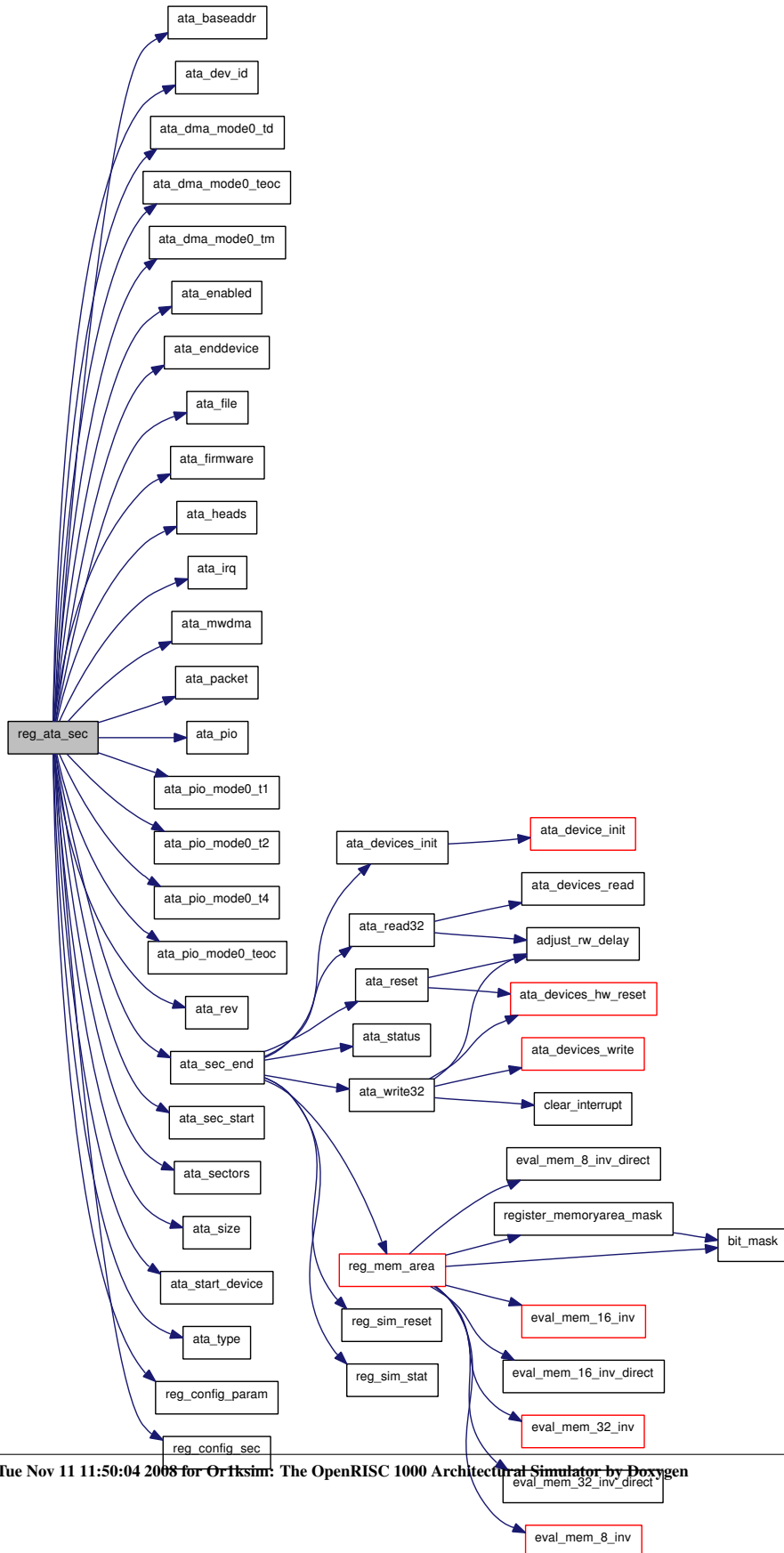
6.102.1.27 `#define ATA_PIO_TIP (1<< 7)`

6.102.1.28 `#define ATA_PWPP (1<< 4)`

6.102.1.29 `#define ATA_PWPPF (1<< 6)`

6.102.2.2 void reg_ata_sec ()

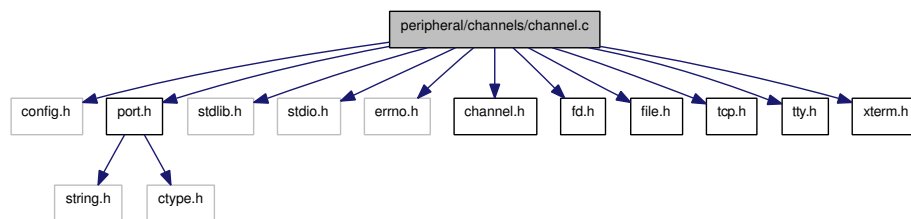
Here is the call graph for this function:



6.103 peripheral/channels/channel.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include "channel.h"
#include "fd.h"
#include "file.h"
#include "tcp.h"
#include "tty.h"
#include "xterm.h"
```

Include dependency graph for channel.c:



Data Structures

- struct [channel_factory](#)

Functions

- static struct [channel_factory](#) * [find_channel_factory](#) (const char *name)
- struct [channel](#) * [channel_init](#) (const char *descriptor)
- int [channel_open](#) (struct [channel](#) *channel)
- int [channel_read](#) (struct [channel](#) *channel, char *buffer, int size)
- int [channel_write](#) (struct [channel](#) *channel, const char *buffer, int size)
- void [channel_close](#) (struct [channel](#) *channel)

Variables

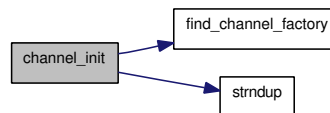
- static struct [channel_factory](#) [preloaded](#) []
- static struct [channel_factory](#) * [head](#) = &[preloaded](#)[0]

6.103.1 Function Documentation

6.103.1.1 void `channel_close` (`struct channel *channel`)

6.103.1.2 `struct channel*` `channel_init` (`const char *descriptor`) [read]

Here is the call graph for this function:



6.103.1.3 int `channel_open` (`struct channel *channel`)

6.103.1.4 int `channel_read` (`struct channel *channel`, `char *buffer`, `int size`)

6.103.1.5 int `channel_write` (`struct channel *channel`, `const char *buffer`, `int size`)

6.103.1.6 `static struct channel_factory *` `find_channel_factory` (`const char *name`) [static, read]

6.103.2 Variable Documentation

6.103.2.1 `struct channel_factory*` `head = &preloaded[0]` [static]

6.103.2.2 `struct channel_factory` `preloaded[]` [static]

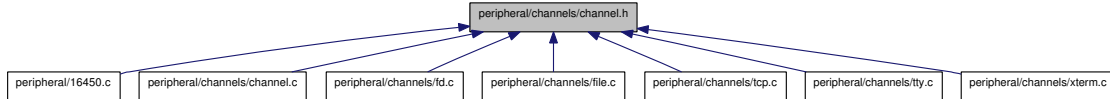
Initial value:

```

{
  {"fd",      &fd_channel_ops,   &preloaded[1]},
  {"file",   &file_channel_ops, &preloaded[2]},
  {"xterm",  &xterm_channel_ops, &preloaded[3]},
  {"tcp",    &tcp_channel_ops,   &preloaded[4]},
  {"tty",    &tty_channel_ops,   NULL}
}
  
```

6.104 peripheral/channels/channel.h File Reference

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



Data Structures

- struct [channel_ops](#)
- struct [channel](#)

Functions

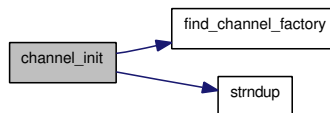
- struct [channel](#) * [channel_init](#) (const char *descriptor)
- int [channel_open](#) (struct [channel](#) *channel)
- int [channel_read](#) (struct [channel](#) *channel, char *buffer, int size)
- int [channel_write](#) (struct [channel](#) *channel, const char *buffer, int size)
- void [channel_close](#) (struct [channel](#) *channel)

6.104.1 Function Documentation

6.104.1.1 void [channel_close](#) (struct [channel](#) * *channel*)

6.104.1.2 struct [channel](#)* [channel_init](#) (const char * *descriptor*) [read]

Here is the call graph for this function:



6.104.1.3 int [channel_open](#) (struct [channel](#) * *channel*)

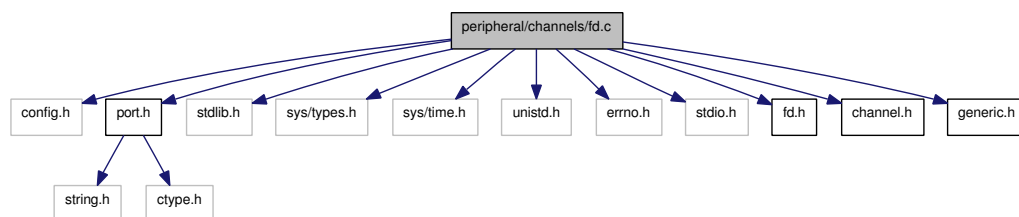
6.104.1.4 int [channel_read](#) (struct [channel](#) * *channel*, char * *buffer*, int *size*)

6.104.1.5 int [channel_write](#) (struct [channel](#) * *channel*, const char * *buffer*, int *size*)

6.105 peripheral/channels/fd.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>
#include <unistd.h>
#include <errno.h>
#include <stdio.h>
#include "fd.h"
#include "channel.h"
#include "generic.h"
```

Include dependency graph for fd.c:



Functions

- static void * [fd_init](#) (const char *args)
- static int [fd_isok](#) (void *data)
- static char * [fd_status](#) (void *data)
- int [fd_read](#) (void *data, char *buffer, int size)
- int [fd_write](#) (void *data, const char *buffer, int size)
- static int [fd_status_fd](#) (int fd, char *str, int size)

Variables

- struct [channel_ops](#) [fd_channel_ops](#)

6.105.1 Function Documentation

6.105.1.1 `static void * fd_init (const char * args)` [static]

6.105.1.2 `static int fd_isok (void * data)` [static]

6.105.1.3 `int fd_read (void * data, char * buffer, int size)`

6.105.1.4 `static char * fd_status (void * data)` [static]

Here is the call graph for this function:



6.105.1.5 `static int fd_status_fd (int fd, char * str, int size)` [static]

6.105.1.6 `int fd_write (void * data, const char * buffer, int size)`

6.105.2 Variable Documentation

6.105.2.1 `struct channel_ops fd_channel_ops`

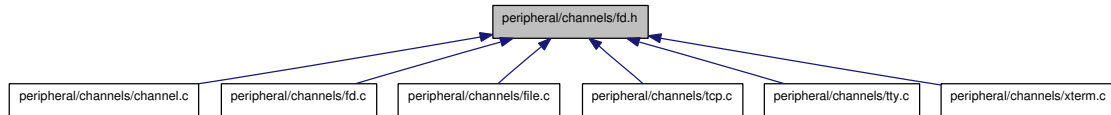
Initial value:

```
{
    .init    = fd_init,
    .open    = generic_open,
    .close   = generic_close,
    .read    = fd_read,
    .write   = fd_write,
    .free    = generic_free,
    .isok    = fd_isok,
    .status  = fd_status,
}
```

Global data structure representing the operations for communicating through a file descriptor [channel](#)

6.106 peripheral/channels/fd.h File Reference

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



Data Structures

- struct [fd_channel](#)

Functions

- int [fd_read](#) (void *data, char *buffer, int size)
- int [fd_write](#) (void *data, const char *buffer, int size)

Variables

- struct [channel_ops](#) [fd_channel_ops](#)

6.106.1 Function Documentation

6.106.1.1 int [fd_read](#) (void * *data*, char * *buffer*, int *size*)

6.106.1.2 int [fd_write](#) (void * *data*, const char * *buffer*, int *size*)

6.106.2 Variable Documentation

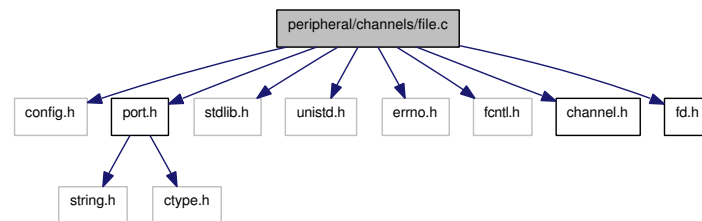
6.106.2.1 struct [channel_ops](#) [fd_channel_ops](#)

Global data structure representing the operations for communicating through a file descriptor [channel](#)

6.107 peripheral/channels/file.c File Reference

```
#include "config.h"  
#include "port.h"  
#include <stdlib.h>  
#include <unistd.h>  
#include <errno.h>  
#include <fcntl.h>  
#include "channel.h"  
#include "fd.h"
```

Include dependency graph for file.c:



Data Structures

- struct [file_channel](#)

Functions

- static void * [file_init](#) (const char *args)
- static int [file_open](#) (void *data)
- static void [file_close](#) (void *data)
- static void [file_free](#) (void *data)

Variables

- struct [channel_ops](#) [file_channel_ops](#)

6.107.1 Function Documentation

6.107.1.1 `static void file_close (void * data)` [static]

6.107.1.2 `static void file_free (void * data)` [static]

6.107.1.3 `static void * file_init (const char * args)` [static]

Here is the call graph for this function:



6.107.1.4 `static int file_open (void * data)` [static]

6.107.2 Variable Documentation

6.107.2.1 `struct channel_ops file_channel_ops`

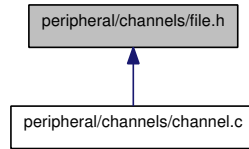
Initial value:

```
{
    .init = file_init,
    .open = file_open,
    .close = file_close,
    .read = fd_read,
    .write = fd_write,
    .free = file_free,
}
```

Data structure with all the operations for communicating with a file [channel](#)

6.108 peripheral/channels/file.h File Reference

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



Variables

- struct [channel_ops](#) [file_channel_ops](#)

6.108.1 Variable Documentation

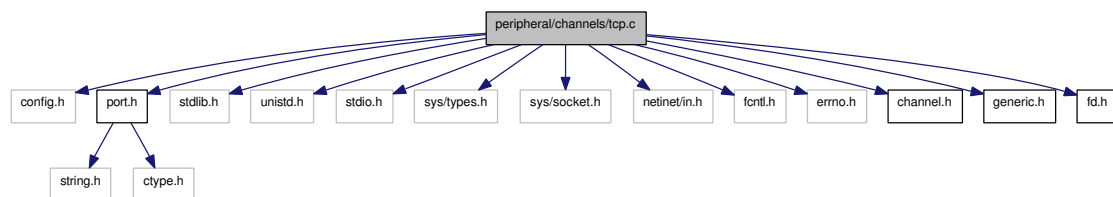
6.108.1.1 struct [channel_ops](#) [file_channel_ops](#)

Data structure with all the operations for communicating with a file [channel](#)

6.109 peripheral/channels/tcp.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <errno.h>
#include "channel.h"
#include "generic.h"
#include "fd.h"
```

Include dependency graph for tcp.c:



Data Structures

- struct [tcp_channel](#)

Functions

- static void * [tcp_init](#) (const char *input)
- static int [tcp_open](#) (void *data)
- static int [tcp_read](#) (void *data, char *buffer, int size)
- static int [tcp_write](#) (void *data, const char *buffer, int size)
- static int [wait_for_tcp_connect](#) (struct [tcp_channel](#) *channel)

Variables

- struct [channel_ops](#) [tcp_channel_ops](#)

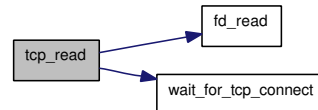
6.109.1 Function Documentation

6.109.1.1 `static void * tcp_init (const char * input)` [static]

6.109.1.2 `static int tcp_open (void * data)` [static]

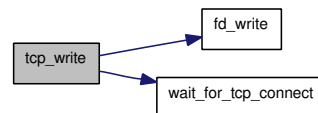
6.109.1.3 `static int tcp_read (void * data, char * buffer, int size)` [static]

Here is the call graph for this function:



6.109.1.4 `static int tcp_write (void * data, const char * buffer, int size)` [static]

Here is the call graph for this function:



6.109.1.5 `static int wait_for_tcp_connect (struct tcp_channel * channel)` [static]

6.109.2 Variable Documentation

6.109.2.1 `struct channel_ops tcp_channel_ops`

Initial value:

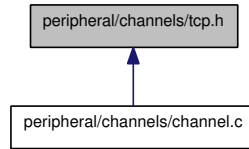
```

{
    .init = tcp_init,
    .open = tcp_open,
    .close = generic_close,
    .read = tcp_read,
    .write = tcp_write,
    .free = generic_free,
}
  
```

Data structure holding all the operations for a TCP/IP [channel](#)

6.110 peripheral/channels/tcp.h File Reference

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



Variables

- struct [channel_ops](#) [tcp_channel_ops](#)

6.110.1 Variable Documentation

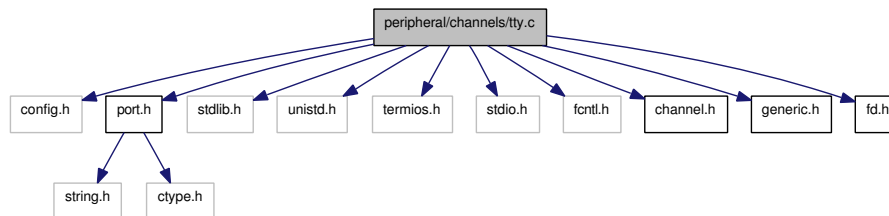
6.110.1.1 struct [channel_ops](#) [tcp_channel_ops](#)

Data structure holding all the operations for a TCP/IP [channel](#)

6.111 peripheral/channels/tty.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <unistd.h>
#include <termios.h>
#include <stdio.h>
#include <fcntl.h>
#include "channel.h"
#include "generic.h"
#include "fd.h"
```

Include dependency graph for tty.c:



Data Structures

- struct [tty_channel](#)

Defines

- #define [DEFAULT_BAUD](#) B19200
- #define [DEFAULT_TTY_DEVICE](#) "/dev/ttyS0"

Functions

- static void * [tty_init](#) (const char *input)
- static int [tty_open](#) (void *data)
- static int [parse_baud](#) (char *baud_string)

Variables

- struct {
 - char * [name](#)
 - int [value](#)
} [baud_table](#) []
- struct [channel_ops](#) [tty_channel_ops](#)

6.111.1 Define Documentation

6.111.1.1 `#define DEFAULT_BAUD B19200`

6.111.1.2 `#define DEFAULT_TTY_DEVICE "/dev/ttyS0"`

6.111.2 Function Documentation

6.111.2.1 `static int parse_baud (char * baud_string)` [static]

6.111.2.2 `static void * tty_init (const char * input)` [static]

Here is the call graph for this function:



6.111.2.3 `static int tty_open (void * data)` [static]

6.111.3 Variable Documentation

6.111.3.1 `struct { ... } baud_table[]` [static]

Table of Baud rates

6.111.3.2 `char* name`

6.111.3.3 `struct channel_ops tty_channel_ops`

Initial value:

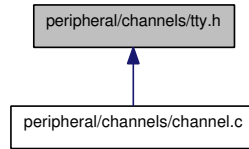
```
{
    .init = tty_init,
    .open = tty_open,
    .close = generic_close,
    .read = fd_read,
    .write = fd_write,
    .free = generic_free,
}
```

Global data structure representing the operations on a TTY [channel](#)

6.111.3.4 `int value`

6.112 peripheral/channels/tty.h File Reference

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



Variables

- struct [channel_ops](#) [tty_channel_ops](#)

6.112.1 Variable Documentation

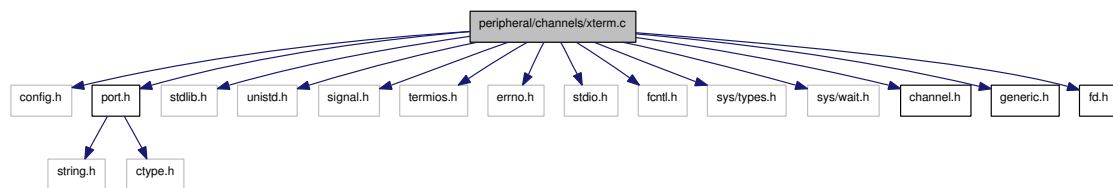
6.112.1.1 struct [channel_ops](#) [tty_channel_ops](#)

Global data structure representing the operations on a TTY [channel](#)

6.113 peripheral/channels/xterm.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include <termios.h>
#include <errno.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/wait.h>
#include "channel.h"
#include "generic.h"
#include "fd.h"
```

Include dependency graph for xterm.c:



Data Structures

- struct [xterm_channel](#)

Defines

- #define [MAX_XTERM_ARGS](#) 100

Functions

- static void [xterm_close](#) (void *data)
- static void * [xterm_init](#) (const char *input)
- static int [xterm_open](#) (void *data)
- static char * [basename](#) (const char *filename)

Variables

- struct [channel_ops](#) [xterm_channel_ops](#)

6.113.1 Define Documentation

6.113.1.1 `#define MAX_XTERM_ARGS 100`

6.113.2 Function Documentation

6.113.2.1 `static char* basename (const char *filename) [static]`

6.113.2.2 `static void xterm_close (void *data) [static]`

6.113.2.3 `static void * xterm_init (const char *input) [static]`

6.113.2.4 `static int xterm_open (void *data) [static]`

Here is the call graph for this function:



6.113.3 Variable Documentation

6.113.3.1 `struct channel_ops xterm_channel_ops`

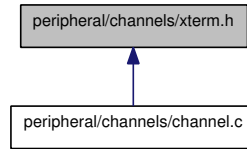
Initial value:

```
{
    .init = xterm_init,
    .open = xterm_open,
    .close = xterm_close,
    .read = fd_read,
    .write = fd_write,
    .free = generic_free,
}
```

Global data structure with the xterm interface functions

6.114 peripheral/channels/xterm.h File Reference

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



Variables

- struct [channel_ops](#) [xterm_channel_ops](#)

6.114.1 Variable Documentation

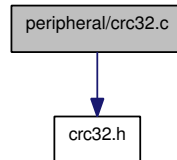
6.114.1.1 struct [channel_ops](#) [xterm_channel_ops](#)

Global data structure with the xterm interface functions

6.115 peripheral/crc32.c File Reference

```
#include "crc32.h"
```

Include dependency graph for crc32.c:



Functions

- unsigned long [crc32](#) (const void *buf, unsigned len)
- void [crc32_init](#) (unsigned long *value)
- void [crc32_feed_bytes](#) (unsigned long *value, const void *buf, unsigned len)
- void [crc32_close](#) (unsigned long *value)

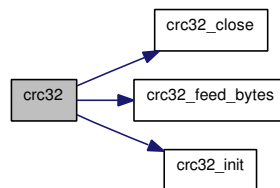
Variables

- static unsigned long [crc32_table](#) [256]

6.115.1 Function Documentation

6.115.1.1 unsigned long [crc32](#) (const void * buf, unsigned len)

Here is the call graph for this function:



6.115.1.2 void [crc32_close](#) (unsigned long * value)

6.115.1.3 void [crc32_feed_bytes](#) (unsigned long * value, const void * buf, unsigned len)

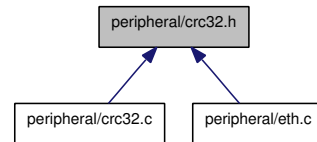
6.115.1.4 void [crc32_init](#) (unsigned long * value)

6.115.2 Variable Documentation

6.115.2.1 unsigned long [crc32_table](#)[256] [static]

6.116 peripheral/crc32.h File Reference

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



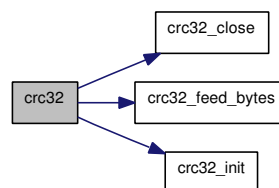
Functions

- unsigned long [crc32](#) (const void *buf, unsigned len)
- void [crc32_init](#) (unsigned long *value)
- void [crc32_feed_bytes](#) (unsigned long *value, const void *buf, unsigned len)
- void [crc32_close](#) (unsigned long *value)

6.116.1 Function Documentation

6.116.1.1 unsigned long [crc32](#) (const void * buf, unsigned len)

Here is the call graph for this function:



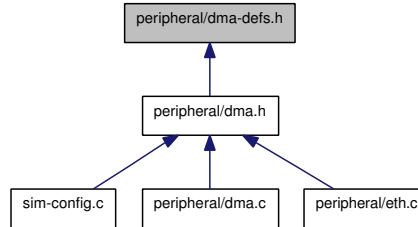
6.116.1.2 void [crc32_close](#) (unsigned long * value)

6.116.1.3 void [crc32_feed_bytes](#) (unsigned long * value, const void * buf, unsigned len)

6.116.1.4 void [crc32_init](#) (unsigned long * value)

6.117 peripheral/dma-defs.h File Reference

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



Defines

- #define [DMA_NUM_CHANNELS](#) 31
- #define [DMA_ADDR_SPACE](#) 0x400
- #define [DMA_CSR](#) 0x00
- #define [DMA_INT_MSK_A](#) 0x04
- #define [DMA_INT_MSK_B](#) 0x08
- #define [DMA_INT_SRC_A](#) 0x0C
- #define [DMA_INT_SRC_B](#) 0x10
- #define [DMA_CH_BASE](#) 0x20
- #define [DMA_CH_SIZE](#) 0x20
- #define [DMA_CH_CSR](#) 0x00
- #define [DMA_CH_SZ](#) 0x04
- #define [DMA_CH_A0](#) 0x08
- #define [DMA_CH_AM0](#) 0x0C
- #define [DMA_CH_A1](#) 0x10
- #define [DMA_CH_AM1](#) 0x14
- #define [DMA_CH_DESC](#) 0x18
- #define [DMA_CH_SWPTR](#) 0x1C
- #define [DMA_CSR_PAUSE_OFFSET](#) 0
- #define [DMA_CH_CSR_CH_EN_OFFSET](#) 0
- #define [DMA_CH_CSR_DST_SEL_OFFSET](#) 1
- #define [DMA_CH_CSR_SRC_SEL_OFFSET](#) 2
- #define [DMA_CH_CSR_INC_DST_OFFSET](#) 3
- #define [DMA_CH_CSR_INC_SRC_OFFSET](#) 4
- #define [DMA_CH_CSR_MODE_OFFSET](#) 5
- #define [DMA_CH_CSR_ARS_OFFSET](#) 6
- #define [DMA_CH_CSR_USE_ED_OFFSET](#) 7
- #define [DMA_CH_CSR_SZ_WB_OFFSET](#) 8
- #define [DMA_CH_CSR_STOP_OFFSET](#) 9
- #define [DMA_CH_CSR_BUSY_OFFSET](#) 10
- #define [DMA_CH_CSR_DONE_OFFSET](#) 11
- #define [DMA_CH_CSR_ERR_OFFSET](#) 12
- #define [DMA_CH_CSR_PRIORITY_OFFSET](#) 13
- #define [DMA_CH_CSR_PRIORITY_WIDTH](#) 3
- #define [DMA_CH_CSR_REST_EN_OFFSET](#) 16
- #define [DMA_CH_CSR_INE_ERR_OFFSET](#) 17

- #define DMA_CH_CSR_INE_DONE_OFFSET 18
- #define DMA_CH_CSR_INE_CHK_DONE_OFFSET 19
- #define DMA_CH_CSR_INT_ERR_OFFSET 20
- #define DMA_CH_CSR_INT_DONE_OFFSET 21
- #define DMA_CH_CSR_INT_CHUNK_DONE_OFFSET 22
- #define DMA_CH_CSR_RESERVED_OFFSET 23
- #define DMA_CH_CSR_RESERVED_WIDTH 9
- #define DMA_CH_CSR_WRITE_MASK 0x000FE3FF
- #define DMA_CH_SZ_TOT_SZ_OFFSET 0
- #define DMA_CH_SZ_TOT_SZ_WIDTH 12
- #define DMA_CH_SZ_CHK_SZ_OFFSET 16
- #define DMA_CH_SZ_CHK_SZ_WIDTH 9
- #define DMA_CH_A0_ADDR_OFFSET 2
- #define DMA_CH_A0_ADDR_WIDTH 30
- #define DMA_CH_A1_ADDR_OFFSET 2
- #define DMA_CH_A1_ADDR_WIDTH 30
- #define DMA_CH_AM0_MASK_OFFSET 4
- #define DMA_CH_AM0_MASK_WIDTH 28
- #define DMA_CH_AM1_MASK_OFFSET 4
- #define DMA_CH_AM1_MASK_WIDTH 28
- #define DMA_CH_DESC_ADDR_OFFSET 2
- #define DMA_CH_DESC_ADDR_WIDTH 30
- #define DMA_CH_SWPTR_PTR_OFFSET 2
- #define DMA_CH_SWPTR_PTR_WIDTH 29
- #define DMA_CH_SWPTR_EN_OFFSET 31
- #define DMA_DESC_CSR 0x00
- #define DMA_DESC_ADR0 0x04
- #define DMA_DESC_ADR1 0x08
- #define DMA_DESC_NEXT 0x0C
- #define DMA_DESC_CSR_EOL_OFFSET 20
- #define DMA_DESC_CSR_INC_SRC_OFFSET 19
- #define DMA_DESC_CSR_INC_DST_OFFSET 18
- #define DMA_DESC_CSR_SRC_SEL_OFFSET 17
- #define DMA_DESC_CSR_DST_SEL_OFFSET 16
- #define DMA_DESC_CSR_TOT_SZ_OFFSET 0
- #define DMA_DESC_CSR_TOT_SZ_WIDTH 12

6.117.1 Define Documentation

6.117.1.1 `#define DMA_ADDR_SPACE 0x400`

6.117.1.2 `#define DMA_CH_A0 0x08`

6.117.1.3 `#define DMA_CH_A0_ADDR_OFFSET 2`

6.117.1.4 `#define DMA_CH_A0_ADDR_WIDTH 30`

6.117.1.5 `#define DMA_CH_A1 0x10`

6.117.1.6 `#define DMA_CH_A1_ADDR_OFFSET 2`

6.117.1.7 `#define DMA_CH_A1_ADDR_WIDTH 30`

6.117.1.8 `#define DMA_CH_AM0 0x0C`

6.117.1.9 `#define DMA_CH_AM0_MASK_OFFSET 4`

6.117.1.10 `#define DMA_CH_AM0_MASK_WIDTH 28`

6.117.1.11 `#define DMA_CH_AM1 0x14`

6.117.1.12 `#define DMA_CH_AM1_MASK_OFFSET 4`

6.117.1.13 `#define DMA_CH_AM1_MASK_WIDTH 28`

6.117.1.14 `#define DMA_CH_BASE 0x20`

Offset of first [channel](#) registers


```

6.117.1.15 #define DMA_CH_CSR 0x00
6.117.1.16 #define DMA_CH_CSR_ARS_OFFSET 6
6.117.1.17 #define DMA_CH_CSR_BUSY_OFFSET 10
6.117.1.18 #define DMA_CH_CSR_CH_EN_OFFSET 0
6.117.1.19 #define DMA_CH_CSR_DONE_OFFSET 11
6.117.1.20 #define DMA_CH_CSR_DST_SEL_OFFSET 1
6.117.1.21 #define DMA_CH_CSR_ERR_OFFSET 12
6.117.1.22 #define DMA_CH_CSR_INC_DST_OFFSET 3
6.117.1.23 #define DMA_CH_CSR_INC_SRC_OFFSET 4
6.117.1.24 #define DMA_CH_CSR_INE_CHK_DONE_OFFSET 19
6.117.1.25 #define DMA_CH_CSR_INE_DONE_OFFSET 18
6.117.1.26 #define DMA_CH_CSR_INE_ERR_OFFSET 17
6.117.1.27 #define DMA_CH_CSR_INT_CHUNK_DONE_OFFSET 22
6.117.1.28 #define DMA_CH_CSR_INT_DONE_OFFSET 21
6.117.1.29 #define DMA_CH_CSR_INT_ERR_OFFSET 20
6.117.1.30 #define DMA_CH_CSR_MODE_OFFSET 5
6.117.1.31 #define DMA_CH_CSR_PRIORITY_OFFSET 13
6.117.1.32 #define DMA_CH_CSR_PRIORITY_WIDTH 3
6.117.1.33 #define DMA_CH_CSR_RESERVED_OFFSET 23
6.117.1.34 #define DMA_CH_CSR_RESERVED_WIDTH 9
6.117.1.35 #define DMA_CH_CSR_REST_EN_OFFSET 16
6.117.1.36 #define DMA_CH_CSR_SRC_SEL_OFFSET 2
6.117.1.37 #define DMA_CH_CSR_STOP_OFFSET 9
6.117.1.38 #define DMA_CH_CSR_SZ_WB_OFFSET 8
6.117.1.39 #define DMA_CH_CSR_USE_ED_OFFSET 7
6.117.1.40 #define DMA_CH_CSR_WRITE_MASK 0x000FE3FF
6.117.1.41 #define DMA_CH_DESC 0x18
6.117.1.42 #define DMA_CH_DESC_ADDR_OFFSET 2
6.117.1.43 #define DMA_CH_DESC_ADDR_WIDTH 30
6.117.1.44 #define DMA_CH_SIZE 0x20

```

6.117.1.45 #define DMA_CH_SWPTR 0x1C

6.117.1.46 #define DMA_CH_SWPTR_EN_OFFSET 31

6.117.1.47 #define DMA_CH_SWPTR_PTR_OFFSET 2

6.117.1.48 #define DMA_CH_SWPTR_PTR_WIDTH 29

6.117.1.49 #define DMA_CH_SZ 0x04

6.117.1.50 #define DMA_CH_SZ_CHK_SZ_OFFSET 16

6.117.1.51 #define DMA_CH_SZ_CHK_SZ_WIDTH 9

6.117.1.52 #define DMA_CH_SZ_TOT_SZ_OFFSET 0

6.117.1.53 #define DMA_CH_SZ_TOT_SZ_WIDTH 12

6.117.1.54 #define DMA_CSR 0x00

6.117.1.55 #define DMA_CSR_PAUSE_OFFSET 0

6.117.1.56 #define DMA_DESC_ADR0 0x04

6.117.1.57 #define DMA_DESC_ADR1 0x08

6.117.1.58 #define DMA_DESC_CSR 0x00

6.117.1.59 #define DMA_DESC_CSR_DST_SEL_OFFSET 16

6.117.1.60 #define DMA_DESC_CSR_EOL_OFFSET 20

6.117.1.61 #define DMA_DESC_CSR_INC_DST_OFFSET 18

6.117.1.62 #define DMA_DESC_CSR_INC_SRC_OFFSET 19

6.117.1.63 #define DMA_DESC_CSR_SRC_SEL_OFFSET 17

6.117.1.64 #define DMA_DESC_CSR_TOT_SZ_OFFSET 0

6.117.1.65 #define DMA_DESC_CSR_TOT_SZ_WIDTH 12

6.117.1.66 #define DMA_DESC_NEXT 0x0C

6.117.1.67 #define DMA_INT_MSK_A 0x04

6.117.1.68 #define DMA_INT_MSK_B 0x08

6.117.1.69 #define DMA_INT_SRC_A 0x0C

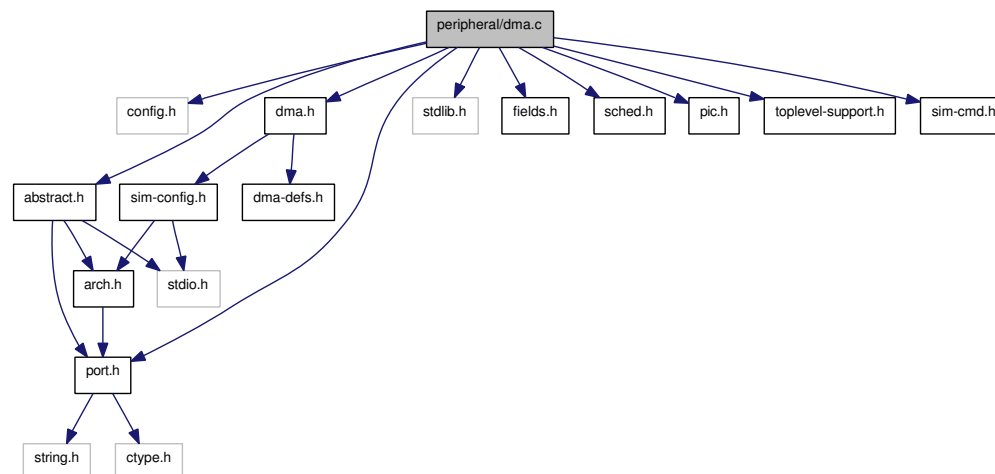
6.117.1.70 #define DMA_INT_SRC_B 0x10

6.117.1.71 #define DMA_NUM_CHANNELS 31

6.118 peripheral/dma.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "dma.h"
#include "fields.h"
#include "abstract.h"
#include "sched.h"
#include "pic.h"
#include "toplevel-support.h"
#include "sim-cmd.h"
```

Include dependency graph for dma.c:



Defines

- #define [CHANNEL_ND_I\(ch\)](#) (TEST_FLAG(ch → regs.csr,DMA_CH_CSR,MODE) && TEST_FLAG(ch → regs.csr,DMA_CH_CSR,USE_ED) && ch → dma_nd_i)

Functions

- static unsigned long [dma_read_ch_csr](#) (struct [dma_channel](#) *channel)
- static void [dma_write_ch_csr](#) (struct [dma_channel](#) *channel, unsigned long value)
- static void [dma_load_descriptor](#) (struct [dma_channel](#) *channel)
- static void [dma_init_transfer](#) (struct [dma_channel](#) *channel)
- static void [dma_channel_terminate_transfer](#) (struct [dma_channel](#) *channel, int generate_interrupt)
- static void [dma_channel_clock](#) (void *dat)
- static void [masked_increase](#) (oraddr_t *value, unsigned long mask)
- static void [dma_reset](#) (void *dat)

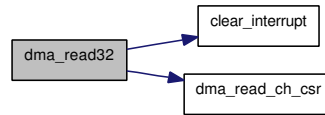
- static void `dma_status` (void *dat)
- static uint32_t `dma_read32` (oraddr_t addr, void *dat)
- static void `dma_write32` (oraddr_t addr, uint32_t value, void *dat)
- void `set_dma_req_i` (struct dma_channel *channel)
- void `clear_dma_req_i` (struct dma_channel *channel)
- void `set_dma_nd_i` (struct dma_channel *channel)
- void `clear_dma_nd_i` (struct dma_channel *channel)
- unsigned `check_dma_ack_o` (struct dma_channel *channel)
- struct dma_channel * `find_dma_controller_ch` (unsigned controller, unsigned channel)
- static void `dma_baseaddr` (union param_val val, void *dat)
- static void `dma_irq` (union param_val val, void *dat)
- static void `dma_vapi_id` (union param_val val, void *dat)
- static void `dma_enabled` (union param_val val, void *dat)
- static void * `dma_sec_start` ()
- static void `dma_sec_end` (void *dat)
- void `reg_dma_sec` (void)

Variables

- static struct `dma_controller` * `dmass` = NULL

6.118.2.11 static uint32_t dma_read32 (oraddr_t addr, void * dat) [static]

Here is the call graph for this function:

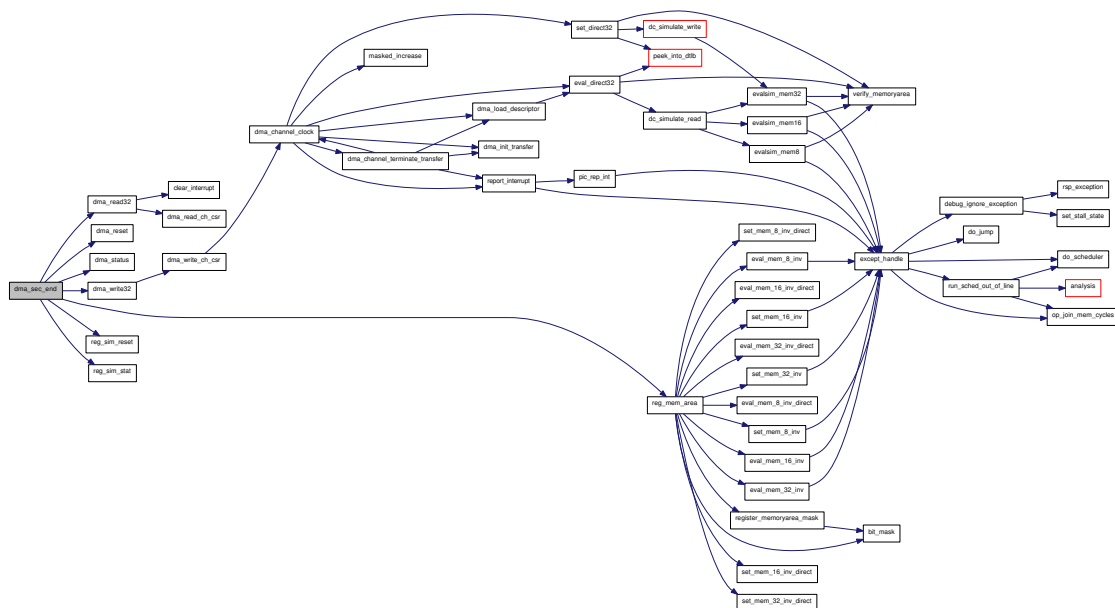


6.118.2.12 static unsigned long dma_read_ch_csr (struct dma_channel * channel) [static]

6.118.2.13 static void dma_reset (void * dat) [static]

6.118.2.14 static void dma_sec_end (void * dat) [static]

Here is the call graph for this function:



6.118.2.15 static void* dma_sec_start () [static]

Initialize a new DMA configuration

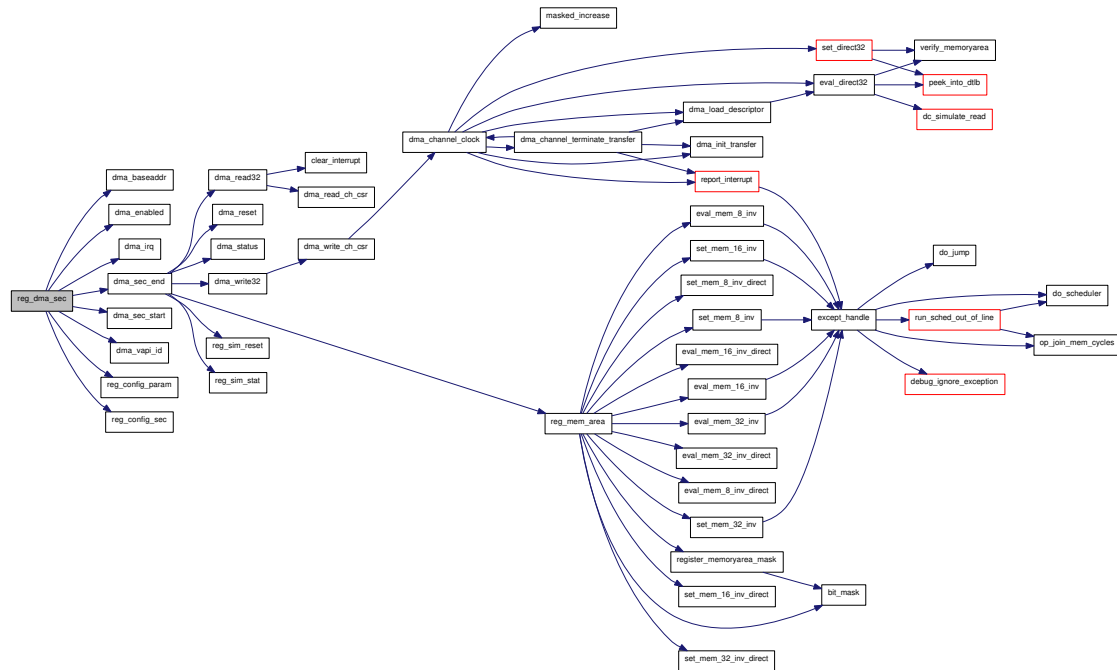
ALL parameters are set explicitly to default values.

6.118.2.20 `struct dma_channel* find_dma_controller_ch (unsigned controller, unsigned channel)`
[read]

6.118.2.21 `static void masked_increase (oraddr_t * value, unsigned long mask)` [static]

6.118.2.22 `void reg_dma_sec (void)`

Here is the call graph for this function:



6.118.2.23 `void set_dma_nd_i (struct dma_channel * channel)`

6.118.2.24 `void set_dma_req_i (struct dma_channel * channel)`

6.118.3 Variable Documentation

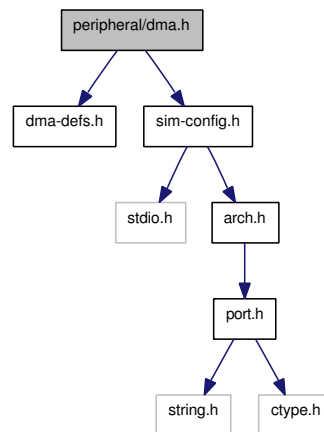
6.118.3.1 `struct dma_controller* dmas = NULL` [static]

6.119 peripheral/dma.h File Reference

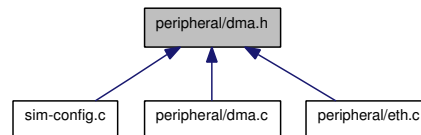
```
#include "dma-defs.h"
```

```
#include "sim-config.h"
```

Include dependency graph for dma.h:



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



Data Structures

- struct [dma_channel](#)
- struct [dma_controller](#)

Functions

- void [set_dma_req_i](#) (struct [dma_channel](#) *channel)
- void [clear_dma_req_i](#) (struct [dma_channel](#) *channel)
- void [set_dma_nd_i](#) (struct [dma_channel](#) *channel)
- void [clear_dma_nd_i](#) (struct [dma_channel](#) *channel)
- unsigned [check_dma_ack_o](#) (struct [dma_channel](#) *channel)
- struct [dma_channel](#) * [find_dma_controller_ch](#) (unsigned controller, unsigned channel)
- void [reg_dma_sec](#) ()

6.119.1 Function Documentation

6.119.1.1 unsigned `check_dma_ack_o` (struct `dma_channel` * *channel*)

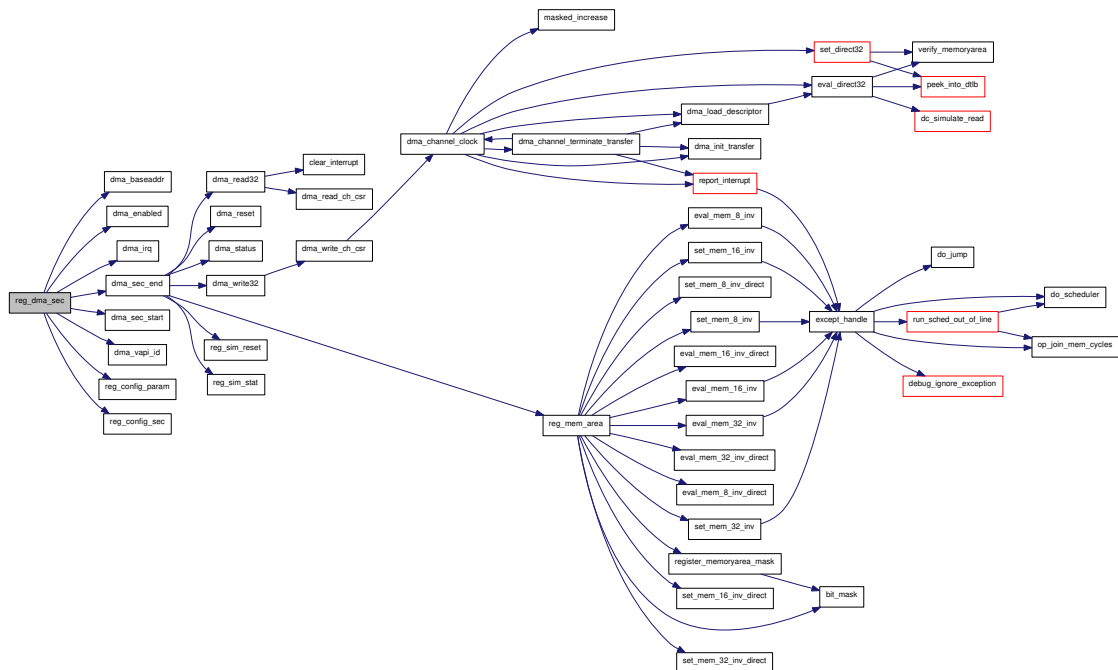
6.119.1.2 void `clear_dma_nd_i` (struct `dma_channel` * *channel*)

6.119.1.3 void `clear_dma_req_i` (struct `dma_channel` * *channel*)

6.119.1.4 struct `dma_channel`* `find_dma_controller_ch` (unsigned *controller*, unsigned *channel*)
[read]

6.119.1.5 void `reg_dma_sec` ()

Here is the call graph for this function:



6.119.1.6 void `set_dma_nd_i` (struct `dma_channel` * *channel*)

6.119.1.7 void `set_dma_req_i` (struct `dma_channel` * *channel*)

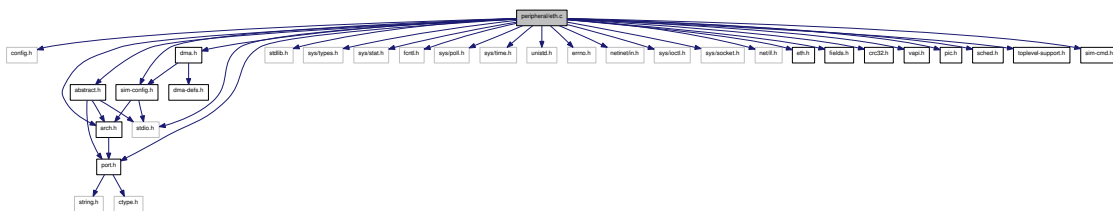
6.120 peripheral/eth.c File Reference

```

#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/poll.h>
#include <sys/time.h>
#include <unistd.h>
#include <errno.h>
#include <netinet/in.h>
#include <sys/ioctl.h>
#include <sys/socket.h>
#include <net/if.h>
#include "arch.h"
#include "abstract.h"
#include "eth.h"
#include "dma.h"
#include "sim-config.h"
#include "fields.h"
#include "crc32.h"
#include "vapi.h"
#include "pic.h"
#include "sched.h"
#include "toplevel-support.h"
#include "sim-cmd.h"

```

Include dependency graph for eth.c:



Data Structures

- struct [ether_addr](#)
- struct [ether_header](#)
- struct [eth_device](#)

Defines

- #define [ETH_ADDR_SPACE](#) 0x1000
- #define [ETH_MODER](#) (4 * 0x00)
- #define [ETH_INT_SOURCE](#) (4 * 0x01)
- #define [ETH_INT_MASK](#) (4 * 0x02)
- #define [ETH_IPGT](#) (4 * 0x03)
- #define [ETH_IPGR1](#) (4 * 0x04)
- #define [ETH_IPGR2](#) (4 * 0x05)
- #define [ETH_PACKETLEN](#) (4 * 0x06)
- #define [ETH_COLLCONF](#) (4 * 0x07)
- #define [ETH_TX_BD_NUM](#) (4 * 0x08)
- #define [ETH_CTRLMODER](#) (4 * 0x09)
- #define [ETH_MIIMODER](#) (4 * 0x0A)
- #define [ETH_MIICOMMAND](#) (4 * 0x0B)
- #define [ETH_MIIADDRESS](#) (4 * 0x0C)
- #define [ETH_MIITX_DATA](#) (4 * 0x0D)
- #define [ETH_MIIRX_DATA](#) (4 * 0x0E)
- #define [ETH_MIISTATUS](#) (4 * 0x0F)
- #define [ETH_MAC_ADDR0](#) (4 * 0x10)
- #define [ETH_MAC_ADDR1](#) (4 * 0x11)
- #define [ETH_HASH0](#) (4 * 0x12)
- #define [ETH_HASH1](#) (4 * 0x13)
- #define [ETH_BD_BASE](#) 0x400
- #define [ETH_BD_COUNT](#) 0x100
- #define [ETH_BD_SPACE](#) (4 * ETH_BD_COUNT)
- #define [ETH_DMA_RX_TX](#) 0x800
- #define [ETH_MODER_DMAEN_OFFSET](#) 17
- #define [ETH_MODER_RECSCALL_OFFSET](#) 16
- #define [ETH_MODER_PAD_OFFSET](#) 15
- #define [ETH_MODER_HUGEN_OFFSET](#) 14
- #define [ETH_MODER_CRCEN_OFFSET](#) 13
- #define [ETH_MODER_DLYCRCEN_OFFSET](#) 12
- #define [ETH_MODER_RST_OFFSET](#) 11
- #define [ETH_MODER_FULLD_OFFSET](#) 10
- #define [ETH_MODER_EXDFREN_OFFSET](#) 9
- #define [ETH_MODER_NOBCKOF_OFFSET](#) 8
- #define [ETH_MODER_LOOPBCK_OFFSET](#) 7
- #define [ETH_MODER_IFG_OFFSET](#) 6
- #define [ETH_MODER_PRO_OFFSET](#) 5
- #define [ETH_MODER_IAM_OFFSET](#) 4
- #define [ETH_MODER_BRO_OFFSET](#) 3
- #define [ETH_MODER_NOPRE_OFFSET](#) 2
- #define [ETH_MODER_TXEN_OFFSET](#) 1

- #define ETH_MODER_RXEN_OFFSET 0
- #define ETH_INT_SOURCE_RXC_OFFSET 6
- #define ETH_INT_SOURCE_TXC_OFFSET 5
- #define ETH_INT_SOURCE_BUSY_OFFSET 4
- #define ETH_INT_SOURCE_RXE_OFFSET 3
- #define ETH_INT_SOURCE_RXB_OFFSET 2
- #define ETH_INT_SOURCE_TXE_OFFSET 1
- #define ETH_INT_SOURCE_TXB_OFFSET 0
- #define ETH_INT_MASK_RXC_M_OFFSET 6
- #define ETH_INT_MASK_TXC_M_OFFSET 5
- #define ETH_INT_MASK_BUSY_M_OFFSET 4
- #define ETH_INT_MASK_RXE_M_OFFSET 3
- #define ETH_INT_MASK_RXB_M_OFFSET 2
- #define ETH_INT_MASK_TXE_M_OFFSET 1
- #define ETH_INT_MASK_TXB_M_OFFSET 0
- #define ETH_PACKETLEN_MINFL_OFFSET 16
- #define ETH_PACKETLEN_MINFL_WIDTH 16
- #define ETH_PACKETLEN_MAXFL_OFFSET 0
- #define ETH_PACKETLEN_MAXFL_WIDTH 16
- #define ETH_COLLCONF_MAXRET_OFFSET 16
- #define ETH_COLLCONF_MAXRET_WIDTH 4
- #define ETH_COLLCONF_COLLVALID_OFFSET 0
- #define ETH_COLLCONF_COLLVALID_WIDTH 6
- #define ETH_CMODER_TXFLOW_OFFSET 2
- #define ETH_CMODER_RXFLOW_OFFSET 1
- #define ETH_CMODER_PASSALL_OFFSET 0
- #define ETH_MIIMODER_MRST_OFFSET 9
- #define ETH_MIIMODER_NOPRE_OFFSET 8
- #define ETH_MIIMODER_CLKDIV_OFFSET 0
- #define ETH_MIIMODER_CLKDIV_WIDTH 8
- #define ETH_MIICOMM_WCDATA_OFFSET 2
- #define ETH_MIICOMM_RSTAT_OFFSET 1
- #define ETH_MIICOMM_SCANS_OFFSET 0
- #define ETH_MIIADDR_RGAD_OFFSET 8
- #define ETH_MIIADDR_RGAD_WIDTH 5
- #define ETH_MIIADDR_FIAD_OFFSET 0
- #define ETH_MIIADDR_FIAD_WIDTH 5
- #define ETH_MIISTAT_NVALID_OFFSET 1
- #define ETH_MIISTAT_BUSY_OFFSET 1
- #define ETH_MIISTAT_FAIL_OFFSET 0
- #define ETH_TX_BD_LENGTH_OFFSET 16
- #define ETH_TX_BD_LENGTH_WIDTH 16
- #define ETH_TX_BD_READY_OFFSET 15
- #define ETH_TX_BD_IRQ_OFFSET 14
- #define ETH_TX_BD_WRAP_OFFSET 13
- #define ETH_TX_BD_PAD_OFFSET 12
- #define ETH_TX_BD_CRC_OFFSET 11
- #define ETH_TX_BD_LAST_OFFSET 10
- #define ETH_TX_BD_PAUSE_OFFSET 9
- #define ETH_TX_BD_UNDERRUN_OFFSET 8

- #define ETH_TX_BD_RETRY_OFFSET 4
- #define ETH_TX_BD_RETRY_WIDTH 4
- #define ETH_TX_BD_RETRANSMIT_OFFSET 3
- #define ETH_TX_BD_COLLISION_OFFSET 2
- #define ETH_TX_BD_DEFER_OFFSET 1
- #define ETH_TX_BD_NO_CARRIER_OFFSET 0
- #define ETH_RX_BD_LENGTH_OFFSET 16
- #define ETH_RX_BD_LENGTH_WIDTH 16
- #define ETH_RX_BD_READY_OFFSET 15
- #define ETH_RX_BD_IRQ_OFFSET 14
- #define ETH_RX_BD_WRAP_OFFSET 13
- #define ETH_RX_BD_MISS_OFFSET 7
- #define ETH_RX_BD_UVERRUN_OFFSET 6
- #define ETH_RX_BD_INVALID_OFFSET 5
- #define ETH_RX_BD_DRIBBLE_OFFSET 4
- #define ETH_RX_BD_TOOBIG_OFFSET 3
- #define ETH_RX_BD_TOOSHORT_OFFSET 2
- #define ETH_RX_BD_CRC_OFFSET 1
- #define ETH_RX_BD_COLLISION_OFFSET 0
- #define ETH_ALEN 6
- #define ETHERTYPE_PUP 0x0200
- #define ETHERTYPE_IP 0x0800
- #define ETHERTYPE_ARP 0x0806
- #define ETHERTYPE_REVARP 0x8035
- #define ETHER_ADDR_LEN ETH_ALEN
- #define ETHER_TYPE_LEN 2
- #define ETHER_CRC_LEN 4
- #define ETHER_HDR_LEN ETH_HLEN
- #define ETHER_MIN_LEN (ETH_ZLEN + ETHER_CRC_LEN)
- #define ETHER_MAX_LEN (ETH_FRAME_LEN + ETHER_CRC_LEN)
- #define ETHER_IS_VALID_LEN(foo) ((foo) >= ETHER_MIN_LEN && (foo) <= ETHER_MAX_LEN)
- #define ETHERTYPE_TRAIL 0x1000
- #define ETHERTYPE_NTRAILER 16
- #define ETHERMTU ETH_DATA_LEN
- #define ETHERMIN (ETHER_MIN_LEN-ETHER_HDR_LEN-ETHER_CRC_LEN)
- #define ETH_TXSTATE_IDLE 0
- #define ETH_TXSTATE_WAIT4BD 10
- #define ETH_TXSTATE_READFIFO 20
- #define ETH_TXSTATE_TRANSMIT 30
- #define ETH_RXSTATE_IDLE 0
- #define ETH_RXSTATE_WAIT4BD 10
- #define ETH_RXSTATE_RECV 20
- #define ETH_RXSTATE_WRITEFIFO 30
- #define ETH_RTX_FILE 0
- #define ETH_RTX_SOCKET 1
- #define ETH_RTX_VAPI 2
- #define ETH_MAXPL 0x10000

Enumerations

- enum { [ETH_VAPI_DATA](#) = 0, [ETH_VAPI_CTRL](#), [ETH_NUM_VAPI_IDS](#) }

Functions

- static void [eth_vapi_read](#) (unsigned long id, unsigned long data, void *dat)
- static void [eth_write32](#) ([oraddr_t](#) addr, [uint32_t](#) value, void *dat)
- static [uint32_t](#) [eth_read32](#) ([oraddr_t](#) addr, void *dat)
- static void [eth_controller_tx_clock](#) (void *)
- static void [eth_controller_rx_clock](#) (void *)
- static [ssize_t](#) [eth_read_rx_file](#) (struct [eth_device](#) *, void *, [size_t](#))
- static void [eth_skip_rx_file](#) (struct [eth_device](#) *, [off_t](#))
- static void [eth_rx_next_packet](#) (struct [eth_device](#) *)
- static void [eth_write_tx_bd_num](#) (struct [eth_device](#) *, unsigned long value)
- static void [eth_reset](#) (void *dat)
- static void [eth_status](#) (void *dat)
- static void [eth_enabled](#) (union [param_val](#) val, void *dat)
- static void [eth_baseaddr](#) (union [param_val](#) val, void *dat)
- static void [eth_dma](#) (union [param_val](#) val, void *dat)
- static void [eth_irq](#) (union [param_val](#) val, void *dat)
- static void [eth_rtx_type](#) (union [param_val](#) val, void *dat)
- static void [eth_rx_channel](#) (union [param_val](#) val, void *dat)
- static void [eth_tx_channel](#) (union [param_val](#) val, void *dat)
- static void [eth_rxfile](#) (union [param_val](#) val, void *dat)
- static void [eth_txfile](#) (union [param_val](#) val, void *dat)
- static void [eth_sockif](#) (union [param_val](#) val, void *dat)
- static void [eth_vapi_id](#) (union [param_val](#) val, void *dat)
- static void * [eth_sec_start](#) (void)
- static void [eth_sec_end](#) (void *dat)
- void [reg_ethernet_sec](#) ()

6.120.1 Define Documentation

6.120.1.1 **#define ETH_ADDR_SPACE 0x1000**

6.120.1.2 **#define ETH_ALEN 6**

6.120.1.3 **#define ETH_BD_BASE 0x400**

6.120.1.4 **#define ETH_BD_COUNT 0x100**

6.120.1.5 **#define ETH_BD_SPACE (4 * ETH_BD_COUNT)**

6.120.1.6 **#define ETH_CMODER_PASSALL_OFFSET 0**

6.120.1.7 **#define ETH_CMODER_RXFLOW_OFFSET 1**

6.120.1.8 **#define ETH_CMODER_TXFLOW_OFFSET 2**

6.120.1.9 **#define ETH_COLLCONF (4 * 0x07)**

6.120.1.10 **#define ETH_COLLCONF_COLLVALID_OFFSET 0**

6.120.1.11 **#define ETH_COLLCONF_COLLVALID_WIDTH 6**

6.120.1.12 **#define ETH_COLLCONF_MAXRET_OFFSET 16**

6.120.1.13 **#define ETH_COLLCONF_MAXRET_WIDTH 4**

6.120.1.14 **#define ETH_CTRLMODER (4 * 0x09)**

6.120.1.15 **#define ETH_DMA_RX_TX 0x800**

6.120.1.16 **#define ETH_HASH0 (4 * 0x12)**

6.120.1.17 **#define ETH_HASH1 (4 * 0x13)**

6.120.1.18 **#define ETH_INT_MASK (4 * 0x02)**

6.120.1.19 **#define ETH_INT_MASK_BUSY_M_OFFSET 4**

6.120.1.20 **#define ETH_INT_MASK_RXB_M_OFFSET 2**

6.120.1.21 **#define ETH_INT_MASK_RXC_M_OFFSET 6**

6.120.1.22 **#define ETH_INT_MASK_RXE_M_OFFSET 3**

6.120.1.23 **#define ETH_INT_MASK_TXB_M_OFFSET 0**

6.120.1.24 **#define ETH_INT_MASK_TXC_M_OFFSET 5**

6.120.1.25 **#define ETH_INT_MASK_TXE_M_OFFSET 1**

6.120.1.26 **#define ETH_INT_SOURCE (4 * 0x01)**

6.120.1.27 **#define ETH_INT_SOURCE_BUSY_OFFSET 4**

6.120.1.28 **#define ETH_INT_SOURCE_RXB_OFFSET 2**

6.120.1.29 **#define ETH_INT_SOURCE_RXC_OFFSET 6**

6.120.1.30 **#define ETH_INT_SOURCE_RXE_OFFSET 3**

ETH_VAPI_CTRL
ETH_NUM_VAPI_IDS

6.120.3 Function Documentation

6.120.3.1 static void eth_baseaddr (union param_val val, void * dat) [static]

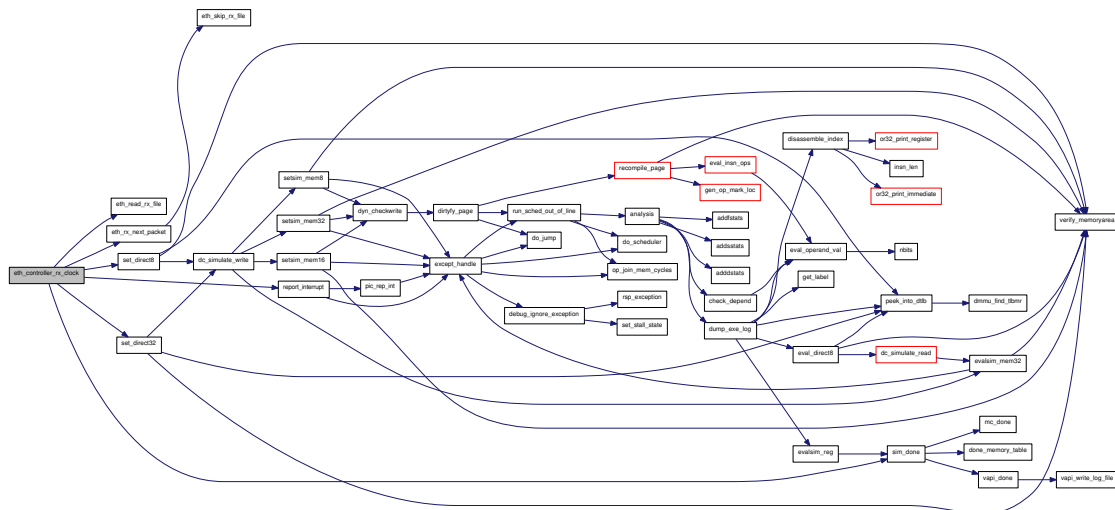
Set the Ethernet interface base address

Parameters:

- ← *val* The value to use
- ← *dat* The [config](#) data structure

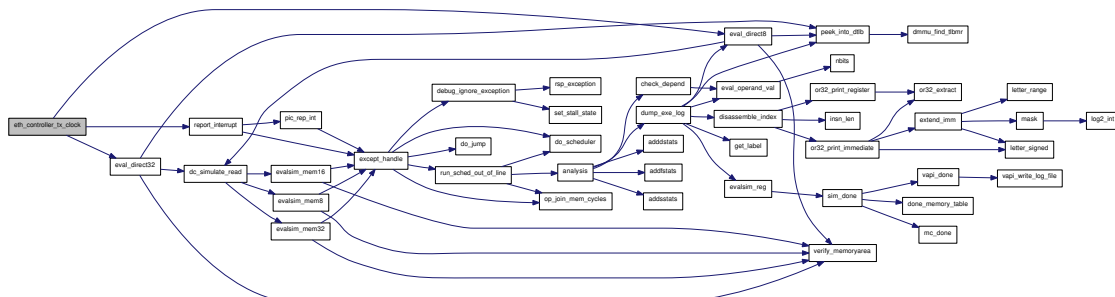
6.120.3.2 static void eth_controller_rx_clock (void * dat) [static]

Here is the call graph for this function:



6.120.3.3 static void eth_controller_tx_clock (void * dat) [static]

Here is the call graph for this function:



6.120.3.4 `static void eth_dma (union param_val val, void * dat)` [static]

Set the Ethernet DMA port

This is not currently supported, so a warning message is printed.

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure

6.120.3.5 `static void eth_enabled (union param_val val, void * dat)` [static]

Enable or disable the Ethernet interface

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure

6.120.3.6 `static void eth_irq (union param_val val, void * dat)` [static]

Set the Ethernet IRQ

Parameters:

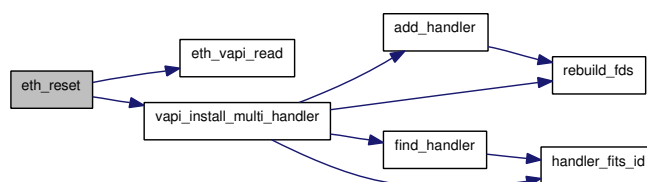
- ← *val* The value to use
- ← *dat* The `config` data structure

6.120.3.7 `static uint32_t eth_read32 (oraddr_t addr, void * dat)` [static]

6.120.3.8 `static ssize_t eth_read_rx_file (struct eth_device * eth, void * buf, size_t count)` [static]

6.120.3.9 `static void eth_reset (void * dat)` [static]

Here is the call graph for this function:



6.120.3.10 static void eth_rtx_type (union param_val val, void * dat) [static]

Set the Ethernet interface type

Currently two types are supported, file and socket. Use of the socket requires a compile time option.

Parameters:

← *val* The value to use. 0 for file, 1 for socket.

← *dat* The [config](#) data structure

6.120.3.11 static void eth_rx_channel (union param_val val, void * dat) [static]

Set the Ethernet DMA Rx [channel](#)

External DMA is not currently supported, so a warning message is printed.

Parameters:

← *val* The value to use

← *dat* The [config](#) data structure

6.120.3.12 static void eth_rx_next_packet (struct eth_device * eth) [static]

Here is the call graph for this function:

**6.120.3.13 static void eth_rxfile (union param_val val, void * dat) [static]**

Set the Ethernet DMA Rx file

Free any previously allocated value.

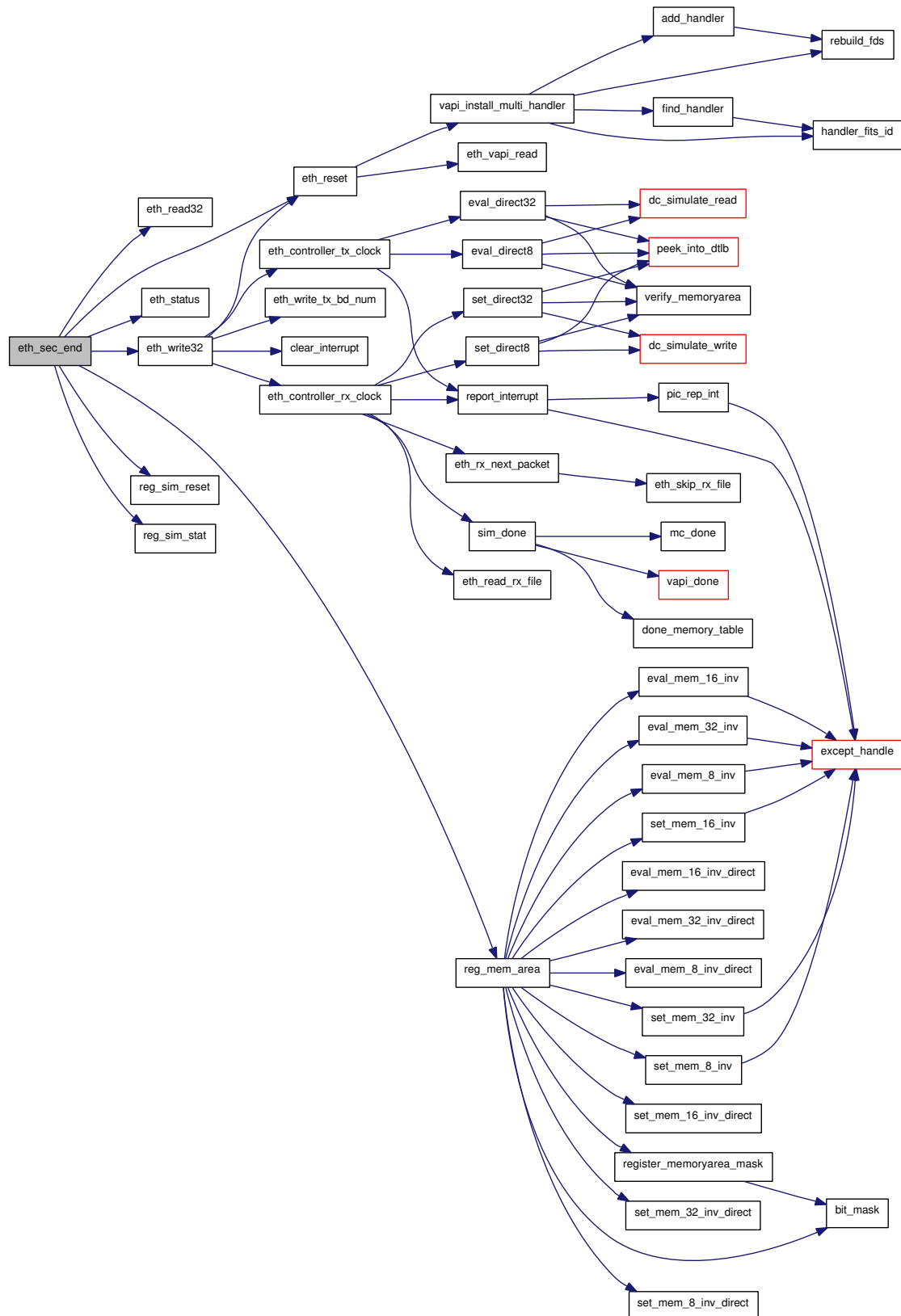
Parameters:

← *val* The value to use

← *dat* The [config](#) data structure

6.120.3.14 static void eth_sec_end (void * dat) [static]

Here is the call graph for this function:



6.120.3.15 `static void* eth_sec_start (void)` [static]

Initialize a new Ethernet configuration

ALL parameters are set explicitly to default values.

6.120.3.16 `static void eth_skip_rx_file (struct eth_device * eth, off_t count)` [static]**6.120.3.17** `static void eth_sockif (union param_val val, void * dat)` [static]

Set the Ethernet socket interface

Free any previously allocated value. This is only meaningful if the socket interface is configured.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.120.3.18 `static void eth_status (void * dat)` [static]**6.120.3.19** `static void eth_tx_channel (union param_val val, void * dat)` [static]

Set the Ethernet DMA Tx `channel`

External DMA is not currently supported, so a warning message is printed.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.120.3.20 `static void eth_txfile (union param_val val, void * dat)` [static]

Set the Ethernet DMA Tx file

Free any previously allocated value.

Parameters:

← *val* The value to use

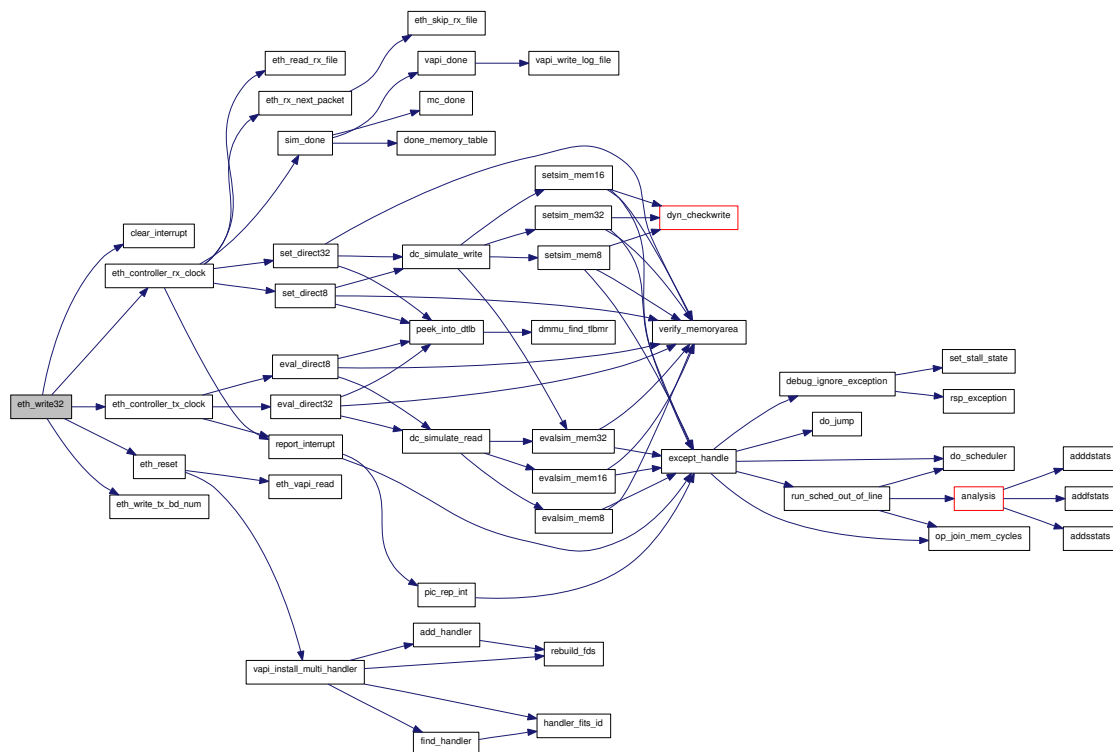
← *dat* The `config` data structure

6.120.3.21 `static void eth_vapi_id (union param_val val, void * dat)` [static]

6.120.3.22 `static void eth_vapi_read (unsigned long id, unsigned long data, void * dat)`
[static]

6.120.3.23 `static void eth_write32 (oraddr_t addr, uint32_t value, void * dat)` [static]

Here is the call graph for this function:

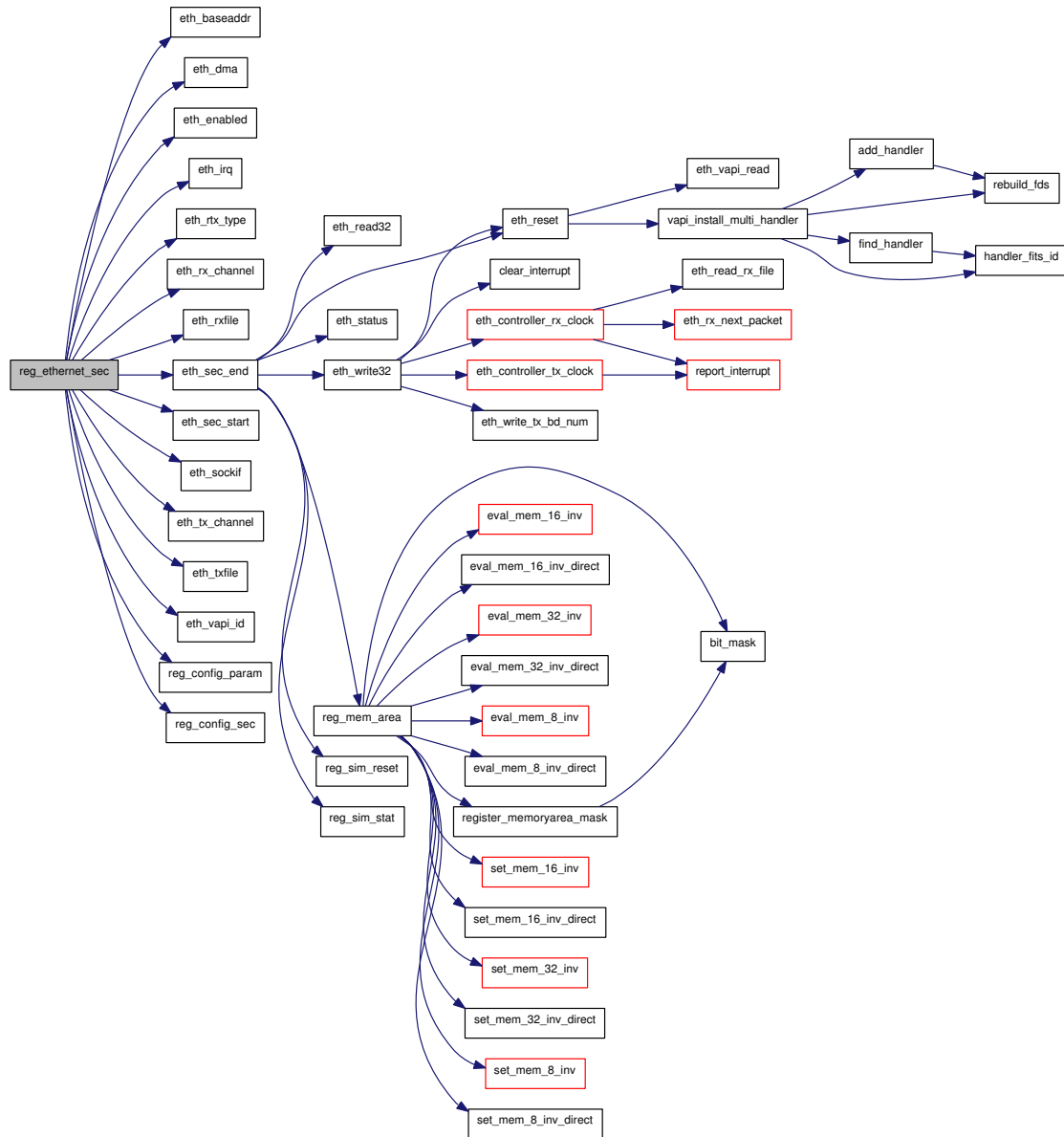


6.120.3.24 `static void eth_write_tx_bd_num (struct eth_device * eth, unsigned long value)`
[static]

6.120.3.25 `void reg_ethernet_sec ()`

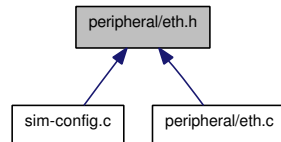
Register a new Ethernet configuration

Here is the call graph for this function:



6.121 peripheral/eth.h File Reference

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



Functions

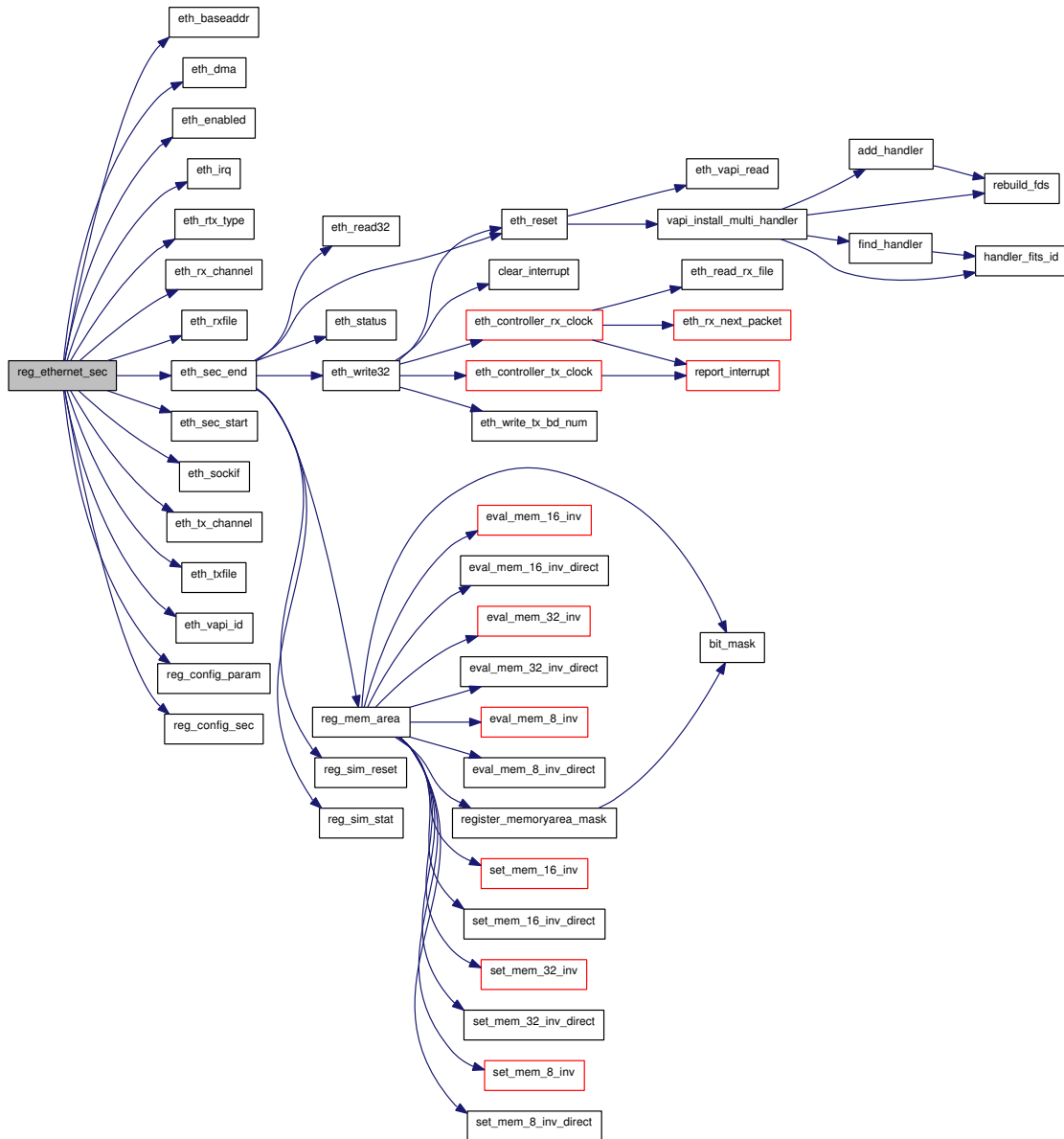
- void [reg_ethernet_sec \(\)](#)

6.121.1 Function Documentation

6.121.1.1 void [reg_ethernet_sec \(\)](#)

Register a new Ethernet configuration

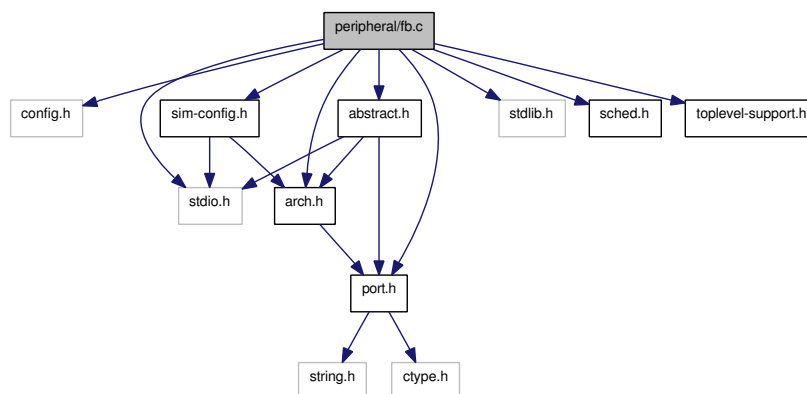
Here is the call graph for this function:



6.122 peripheral/fb.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include "arch.h"
#include "sim-config.h"
#include "abstract.h"
#include "sched.h"
#include "toplevel-support.h"
```

Include dependency graph for fb.c:



Data Structures

- struct `fb_state`

Defines

- #define `FB_SIZEX` 640
- #define `FB_SIZEY` 480
- #define `CAM_SIZEX` 352
- #define `CAM_SIZEY` 288
- #define `REFRESH_DIVIDER` 20
- #define `FB_CTRL` 0x0000
- #define `FB_BUFADDR` 0x0004
- #define `FB_CAMBUFADDR` 0x0008
- #define `FB_CAMPOSADDR` 0x000c
- #define `FB_PAL` 0x0400
- #define `FB_WRAP` (512*1024)
- #define `CNV16(x)` (x)
- #define `CNV32(x)` (x)

Functions

- static void `change_buf_addr` (struct `fb_state` *fb, `oraddr_t` addr)
- static void `fb_write32` (`oraddr_t` addr, `uint32_t` value, void *dat)
- static `oraddr_t` `fb_read32` (`oraddr_t` addr, void *dat)
- static int `fb_dump_image8` (struct `fb_state` *fb, char *filename)
- static int `fb_dump_image24` (struct `fb_state` *fb, char *filename)
- static void `fb_job` (void *dat)
- static void `fb_reset` (void *dat)
- static void `fb_enabled` (union `param_val` val, void *dat)
- static void `fb_baseaddr` (union `param_val` val, void *dat)
- static void `fb_refresh_rate` (union `param_val` val, void *dat)
- static void `fb_filename` (union `param_val` val, void *dat)
- static void * `fb_sec_start` ()
- static void `fb_sec_end` (void *dat)
- void `reg_fb_sec` ()

6.122.1 Define Documentation

6.122.1.1 `#define CAM_SIZEX 352`

6.122.1.2 `#define CAM_SIZEY 288`

6.122.1.3 `#define CNV16(x) (x)`

6.122.1.4 `#define CNV32(x) (x)`

6.122.1.5 `#define FB_BUFADDR 0x0004`

6.122.1.6 `#define FB_CAMBUFADDR 0x0008`

6.122.1.7 `#define FB_CAMPOSADDR 0x000c`

6.122.1.8 `#define FB_CTRL 0x0000`

6.122.1.9 `#define FB_PAL 0x0400`

6.122.1.10 `#define FB_SIZEX 640`

6.122.1.11 `#define FB_SIZEY 480`

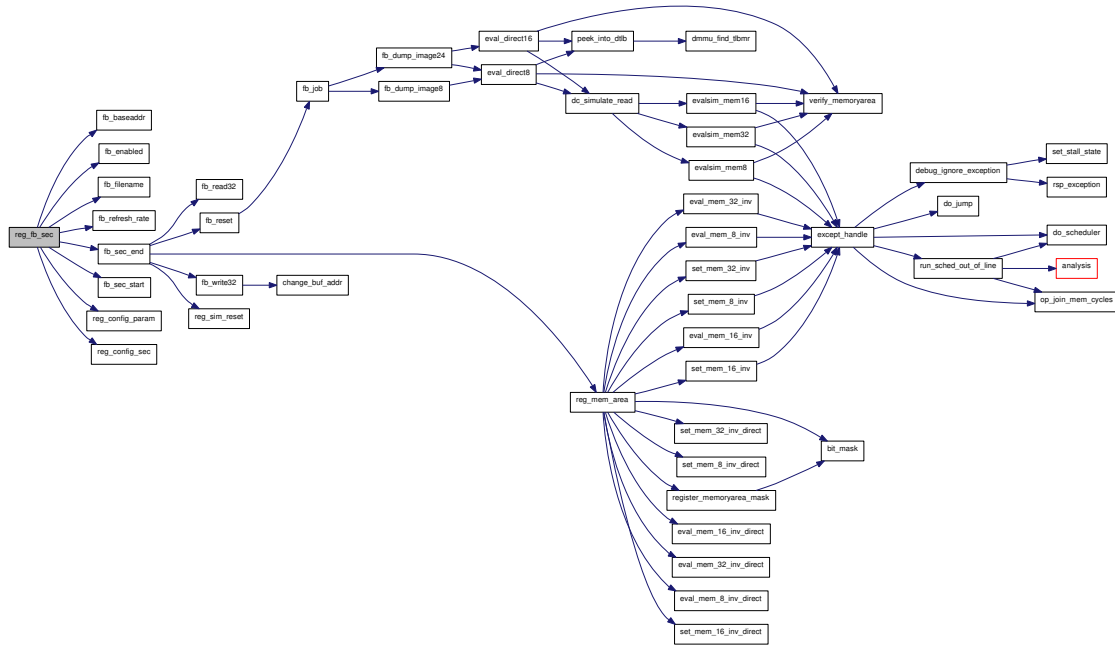
6.122.1.12 `#define FB_WRAP (512*1024)`

6.122.1.13 `#define REFRESH_DIVIDER 20`

Relative amount of time spent in refresh

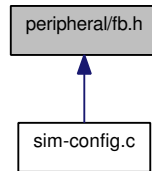
6.122.2.14 void reg_fb_sec ()

Here is the call graph for this function:



6.123 peripheral/fb.h File Reference

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



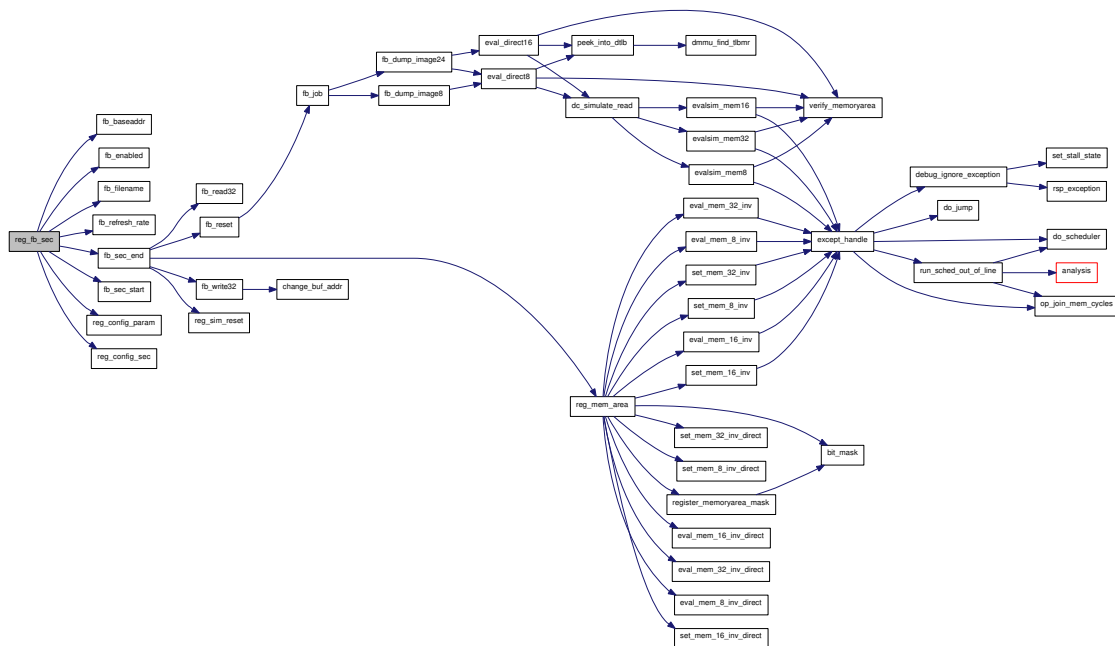
Functions

- void [reg_fb_sec \(\)](#)

6.123.1 Function Documentation

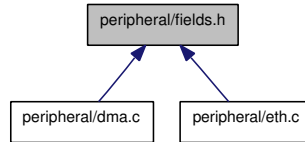
6.123.1.1 void reg_fb_sec ()

Here is the call graph for this function:



6.124 peripheral/fields.h File Reference

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



Defines

- #define **FLAG_SHIFT**(reg_name, flag_name) (reg_name##_##flag_name##_OFFSET)
- #define **FLAG_MASK**(reg_name, flag_name) (1LU << reg_name##_##flag_name##_OFFSET)
- #define **TEST_FLAG**(reg_value, reg_name, flag_name) (((reg_value) >> reg_name##_##flag_name##_OFFSET) & 1LU)
- #define **SET_FLAG**(reg_value, reg_name, flag_name) { (reg_value) |= 1LU << reg_name##_##flag_name##_OFFSET; }
- #define **CLEAR_FLAG**(reg_value, reg_name, flag_name) { (reg_value) &= ~(1LU << reg_name##_##flag_name##_OFFSET); }
- #define **ASSIGN_FLAG**(reg_value, reg_name, flag_name, flag_value)
- #define **FIELD_SHIFT**(reg_name, field_name) (reg_name##_##field_name##_OFFSET)
- #define **FIELD_MASK**(reg_name, field_name) ((~(~0LU << reg_name##_##field_name##_WIDTH)) << reg_name##_##field_name##_OFFSET)
- #define **GET_FIELD**(reg_value, reg_name, field_name) (((reg_value) >> reg_name##_##field_name##_OFFSET) & (~(~0LU << reg_name##_##field_name##_WIDTH)))
- #define **SET_FIELD**(reg_value, reg_name, field_name, field_value)

6.124.1 Define Documentation

6.124.1.1 #define ASSIGN_FLAG(reg_value, reg_name, flag_name, flag_value)

Value:

```

{ \
    (reg_value) = flag_value ? ((reg_value) | (1LU << reg_name##_##flag_name##_OFFSET)) : ((reg_value) & ~
  
```


6.124.1.2 **#define CLEAR_FLAG**(reg_value, reg_name, flag_name) { (reg_value) &= ~(1LU << reg_name##_##flag_name##_OFFSET); }

6.124.1.3 **#define FIELD_MASK**(reg_name, field_name) ((~(~0LU << reg_name##_##field_name##_WIDTH)) << reg_name##_##field_name##_OFFSET)

6.124.1.4 **#define FIELD_SHIFT**(reg_name, field_name) (reg_name##_##field_name##_OFFSET)

6.124.1.5 **#define FLAG_MASK**(reg_name, flag_name) (1LU << reg_name##_##flag_name##_OFFSET)

6.124.1.6 **#define FLAG_SHIFT**(reg_name, flag_name) (reg_name##_##flag_name##_OFFSET)

6.124.1.7 **#define GET_FIELD**(reg_value, reg_name, field_name) (((reg_value) >> reg_name##_##field_name##_OFFSET) & (~(~0LU << reg_name##_##field_name##_WIDTH)))

6.124.1.8 **#define SET_FIELD**(reg_value, reg_name, field_name, field_value)

Value:

```
{ \
    (reg_value) = ((reg_value) & ~((~(~0LU << reg_name##_##field_name##_WIDTH)) << reg_name##_##field_name##_OFFSET)) | ((field_value) << reg_name##_##field_name##_OFFSET); }
```

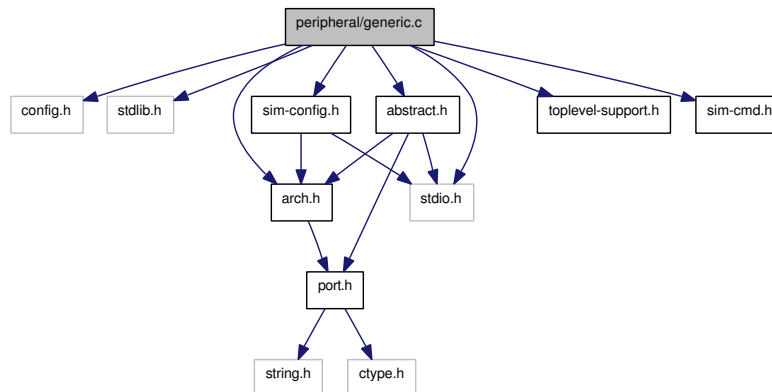
6.124.1.9 **#define SET_FLAG**(reg_value, reg_name, flag_name) { (reg_value) |= 1LU << reg_name##_##flag_name##_OFFSET; }

6.124.1.10 **#define TEST_FLAG**(reg_value, reg_name, flag_name) (((reg_value) >> reg_name##_##flag_name##_OFFSET) & 1LU)

6.125 peripheral/generic.c File Reference

```
#include "config.h"
#include <stdlib.h>
#include <stdio.h>
#include "arch.h"
#include "sim-config.h"
#include "abstract.h"
#include "toplevel-support.h"
#include "sim-cmd.h"
```

Include dependency graph for generic.c:



Data Structures

- struct [dev_generic](#)

Functions

- static unsigned long int [ext_read](#) (unsigned long int addr, unsigned long int mask)
- static void [ext_write](#) (unsigned long int addr, unsigned long int mask, unsigned long int value)
- static uint8_t [generic_read_byte](#) (oraddr_t addr, void *dat)
- static void [generic_write_byte](#) (oraddr_t addr, uint8_t value, void *dat)
- static uint16_t [generic_read_hw](#) (oraddr_t addr, void *dat)
- static void [generic_write_hw](#) (oraddr_t addr, uint16_t value, void *dat)
- static uint32_t [generic_read_word](#) (oraddr_t addr, void *dat)
- static void [generic_write_word](#) (oraddr_t addr, uint32_t value, void *dat)
- static void [generic_reset](#) (void *dat)
- static void [generic_status](#) (void *dat)
- static void [generic_enabled](#) (union param_val val, void *dat)
- static void [generic_byte_enabled](#) (union param_val val, void *dat)
- static void [generic_hw_enabled](#) (union param_val val, void *dat)
- static void [generic_word_enabled](#) (union param_val val, void *dat)

- static void `generic_name` (union `param_val` val, void *dat)
- static void `generic_baseaddr` (union `param_val` val, void *dat)
- static void `generic_size` (union `param_val` val, void *dat)
- static void * `generic_sec_start` ()
- static void `generic_sec_end` (void *dat)
- void `reg_generic_sec` (void)

6.125.1 Function Documentation

6.125.1.1 static unsigned long int `ext_read` (unsigned long int *addr*, unsigned long int *mask*)
[static]

6.125.1.2 static void `ext_write` (unsigned long int *addr*, unsigned long int *mask*, unsigned long int *value*) [static]

6.125.1.3 static void `generic_baseaddr` (union `param_val` *val*, void * *dat*) [static]

6.125.1.4 static void `generic_byte_enabled` (union `param_val` *val*, void * *dat*) [static]

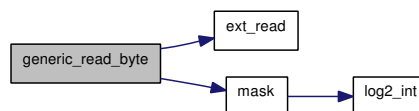
6.125.1.5 static void `generic_enabled` (union `param_val` *val*, void * *dat*) [static]

6.125.1.6 static void `generic_hw_enabled` (union `param_val` *val*, void * *dat*) [static]

6.125.1.7 static void `generic_name` (union `param_val` *val*, void * *dat*) [static]

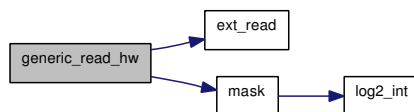
6.125.1.8 static uint8_t `generic_read_byte` (oraddr_t *addr*, void * *dat*) [static]

Here is the call graph for this function:



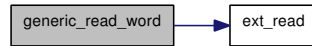
6.125.1.9 static uint16_t `generic_read_hw` (oraddr_t *addr*, void * *dat*) [static]

Here is the call graph for this function:



6.125.1.10 `static uint32_t generic_read_word (oraddr_t addr, void * dat) [static]`

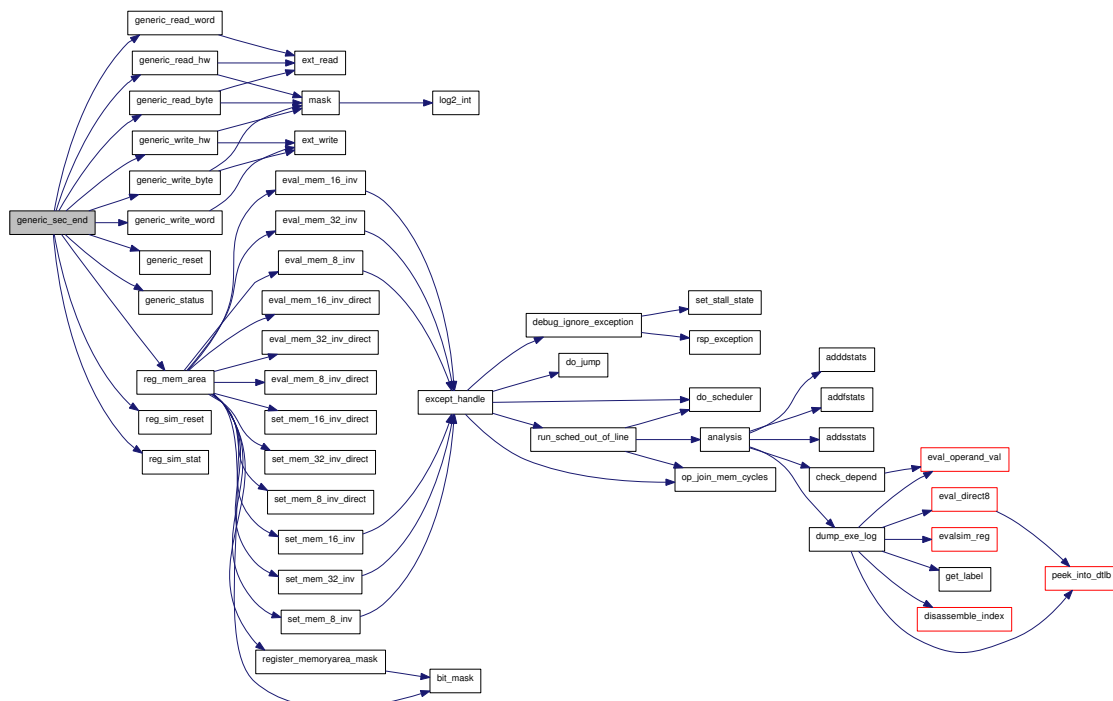
Here is the call graph for this function:



6.125.1.11 `static void generic_reset (void * dat) [static]`

6.125.1.12 `static void generic_sec_end (void * dat) [static]`

Here is the call graph for this function:



6.125.1.13 `static void* generic_sec_start ()` [static]

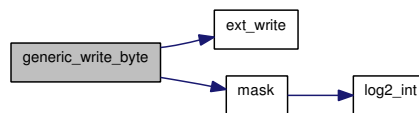
6.125.1.14 `static void generic_size (union param_val val, void * dat)` [static]

6.125.1.15 `static void generic_status (void * dat)` [static]

6.125.1.16 `static void generic_word_enabled (union param_val val, void * dat)` [static]

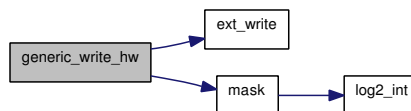
6.125.1.17 `static void generic_write_byte (oraddr_t addr, uint8_t value, void * dat)` [static]

Here is the call graph for this function:



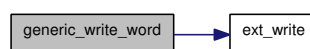
6.125.1.18 `static void generic_write_hw (oraddr_t addr, uint16_t value, void * dat)` [static]

Here is the call graph for this function:



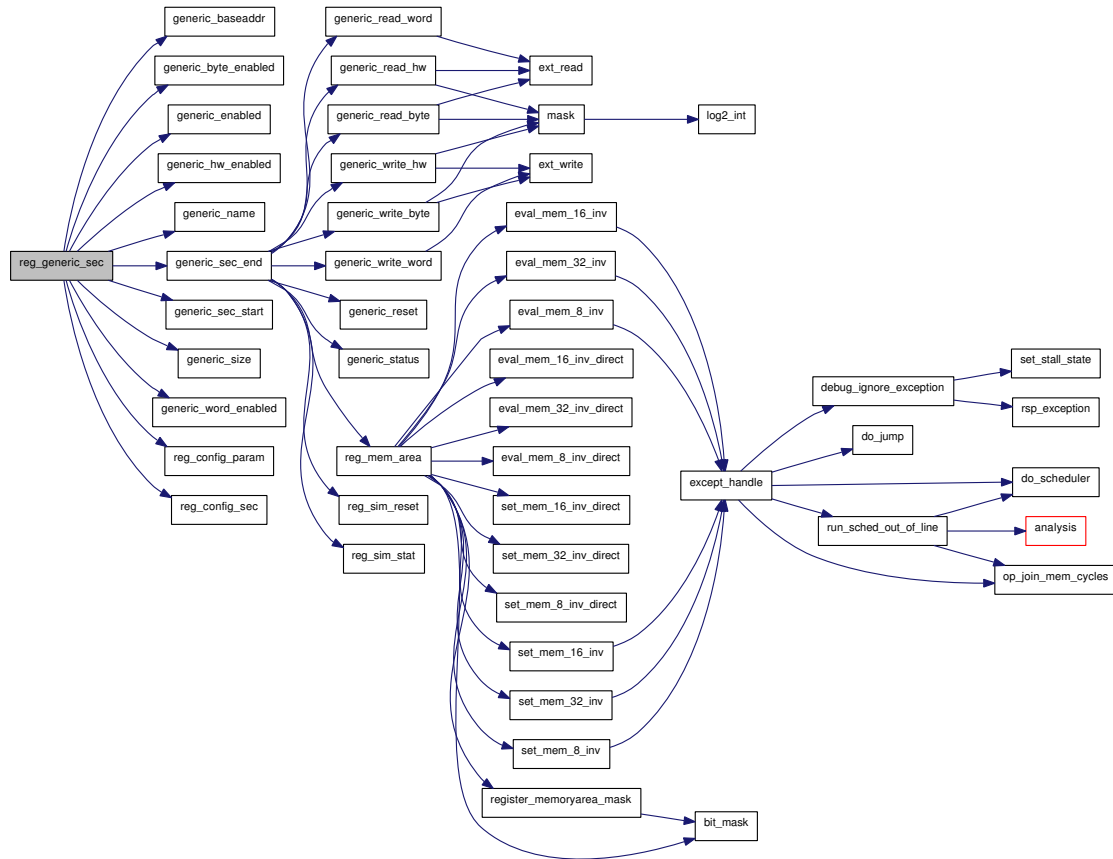
6.125.1.19 `static void generic_write_word (oraddr_t addr, uint32_t value, void * dat)` [static]

Here is the call graph for this function:



6.125.1.20 void reg_generic_sec (void)

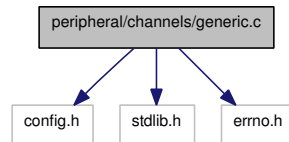
Here is the call graph for this function:



6.126 peripheral/channels/generic.c File Reference

```
#include "config.h"  
#include <stdlib.h>  
#include <errno.h>
```

Include dependency graph for generic.c:



Functions

- int [generic_open](#) (void *data)
- void [generic_close](#) (void *data)
- void [generic_free](#) (void *data)

6.126.1 Function Documentation

6.126.1.1 void [generic_close](#) (void * *data*)

6.126.1.2 void [generic_free](#) (void * *data*)

6.126.1.3 int [generic_open](#) (void * *data*)

6.127 peripheral/generic.h File Reference

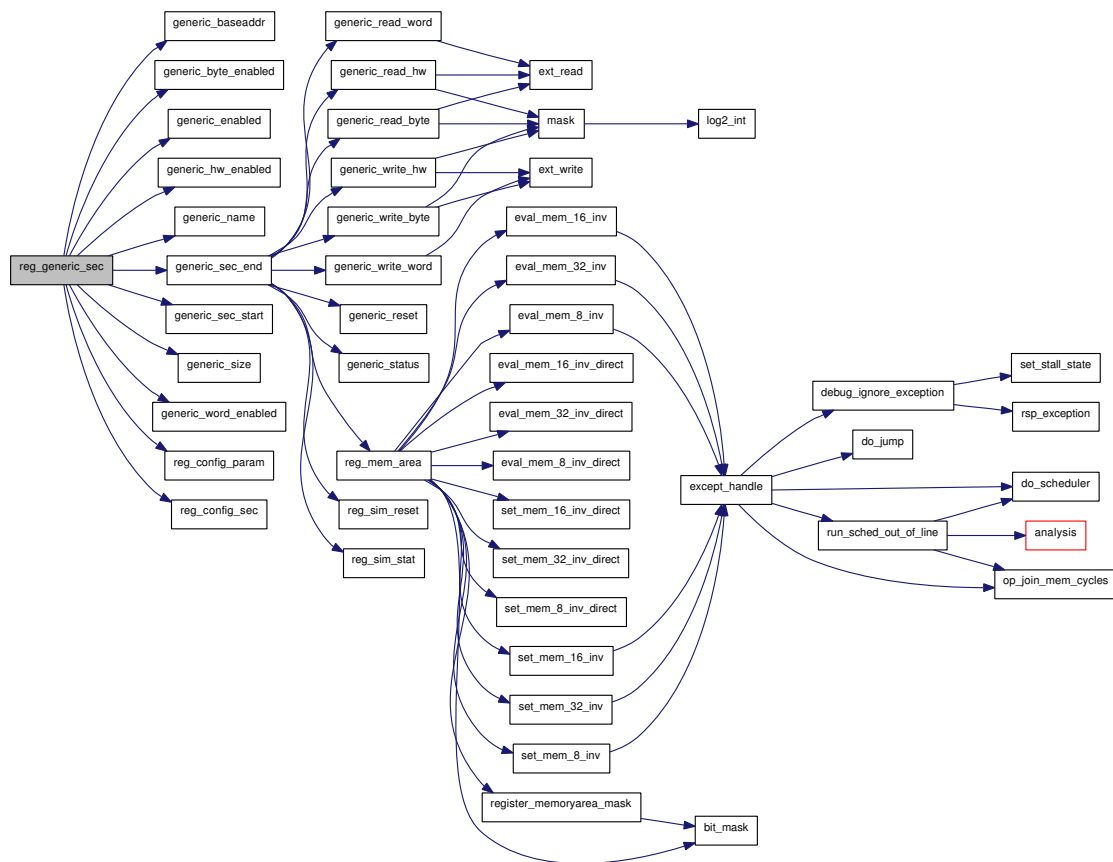
Functions

- void [reg_generic_sec](#) ()

6.127.1 Function Documentation

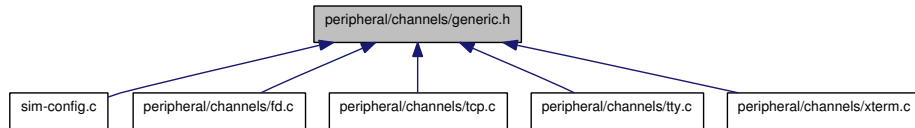
6.127.1.1 void [reg_generic_sec](#) ()

Here is the call graph for this function:



6.128 peripheral/channels/generic.h File Reference

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



Functions

- int [generic_open](#) (void *data)
- void [generic_close](#) (void *data)
- void [generic_free](#) (void *data)

6.128.1 Function Documentation

6.128.1.1 void [generic_close](#) (void * *data*)

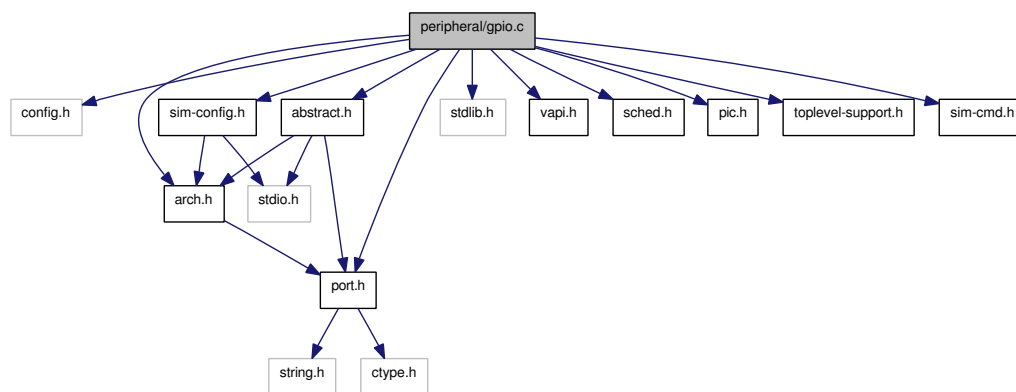
6.128.1.2 void [generic_free](#) (void * *data*)

6.128.1.3 int [generic_open](#) (void * *data*)

6.129 peripheral/gpio.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "sim-config.h"
#include "arch.h"
#include "vapi.h"
#include "sched.h"
#include "pic.h"
#include "abstract.h"
#include "toplevel-support.h"
#include "sim-cmd.h"
```

Include dependency graph for gpio.c:



Data Structures

- struct [gpio_device](#)

Defines

- #define [GPIO_ADDR_SPACE](#) 0x20
- #define [RGPIO_IN](#) 0x00
- #define [RGPIO_OUT](#) 0x04
- #define [RGPIO_OE](#) 0x08
- #define [RGPIO_INTE](#) 0x0C
- #define [RGPIO_PTRIG](#) 0x10
- #define [RGPIO_AUX](#) 0x14
- #define [RGPIO_CTRL](#) 0x18
- #define [RGPIO_INTS](#) 0x1C
- #define [RGPIO_CTRL_ECLK](#) 0x00000001
- #define [RGPIO_CTRL_NEC](#) 0x00000002

- #define [RGPIO_CTRL_INTE](#) 0x00000004
- #define [RGPIO_CTRL_INTS](#) 0x00000008

Enumerations

- enum {

 [GPIO_VAPI_DATA](#) = 0, [GPIO_VAPI_AUX](#), [GPIO_VAPI_CLOCK](#), [GPIO_VAPI_RGPIO_OE](#),

 [GPIO_VAPI_RGPIO_INTE](#), [GPIO_VAPI_RGPIO_PTRIG](#), [GPIO_VAPI_RGPIO_AUX](#), [GPIO_VAPI_RGPIO_CTRL](#),

 [GPIO_NUM_VAPI_IDS](#) }

Functions

- static void [gpio_vapi_read](#) (unsigned long id, unsigned long data, void *dat)
- static void [gpio_external_clock](#) (unsigned long value, struct [gpio_device](#) *gpio)
- static void [gpio_device_clock](#) (struct [gpio_device](#) *gpio)
- static void [gpio_clock](#) (void *dat)
- static void [gpio_reset](#) (void *dat)
- static void [gpio_status](#) (void *dat)
- static uint32_t [gpio_read32](#) (oraddr_t addr, void *dat)
- static void [gpio_write32](#) (oraddr_t addr, uint32_t value, void *dat)
- static void [gpio_do_int](#) (void *dat)
- static void [gpio_baseaddr](#) (union [param_val](#) val, void *dat)
- static void [gpio_irq](#) (union [param_val](#) val, void *dat)
- static void [gpio_base_vapi_id](#) (union [param_val](#) val, void *dat)
- static void [gpio_enabled](#) (union [param_val](#) val, void *dat)
- static void * [gpio_sec_start](#) (void)
- static void [gpio_sec_end](#) (void *dat)
- void [reg_gpio_sec](#) (void)

6.129.1 Define Documentation

- 6.129.1.1 `#define GPIO_ADDR_SPACE 0x20`
- 6.129.1.2 `#define RGPIO_AUX 0x14`
- 6.129.1.3 `#define RGPIO_CTRL 0x18`
- 6.129.1.4 `#define RGPIO_CTRL_ECLK 0x00000001`
- 6.129.1.5 `#define RGPIO_CTRL_INTE 0x00000004`
- 6.129.1.6 `#define RGPIO_CTRL_INTS 0x00000008`
- 6.129.1.7 `#define RGPIO_CTRL_NEC 0x00000002`
- 6.129.1.8 `#define RGPIO_IN 0x00`
- 6.129.1.9 `#define RGPIO_INTE 0x0C`
- 6.129.1.10 `#define RGPIO_INTS 0x1C`
- 6.129.1.11 `#define RGPIO_OE 0x08`
- 6.129.1.12 `#define RGPIO_OUT 0x04`
- 6.129.1.13 `#define RGPIO_PTRIG 0x10`

6.129.2 Enumeration Type Documentation

6.129.2.1 anonymous enum

Enumerator:

GPIO_VAPI_DATA

GPIO_VAPI_AUX

GPIO_VAPI_CLOCK

GPIO_VAPI_RGPIO_OE

GPIO_VAPI_RGPIO_INTE

GPIO_VAPI_RGPIO_PTRIG

GPIO_VAPI_RGPIO_AUX

GPIO_VAPI_RGPIO_CTRL

GPIO_NUM_VAPI_IDS

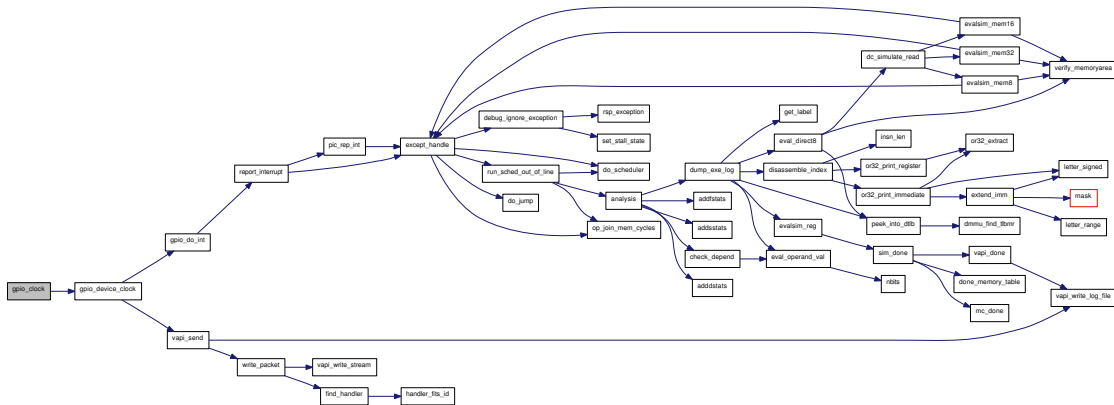
6.129.3 Function Documentation

6.129.3.1 `static void gpio_base_vapi_id (union param_val val, void * dat)` [static]

6.129.3.2 `static void gpio_baseaddr (union param_val val, void * dat)` [static]

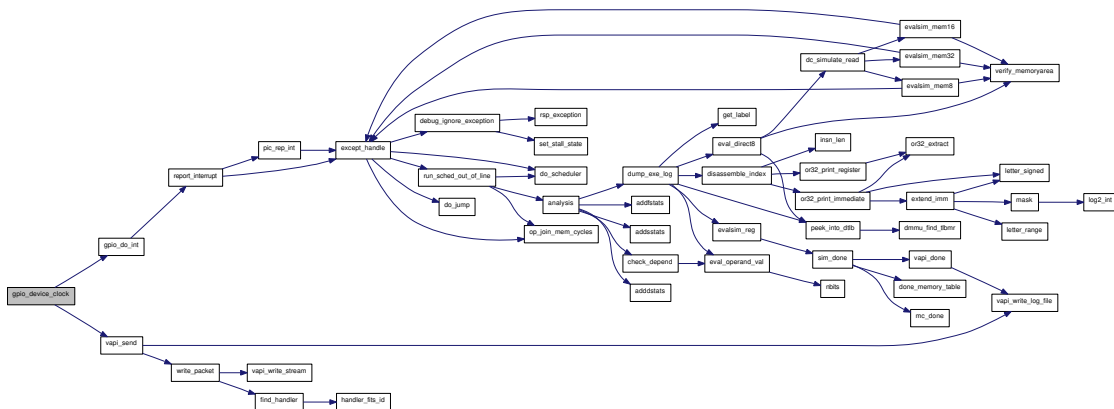
6.129.3.3 `static void gpio_clock (void * dat)` [static]

Here is the call graph for this function:



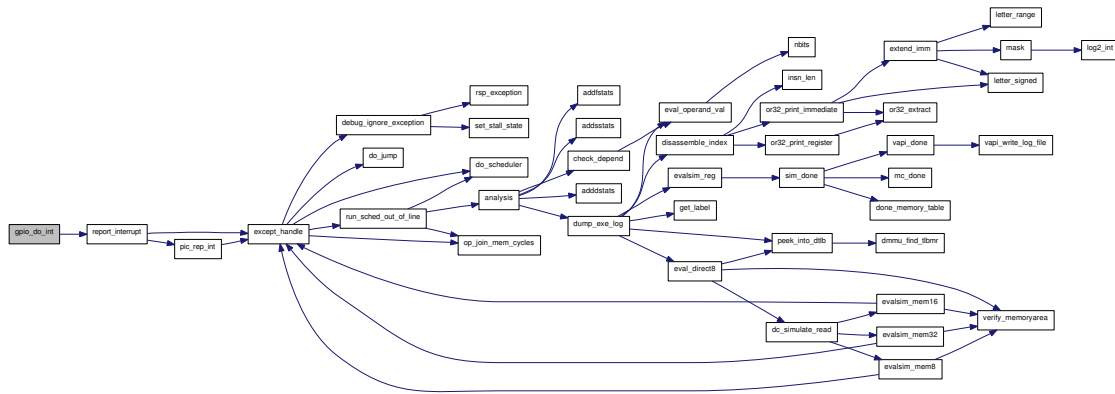
6.129.3.4 `static void gpio_device_clock (struct gpio_device * gpio)` [static]

Here is the call graph for this function:



6.129.3.5 static void gpio_do_int (void * *dat*) [static]

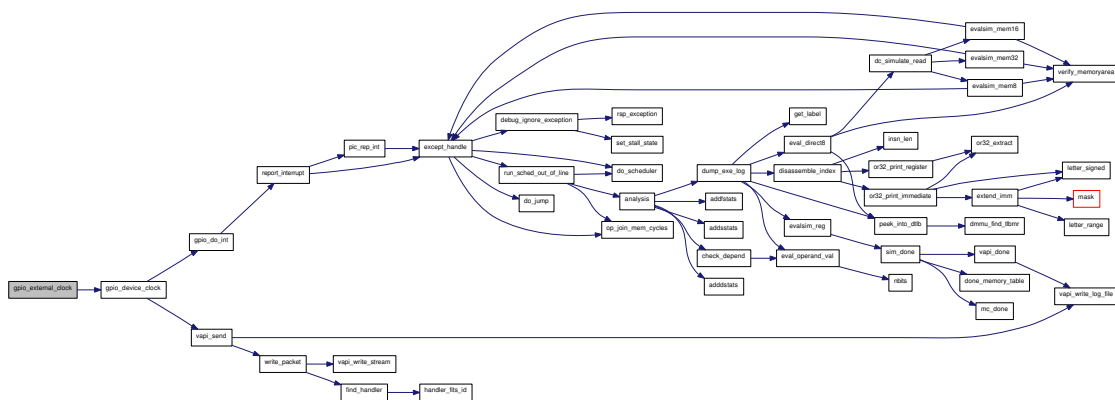
Here is the call graph for this function:



6.129.3.6 static void gpio_enabled (union param_val *val*, void * *dat*) [static]

6.129.3.7 static void gpio_external_clock (unsigned long *value*, struct gpio_device * *gpio*) [static]

Here is the call graph for this function:

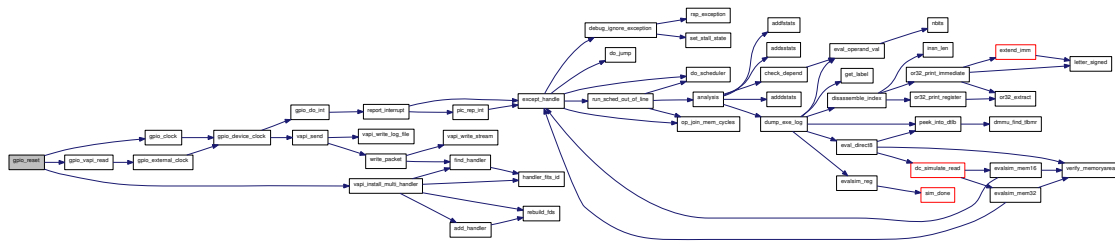


6.129.3.8 static void gpio_irq (union param_val val, void * dat) [static]

6.129.3.9 static uint32_t gpio_read32 (oraddr_t addr, void * dat) [static]

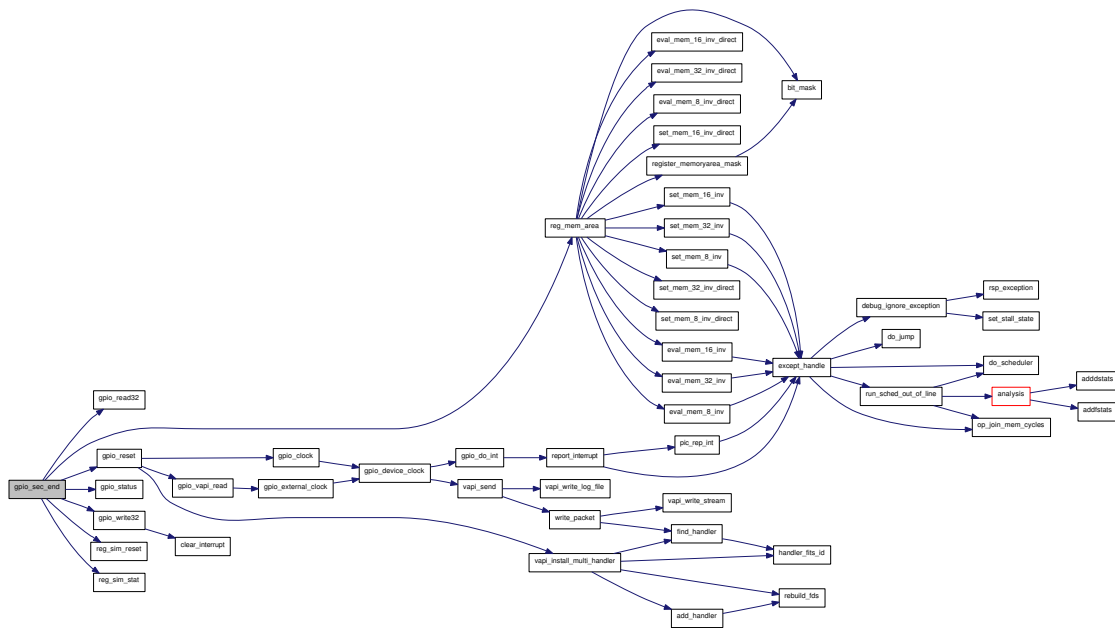
6.129.3.10 static void gpio_reset (void * dat) [static]

Here is the call graph for this function:



6.129.3.11 static void gpio_sec_end (void * dat) [static]

Here is the call graph for this function:

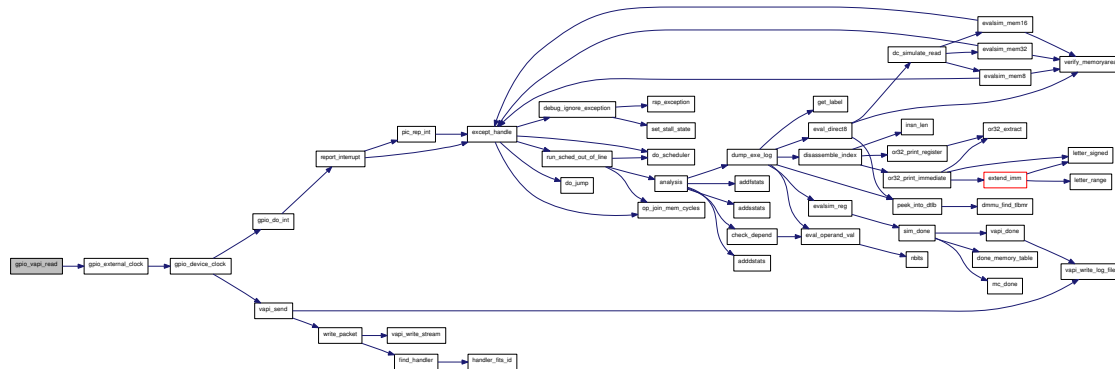


6.129.3.12 `static void* gpio_sec_start (void)` [static]

6.129.3.13 `static void gpio_status (void * dat)` [static]

6.129.3.14 `static void gpio_vapi_read (unsigned long id, unsigned long data, void * dat)`
[static]

Here is the call graph for this function:



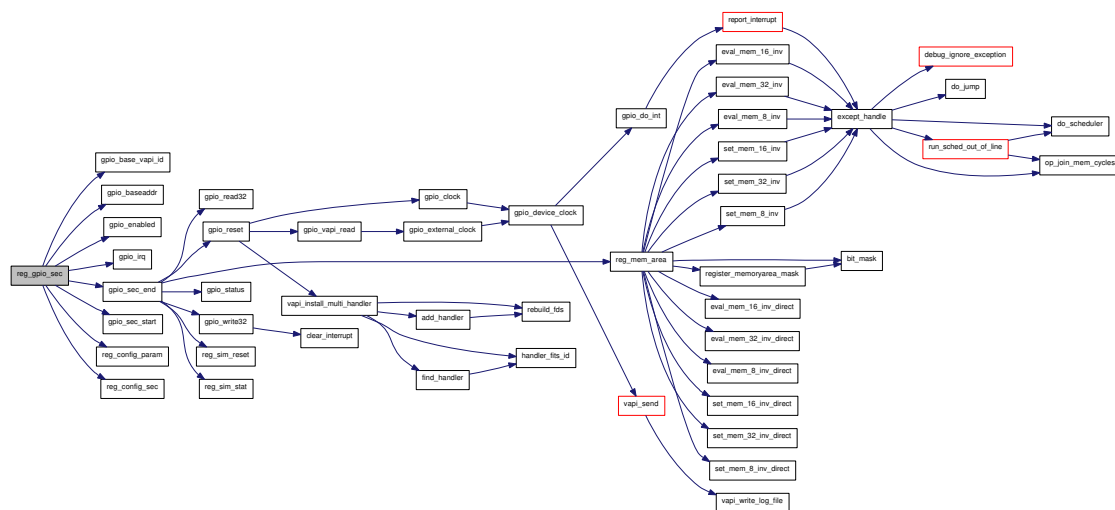
6.129.3.15 `static void gpio_write32 (oraddr_t addr, uint32_t value, void * dat)` [static]

Here is the call graph for this function:



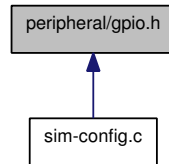
6.129.3.16 `void reg_gpio_sec (void)`

Here is the call graph for this function:



6.130 peripheral/gpio.h File Reference

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



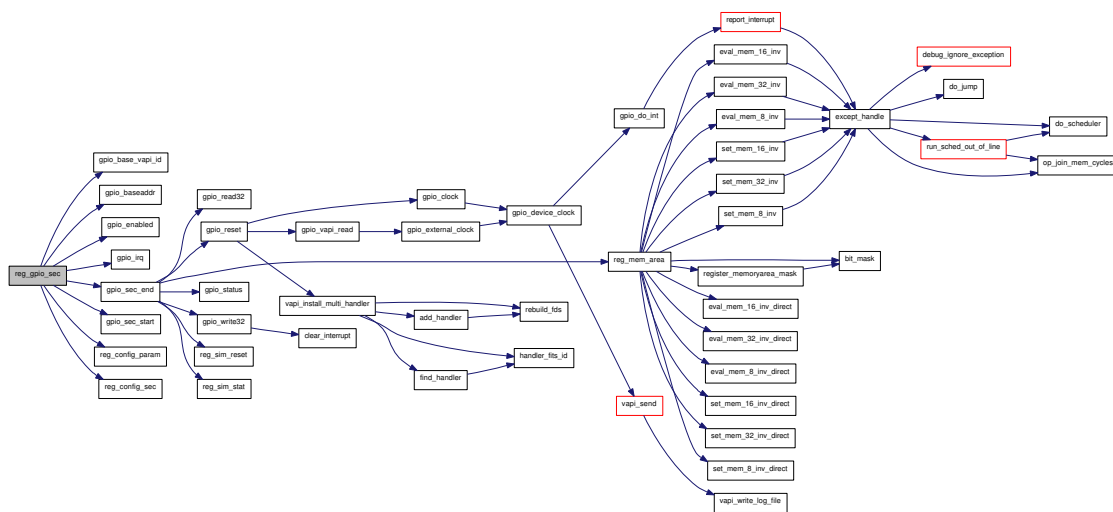
Functions

- void [reg_gpio_sec \(\)](#)

6.130.1 Function Documentation

6.130.1.1 void [reg_gpio_sec \(\)](#)

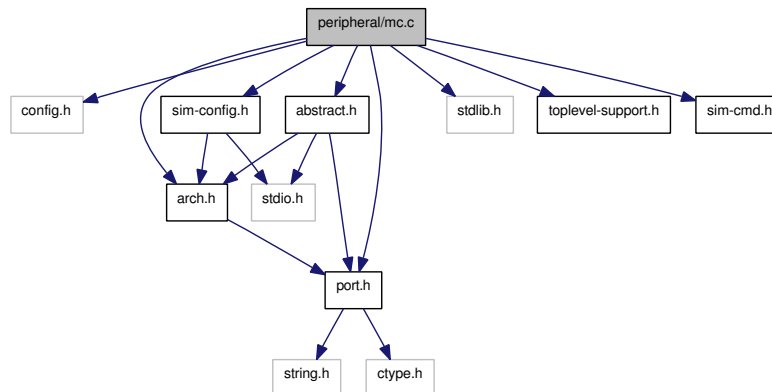
Here is the call graph for this function:



6.131 peripheral/mc.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "arch.h"
#include "abstract.h"
#include "sim-config.h"
#include "toplevel-support.h"
#include "sim-cmd.h"
```

Include dependency graph for mc.c:



Data Structures

- struct [mc_area](#)
- struct [mc](#)

Defines

- #define [N_CE](#) 8
- #define [MC_CSR](#) 0x00
- #define [MC_POC](#) 0x04
- #define [MC_BA_MASK](#) 0x08
- #define [MC_CSC](#)(i) (0x10 + (i) * 8)
- #define [MC_TMS](#)(i) (0x14 + (i) * 8)
- #define [MC_ADDR_SPACE](#) (MC_CSC(N_CE))
- #define [MC_POC_EN_BW_OFFSET](#) 0
- #define [MC_POC_EN_BW_WIDTH](#) 2
- #define [MC_POC_EN_MEMTYPE_OFFSET](#) 2
- #define [MC_POC_EN_MEMTYPE_WIDTH](#) 2
- #define [MC_CSC_EN_OFFSET](#) 0
- #define [MC_CSC_MEMTYPE_OFFSET](#) 1

- #define MC_CSC_MEMTYPE_WIDTH 2
- #define MC_CSC_BW_OFFSET 4
- #define MC_CSC_BW_WIDTH 2
- #define MC_CSC_MS_OFFSET 6
- #define MC_CSC_MS_WIDTH 2
- #define MC_CSC_WP_OFFSET 8
- #define MC_CSC_BAS_OFFSET 9
- #define MC_CSC_KRO_OFFSET 10
- #define MC_CSC_PEN_OFFSET 11
- #define MC_CSC_SEL_OFFSET 16
- #define MC_CSC_SEL_WIDTH 8
- #define MC_CSC_MEMTYPE_SDRAM 0
- #define MC_CSC_MEMTYPE_SSRAM 1
- #define MC_CSC_MEMTYPE_ASYNC 2
- #define MC_CSC_MEMTYPE_SYNC 3
- #define MC_CSR_VALID 0xFF000703LU
- #define MC_POC_VALID 0x0000000FLU
- #define MC_BA_MASK_VALID 0x000003FFLU
- #define MC_CSC_VALID 0x00FF0FFFU
- #define MC_TMS_SDRAM_VALID 0x0FFF83FFU
- #define MC_TMS_SSRAM_VALID 0x00000000LU
- #define MC_TMS_ASYNC_VALID 0x03FFFFFFU
- #define MC_TMS_SYNC_VALID 0x01FFFFFFU
- #define MC_TMS_VALID 0xFFFFFFFU
- #define MC_TMS_SDRAM_TRFC_OFFSET 24
- #define MC_TMS_SDRAM_TRFC_WIDTH 4
- #define MC_TMS_SDRAM_TRP_OFFSET 20
- #define MC_TMS_SDRAM_TRP_WIDTH 4
- #define MC_TMS_SDRAM_TRCD_OFFSET 17
- #define MC_TMS_SDRAM_TRCD_WIDTH 4
- #define MC_TMS_SDRAM_TWR_OFFSET 15
- #define MC_TMS_SDRAM_TWR_WIDTH 2
- #define MC_TMS_SDRAM_WBL_OFFSET 9
- #define MC_TMS_SDRAM_OM_OFFSET 7
- #define MC_TMS_SDRAM_OM_WIDTH 2
- #define MC_TMS_SDRAM_CL_OFFSET 4
- #define MC_TMS_SDRAM_CL_WIDTH 3
- #define MC_TMS_SDRAM_BT_OFFSET 3
- #define MC_TMS_SDRAM_BL_OFFSET 0
- #define MC_TMS_SDRAM_BL_WIDTH 3
- #define MC_TMS_ASYNC_TWWD_OFFSET 20
- #define MC_TMS_ASYNC_TWWD_WIDTH 6
- #define MC_TMS_ASYNC_TWD_OFFSET 16
- #define MC_TMS_ASYNC_TWD_WIDTH 4
- #define MC_TMS_ASYNC_TWPW_OFFSET 12
- #define MC_TMS_ASYNC_TWPW_WIDTH 4
- #define MC_TMS_ASYNC_TRDZ_OFFSET 8
- #define MC_TMS_ASYNC_TRDZ_WIDTH 4
- #define MC_TMS_ASYNC_TRDV_OFFSET 0
- #define MC_TMS_ASYNC_TRDV_WIDTH 8

- #define `MC_TMS_SYNC_TTO_OFFSET` 16
- #define `MC_TMS_SYNC_TTO_WIDTH` 9
- #define `MC_TMS_SYNC_TWR_OFFSET` 12
- #define `MC_TMS_SYNC_TWR_WIDTH` 4
- #define `MC_TMS_SYNC_TRDZ_OFFSET` 8
- #define `MC_TMS_SYNC_TRDZ_WIDTH` 4
- #define `MC_TMS_SYNC_TRDV_OFFSET` 0
- #define `MC_TMS_SYNC_TRDV_WIDTH` 8

Functions

- void `set_csc_tms` (int cs, uint32_t csc, uint32_t tms, struct `mc` *mc)
- void `mc_write_word` (oraddr_t addr, uint32_t value, void *dat)
- uint32_t `mc_read_word` (oraddr_t addr, void *dat)
- void `mc_reset` (void *dat)
- void `mc_done` ()
- void `mc_status` (void *dat)
- void `mc_reg_mem_area` (struct `dev_memarea` *mem, unsigned int cs, int mc)
- static void `mc_enabled` (union `param_val` val, void *dat)
- static void `mc_baseaddr` (union `param_val` val, void *dat)
- static void `mc_poc` (union `param_val` val, void *dat)
- static void `mc_index` (union `param_val` val, void *dat)
- static void * `mc_sec_start` ()
- static void `mc_sec_end` (void *dat)
- void `reg_mc_sec` (void)

Variables

- static struct `mc` * `mcs` = NULL
- static struct `mc_area` * `mc_areas` = NULL

6.131.1 Define Documentation

6.131.1.1 **#define MC_ADDR_SPACE (MC_CSC(N_CE))**

6.131.1.2 **#define MC_BA_MASK 0x08**

6.131.1.3 **#define MC_BA_MASK_VALID 0x000003FFLU**

6.131.1.4 **#define MC_CSC(i) (0x10 + (i) * 8)**

6.131.1.5 **#define MC_CSC_BAS_OFFSET 9**

6.131.1.6 **#define MC_CSC_BW_OFFSET 4**

6.131.1.7 **#define MC_CSC_BW_WIDTH 2**

6.131.1.8 **#define MC_CSC_EN_OFFSET 0**

6.131.1.9 **#define MC_CSC_KRO_OFFSET 10**

6.131.1.10 **#define MC_CSC_MEMTYPE_ASYNC 2**

6.131.1.11 **#define MC_CSC_MEMTYPE_OFFSET 1**

6.131.1.12 **#define MC_CSC_MEMTYPE_SDRAM 0**

6.131.1.13 **#define MC_CSC_MEMTYPE_SSRAM 1**

6.131.1.14 **#define MC_CSC_MEMTYPE_SYNC 3**

6.131.1.15 **#define MC_CSC_MEMTYPE_WIDTH 2**

6.131.1.16 **#define MC_CSC_MS_OFFSET 6**

6.131.1.17 **#define MC_CSC_MS_WIDTH 2**

6.131.1.18 **#define MC_CSC_PEN_OFFSET 11**

6.131.1.19 **#define MC_CSC_SEL_OFFSET 16**

6.131.1.20 **#define MC_CSC_SEL_WIDTH 8**

6.131.1.21 **#define MC_CSC_VALID 0x00FF0FFF LU**

6.131.1.22 **#define MC_CSC_WP_OFFSET 8**

6.131.1.23 **#define MC_CSR 0x00**

6.131.1.24 **#define MC_CSR_VALID 0xFF000703 LU**

6.131.1.25 **#define MC_POC 0x04**

6.131.1.26 **#define MC_POC_EN_BW_OFFSET 0**

6.131.1.27 **#define MC_POC_EN_BW_WIDTH 2**

6.131.1.28 **#define MC_POC_EN_MEMTYPE_OFFSET 2**

6.131.1.29 **#define MC_POC_EN_MEMTYPE_WIDTH 2**

6.131.1.30 **#define MC_POC_VALID 0x0000000FLU**

6.131.2.3 `static void mc_enabled (union param_val val, void * dat) [static]`

6.131.2.4 `static void mc_index (union param_val val, void * dat) [static]`

6.131.2.5 `static void mc_poc (union param_val val, void * dat) [static]`

Set the power on configuration state

Only the bottom 4 bits are significant. Other bits are truncated with a warning.

Parameters:

← *val* The value to use

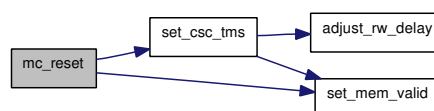
← *dat* The [config](#) data structure

6.131.2.6 `uint32_t mc_read_word (oraddr_t addr, void * dat)`

6.131.2.7 `void mc_reg_mem_area (struct dev_memarea * mem, unsigned int cs, int mc)`

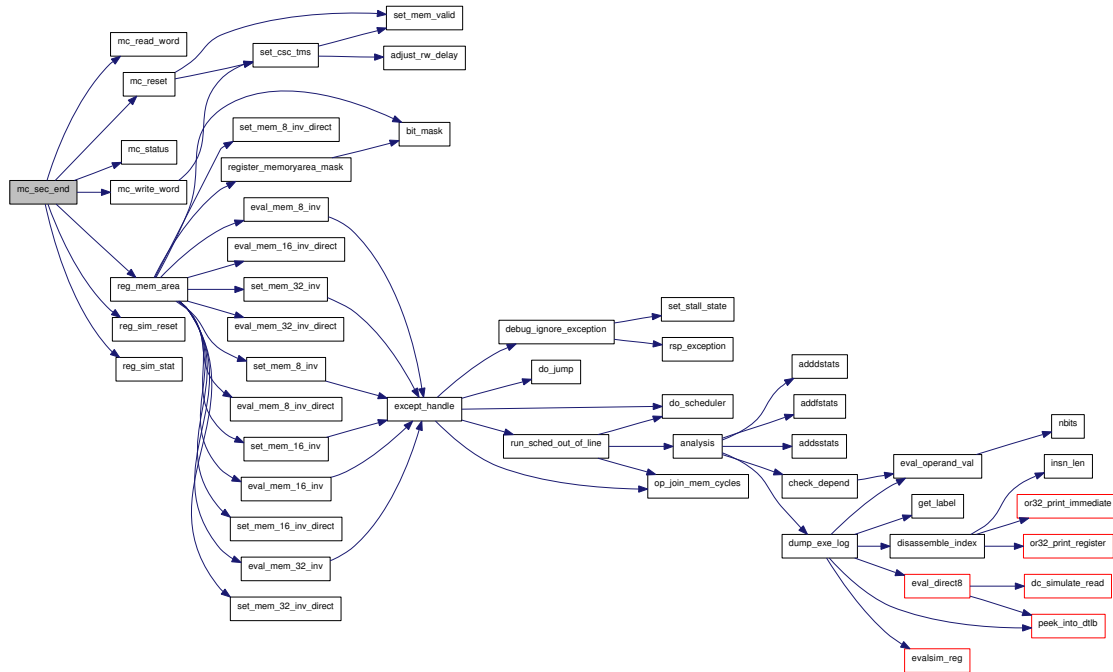
6.131.2.8 `void mc_reset (void * dat)`

Here is the call graph for this function:



6.131.2.9 static void mc_sec_end (void * dat) [static]

Here is the call graph for this function:



6.131.2.10 static void* mc_sec_start () [static]

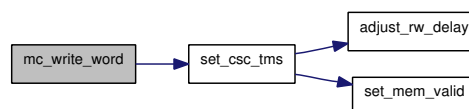
Initialize a new memory controller configuration

ALL parameters are set explicitly to default values.

6.131.2.11 void mc_status (void * dat)

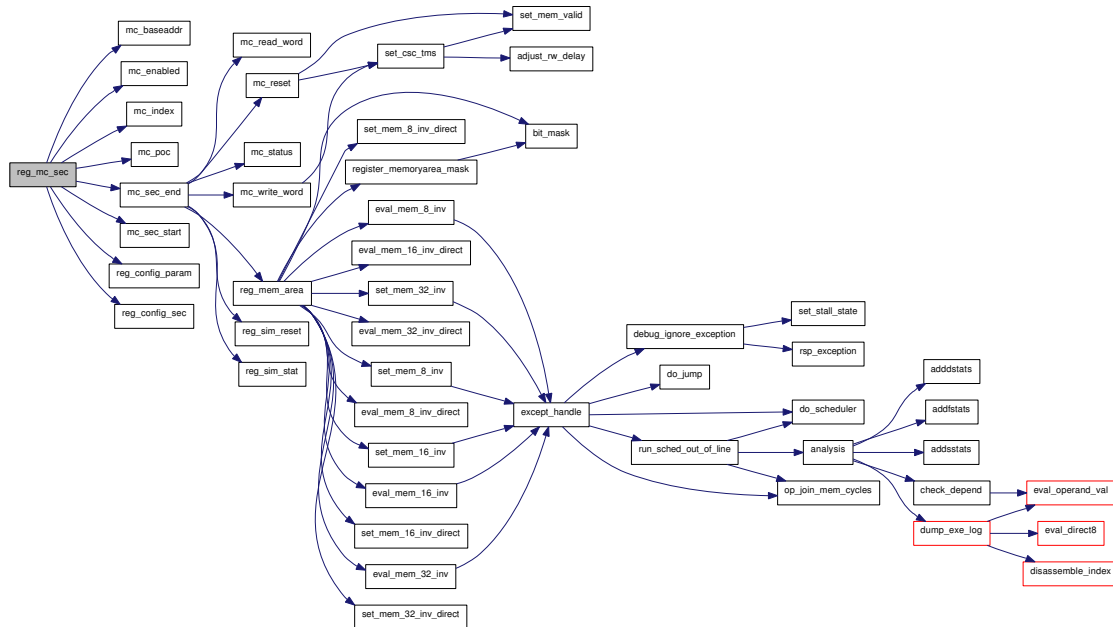
6.131.2.12 void mc_write_word (oraddr_t addr, uint32_t value, void * dat)

Here is the call graph for this function:



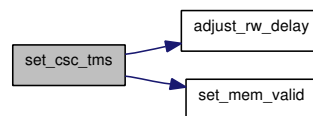
6.131.2.13 void reg_mc_sec (void)

Here is the call graph for this function:



6.131.2.14 void set_csc_tms (int cs, uint32_t csc, uint32_t tms, struct mc * mc)

Here is the call graph for this function:



6.131.3 Variable Documentation

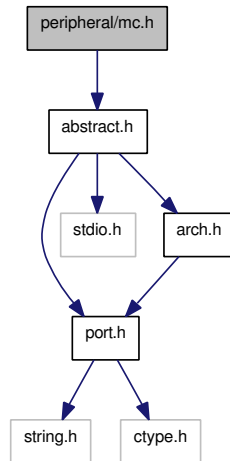
6.131.3.1 struct mc_area* mc_areas = NULL [static]

6.131.3.2 struct mc* mcs = NULL [static]

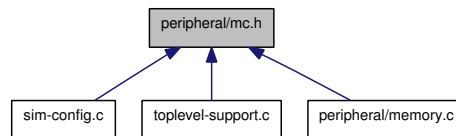
6.132 peripheral/mc.h File Reference

```
#include "abstract.h"
```

Include dependency graph for mc.h:



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



Functions

- void `mc_done` ()
- void `reg_mc_sec` ()
- void `mc_reg_mem_area` (struct `dev_memarea` *mem, unsigned int cs, int mc)

6.132.1 Function Documentation

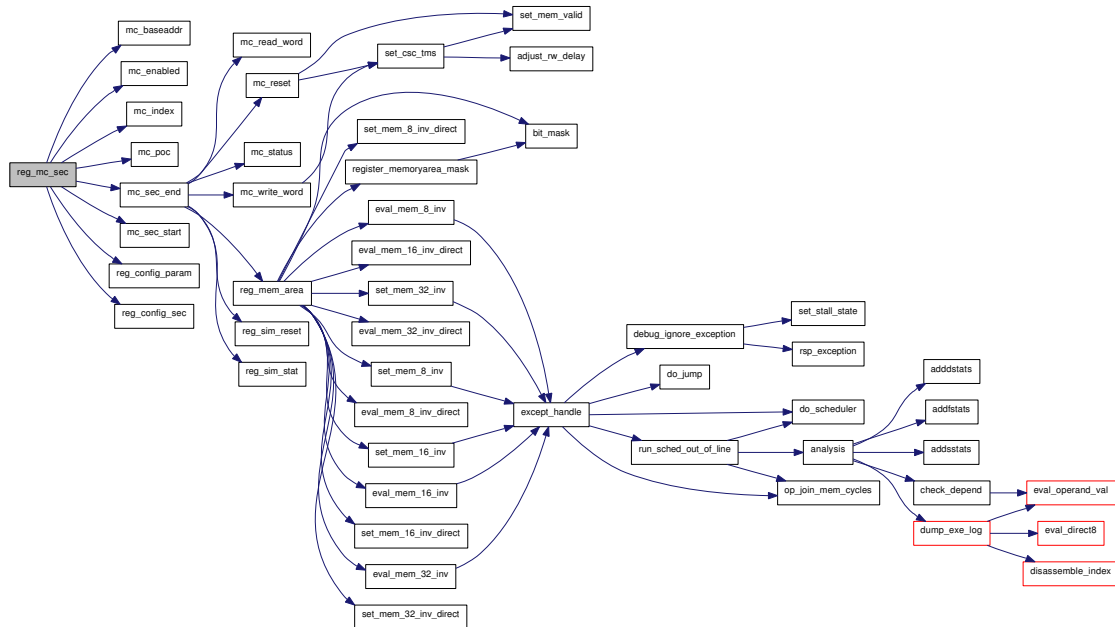
6.132.1.1 void mc_done ()

Free all allocated memory

6.132.1.2 void mc_reg_mem_area (struct dev_memarea * *mem*, unsigned int *cs*, int *mc*)

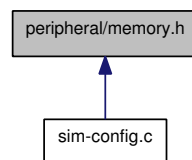
6.132.1.3 void reg_mc_sec ()

Here is the call graph for this function:



6.133 peripheral/memory.h File Reference

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



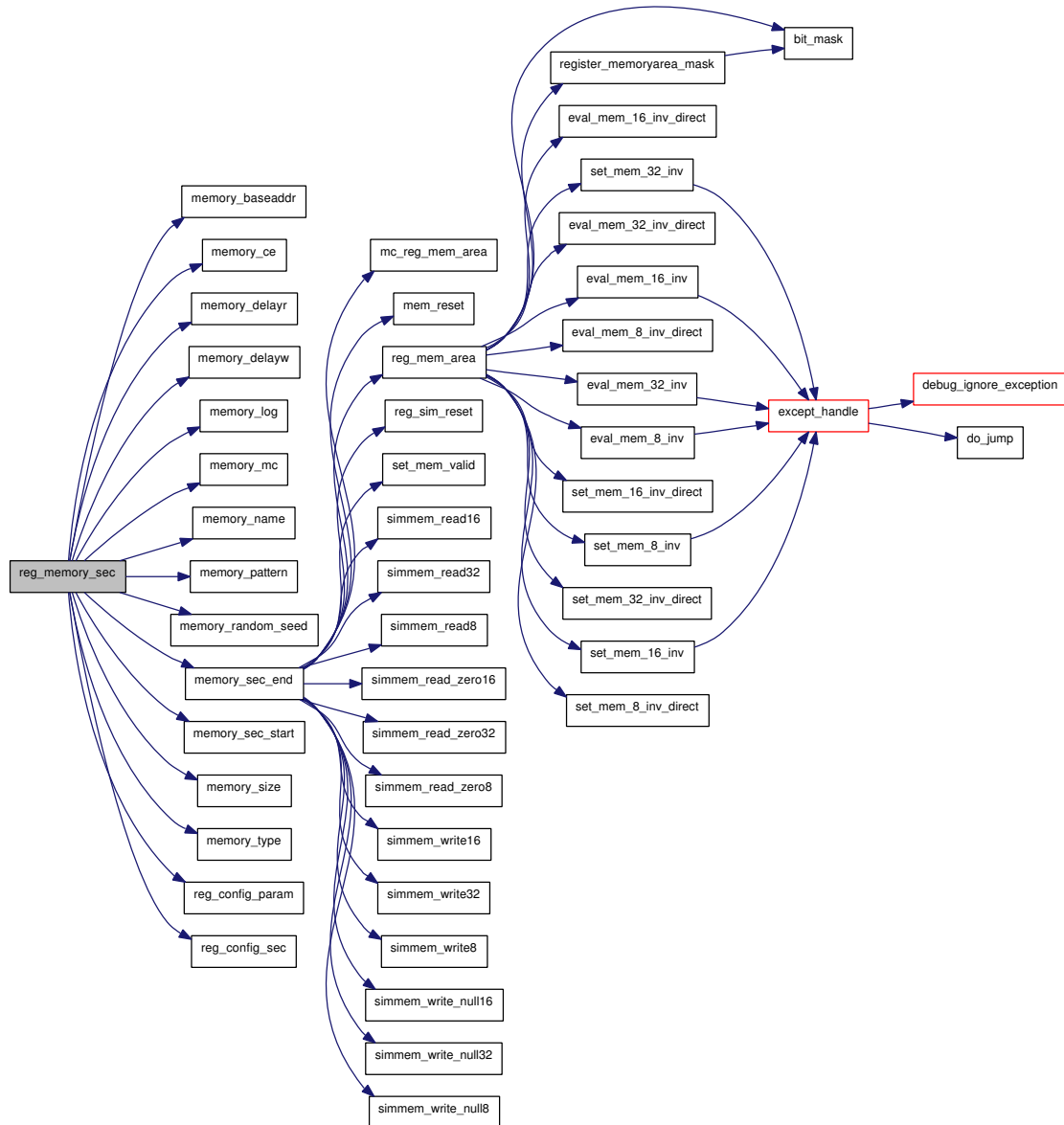
Functions

- void [reg_memory_sec](#) ()

6.133.1 Function Documentation

6.133.1.1 void reg_memory_sec ()

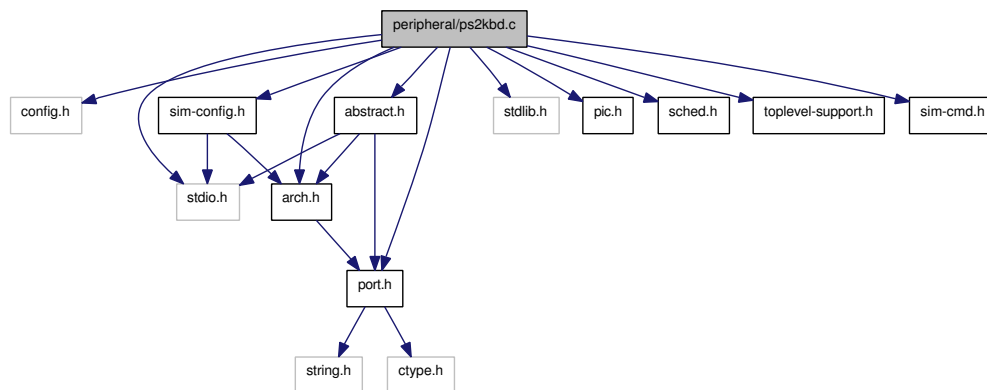
Here is the call graph for this function:



6.134 peripheral/ps2kbd.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include "arch.h"
#include "pic.h"
#include "sim-config.h"
#include "abstract.h"
#include "sched.h"
#include "toplevel-support.h"
#include "sim-cmd.h"
```

Include dependency graph for ps2kbd.c:



Data Structures

- struct [kbd_state](#)

Defines

- #define [KBD_CTRL](#) 4
- #define [KBD_DATA](#) 0
- #define [KBD_SPACE](#) 8
- #define [KBD_KCMD_RST](#) 0xFF
- #define [KBD_KCMD_DK](#) 0xF5
- #define [KBD_KCMD_EK](#) 0xF4
- #define [KBD_KCMD_ECHO](#) 0xFF
- #define [KBD_KCMD_SRL](#) 0xED
- #define [KBD_KRESP_RSTOK](#) 0xAA
- #define [KBD_KRESP_ECHO](#) 0xEE
- #define [KBD_KRESP_ACK](#) 0xFA

- #define `KBD_CCMD_RCB` 0x20
- #define `KBD_CCMD_WCB` 0x60
- #define `KBD_CCMD_ST1` 0xAA
- #define `KBD_CCMD_ST2` 0xAB
- #define `KBD_CCMD_DKI` 0xAD
- #define `KBD_CCMD_EKI` 0xAE
- #define `KBD_STATUS_OBF` 0x01
- #define `KBD_STATUS_IBF` 0x02
- #define `KBD_STATUS_SYS` 0x04
- #define `KBD_STATUS_A2` 0x08
- #define `KBD_STATUS_INH` 0x10
- #define `KBD_STATUS_MOBF` 0x20
- #define `KBD_STATUS_TO` 0x40
- #define `KBD_STATUS_PERR` 0x80
- #define `KBD_CCMDBYTE_INT` 0x01
- #define `KBD_CCMDBYTE_INT2` 0x02
- #define `KBD_CCMDBYTE_SYS` 0x04
- #define `KBD_CCMDBYTE_EN` 0x10
- #define `KBD_CCMDBYTE_EN2` 0x20
- #define `KBD_CCMDBYTE_XLAT` 0x40
- #define `KBD_MAX_BUF` 0x100
- #define `KBD_BAUD_RATE` 1200

Functions

- static void `kbd_put` (struct `kbd_state` *kbd, unsigned char c)
- static void `scan_decode` (struct `kbd_state` *kbd, unsigned char c)
- static void `kbd_write8` (`oraddr_t` addr, `uint8_t` value, void *dat)
- static `uint8_t` `kbd_read8` (`oraddr_t` addr, void *dat)
- static void `kbd_job` (void *dat)
- static void `kbd_reset` (void *dat)
- static void `kbd_info` (void *dat)
- static void `kbd_enabled` (union `param_val` val, void *dat)
- static void `kbd_baseaddr` (union `param_val` val, void *dat)
- static void `kbd_irq` (union `param_val` val, void *dat)
- static void `kbd_rxfile` (union `param_val` val, void *dat)
- static void * `kbd_sec_start` ()
- static void `kbd_sec_end` (void *dat)
- void `reg_kbd_sec` ()

Variables

- struct {
 - unsigned char `shift`
 - unsigned char `code`
- } `scan_table` [128]

6.134.1 Define Documentation

- 6.134.1.1 `#define KBD_BAUD_RATE 1200`
- 6.134.1.2 `#define KBD_CCMD_DKI 0xAD`
- 6.134.1.3 `#define KBD_CCMD_EKI 0xAE`
- 6.134.1.4 `#define KBD_CCMD_RCB 0x20`
- 6.134.1.5 `#define KBD_CCMD_ST1 0xAA`
- 6.134.1.6 `#define KBD_CCMD_ST2 0xAB`
- 6.134.1.7 `#define KBD_CCMD_WCB 0x60`
- 6.134.1.8 `#define KBD_CCMDBYTE_EN 0x10`
- 6.134.1.9 `#define KBD_CCMDBYTE_EN2 0x20`
- 6.134.1.10 `#define KBD_CCMDBYTE_INT 0x01`
- 6.134.1.11 `#define KBD_CCMDBYTE_INT2 0x02`
- 6.134.1.12 `#define KBD_CCMDBYTE_SYS 0x04`
- 6.134.1.13 `#define KBD_CCMDBYTE_XLAT 0x40`
- 6.134.1.14 `#define KBD_CTRL 4`
- 6.134.1.15 `#define KBD_DATA 0`
- 6.134.1.16 `#define KBD_KCMD_DK 0xF5`
- 6.134.1.17 `#define KBD_KCMD_ECHO 0xFF`
- 6.134.1.18 `#define KBD_KCMD_EK 0xF4`
- 6.134.1.19 `#define KBD_KCMD_RST 0xFF`
- 6.134.1.20 `#define KBD_KCMD_SRL 0xED`
- 6.134.1.21 `#define KBD_KRESP_ACK 0xFA`
- 6.134.1.22 `#define KBD_KRESP_ECHO 0xEE`
- 6.134.1.23 `#define KBD_KRESP_RSTOK 0xAA`
- 6.134.1.24 `#define KBD_MAX_BUF 0x100`
- 6.134.1.25 `#define KBD_SPACE 8`
- 6.134.1.26 `#define KBD_STATUS_A2 0x08`
- 6.134.1.27 `#define KBD_STATUS_IBF 0x02`

Generated on Tue Nov 11 15:03:20 2008 for Olskinn. The OpenRISC 1000 Architectural Simulator by Doxygen

6.134.1.28 `#define KBD_STATUS_INH 0x10`

6.134.1.29 `#define KBD_STATUS_MOBF 0x20`

6.134.1.30 `#define KBD_STATUS_OBF 0x01`

6.134.2.6 `static void kbd_put (struct kbd_state * kbd, unsigned char c)` [static]

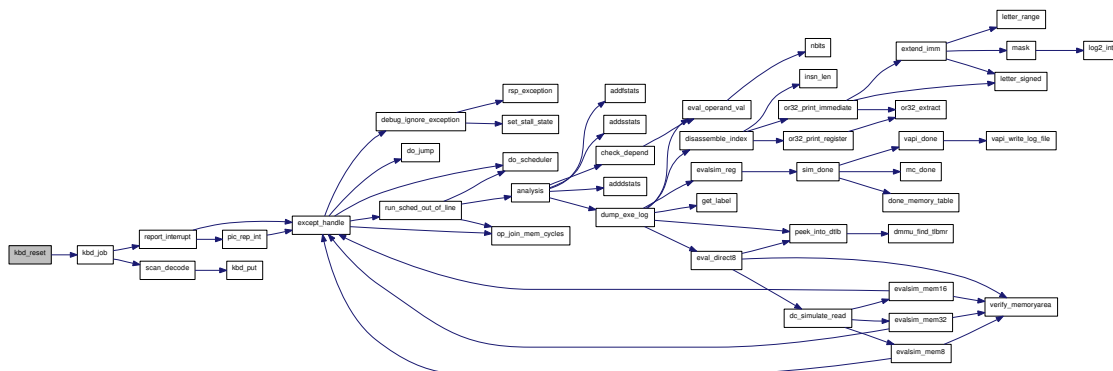
6.134.2.7 `static uint8_t kbd_read8 (oraddr_t addr, void * dat)` [static]

Here is the call graph for this function:



6.134.2.8 `static void kbd_reset (void * dat)` [static]

Here is the call graph for this function:



6.134.2.9 `static void kbd_rxfile (union param_val val, void * dat)` [static]

Set the keyboard input file

Free any previously allocated value.

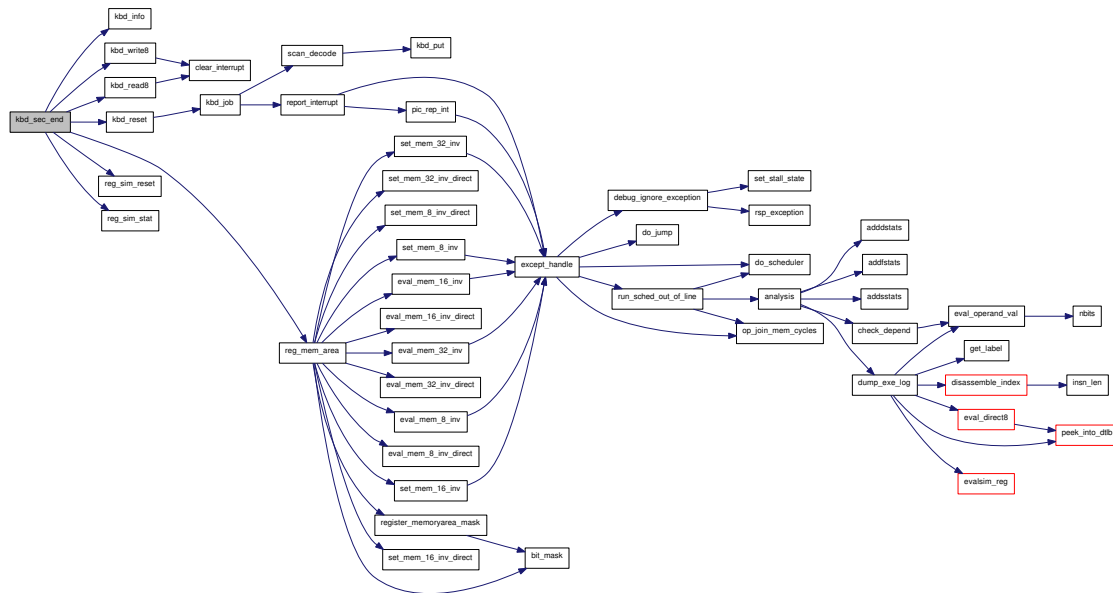
Parameters:

← *val* The value to use

← *dat* The `config` data structure

6.134.2.10 static void kbd_sec_end (void * dat) [static]

Here is the call graph for this function:



6.134.2.11 static void* kbd_sec_start () [static]

Initialize a new keyboard configuration

ALL parameters are set explicitly to default values.

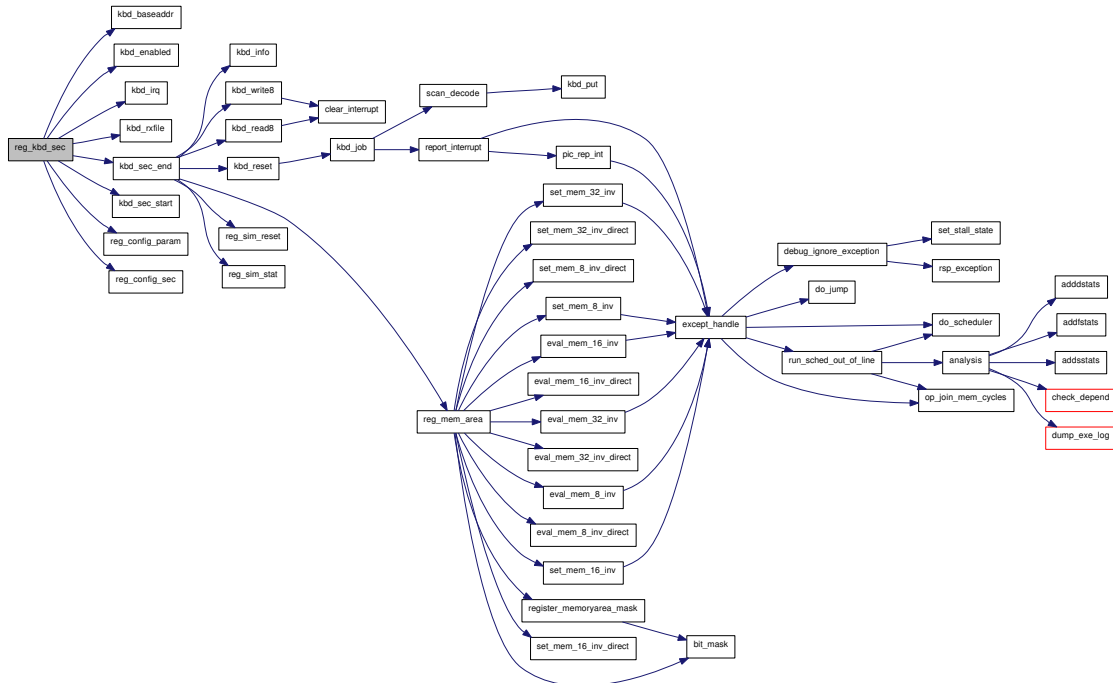
6.134.2.12 static void kbd_write8 (oraddr_t addr, uint8_t value, void * dat) [static]

Here is the call graph for this function:



6.134.2.13 void reg_kbd_sec ()

Here is the call graph for this function:



6.134.2.14 static void scan_decode (struct kbd_state * kbd, unsigned char c) [static]

Here is the call graph for this function:



6.134.3 Variable Documentation

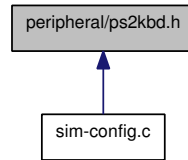
6.134.3.1 unsigned char code

6.134.3.2 const { ... } scan_table[128] [static]

6.134.3.3 unsigned char shift

6.135 peripheral/ps2kbd.h File Reference

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



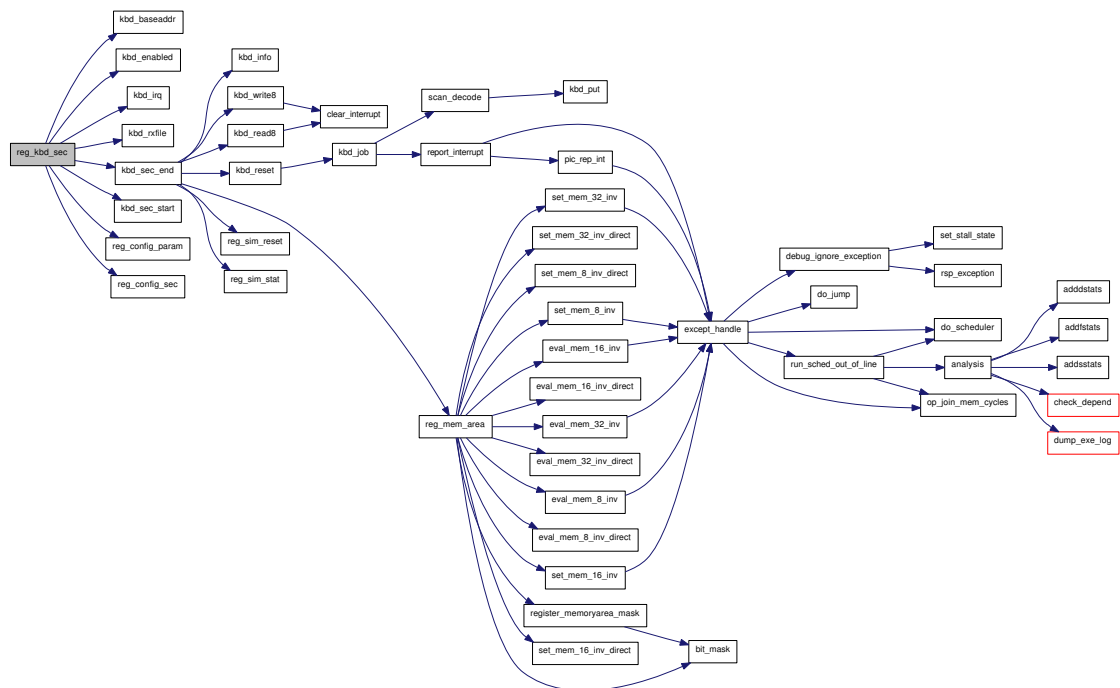
Functions

- void [reg_kbd_sec](#) ()

6.135.1 Function Documentation

6.135.1.1 void [reg_kbd_sec](#) ()

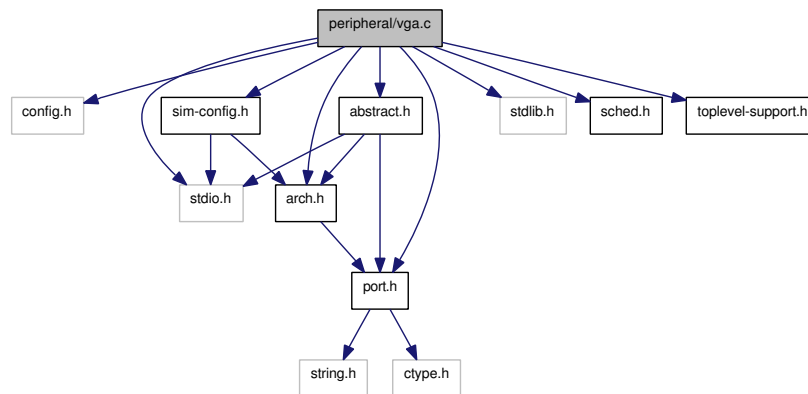
Here is the call graph for this function:



6.136 peripheral/vga.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include "arch.h"
#include "sim-config.h"
#include "abstract.h"
#include "sched.h"
#include "toplevel-support.h"
```

Include dependency graph for vga.c:



Data Structures

- struct [vga_state](#)
- struct [BMP_HEADER](#)
- struct [INFOHEADER](#)

Defines

- #define [VGA_CTRL](#) 0x00
- #define [VGA_STAT](#) 0x04
- #define [VGA_HTIM](#) 0x08
- #define [VGA_VTIM](#) 0x0c
- #define [VGA_HVLEN](#) 0x10
- #define [VGA_VBARA](#) 0x14
- #define [VGA_VBARB](#) 0x18
- #define [VGA_CLUTA](#) 0x800
- #define [VGA_CLUTB](#) 0xc00
- #define [VGA_MASK](#) 0xff
- #define [VGA_ADDR_SPACE](#) 1024

- #define `VGA_CTRL_VEN` 0x00000001
- #define `VGA_CTRL_CD` 0x00000300
- #define `VGA_CTRL_PC` 0x00000400

Functions

- void `vga_write32` (`oraddr_t` addr, `uint32_t` value, void *dat)
- `uint32_t` `vga_read32` (`oraddr_t` addr, void *dat)
- static int `vga_dump_image` (char *filename, struct `vga_state` *vga)
- void `vga_job` (void *dat)
- void `vga_reset` (void *dat)
- static void `vga_enabled` (union `param_val` val, void *dat)
- static void `vga_baseaddr` (union `param_val` val, void *dat)
- static void `vga_irq` (union `param_val` val, void *dat)
- static void `vga_refresh_rate` (union `param_val` val, void *dat)
- static void `vga_filename` (union `param_val` val, void *dat)
- static void * `vga_sec_start` ()
- static void `vga_sec_end` (void *dat)
- void `reg_vga_sec` ()

6.136.1 Define Documentation

6.136.1.1 **#define VGA_ADDR_SPACE 1024**

6.136.1.2 **#define VGA_CLUTA 0x800**

6.136.1.3 **#define VGA_CLUTB 0xc00**

6.136.1.4 **#define VGA_CTRL 0x00**

6.136.1.5 **#define VGA_CTRL_CD 0x00000300**

6.136.1.6 **#define VGA_CTRL_PC 0x00000400**

6.136.1.7 **#define VGA_CTRL_VEN 0x00000001**

6.136.1.8 **#define VGA_HTIM 0x08**

6.136.1.9 **#define VGA_HVLEN 0x10**

6.136.1.10 **#define VGA_MASK 0xff**

6.136.1.11 **#define VGA_STAT 0x04**

6.136.1.12 **#define VGA_VBARA 0x14**

6.136.1.13 **#define VGA_VBARB 0x18**

6.136.1.14 **#define VGA_VTIM 0x0c**

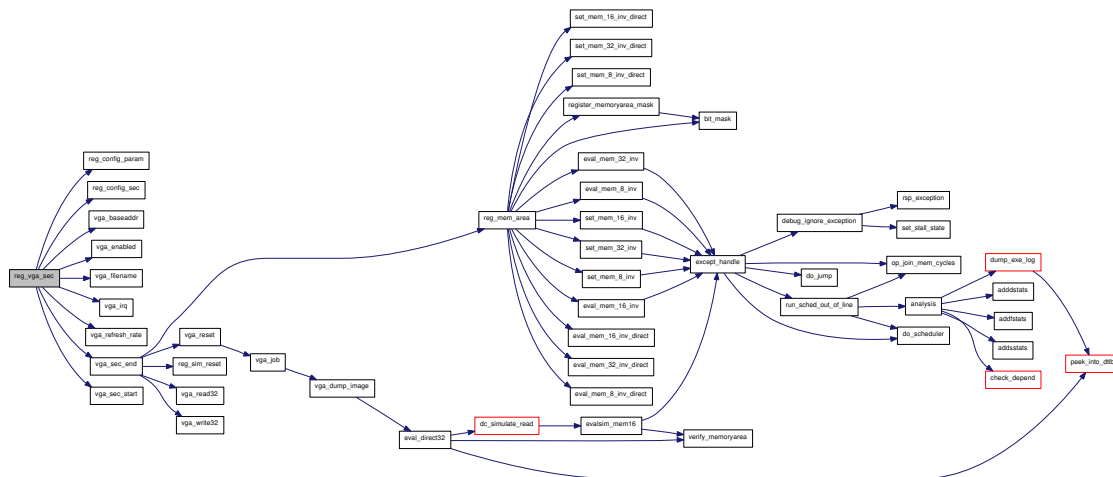
6.136.2 Function Documentation

6.136.2.1 **void reg_vga_sec ()**

Create a new VGA configuration

ALL parameters are set explicitly to default values. Alternative naming for file parameter supported.

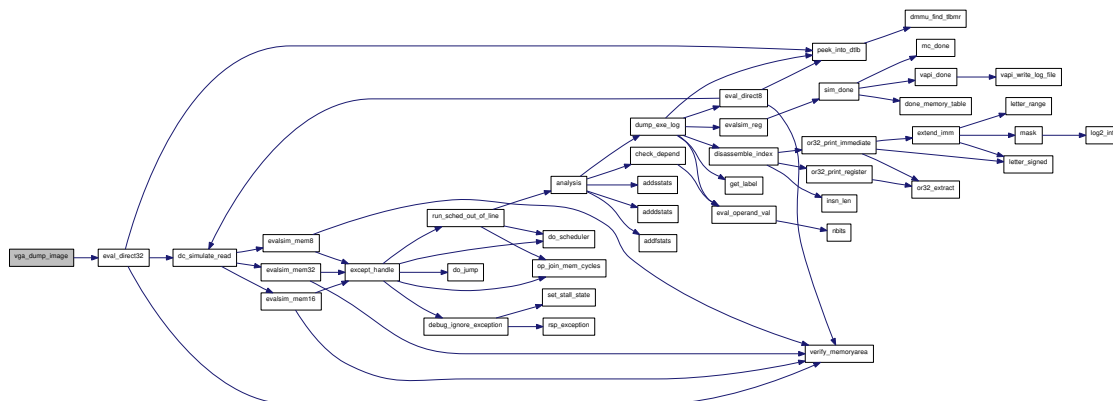
Here is the call graph for this function:



6.136.2.2 `static void vga_baseaddr (union param_val val, void * dat) [static]`

6.136.2.3 `static int vga_dump_image (char * filename, struct vga_state * vga) [static]`

Here is the call graph for this function:



6.136.2.4 `static void vga_enabled (union param_val val, void * dat) [static]`

6.136.2.5 `static void vga_filename (union param_val val, void * dat) [static]`

Set the VGA output file

Free any previously allocated value.

Parameters:

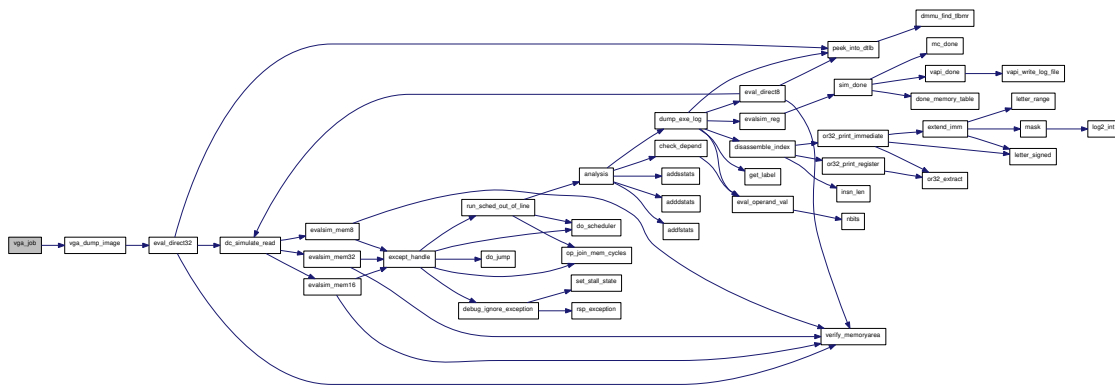
← *val* The value to use

← *dat* The `config` data structure

6.136.2.6 static void vga_irq (union param_val val, void * dat) [static]

6.136.2.7 void vga_job (void * dat)

Here is the call graph for this function:

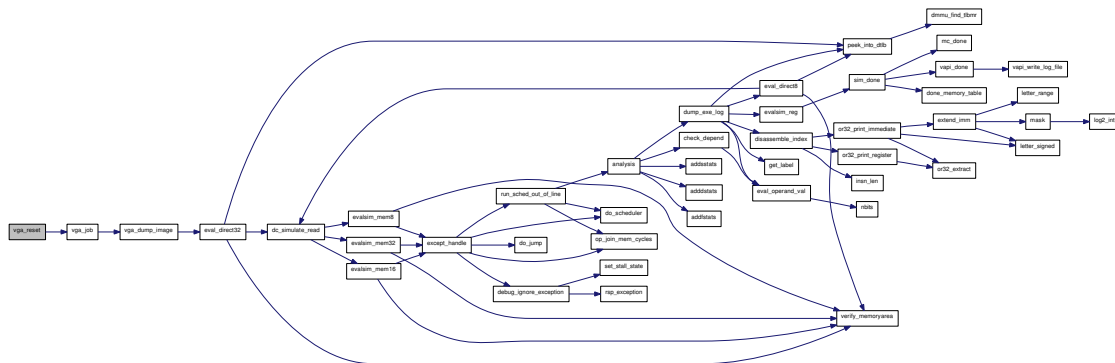


6.136.2.8 uint32_t vga_read32 (oraddr_t addr, void * dat)

6.136.2.9 static void vga_refresh_rate (union param_val val, void * dat) [static]

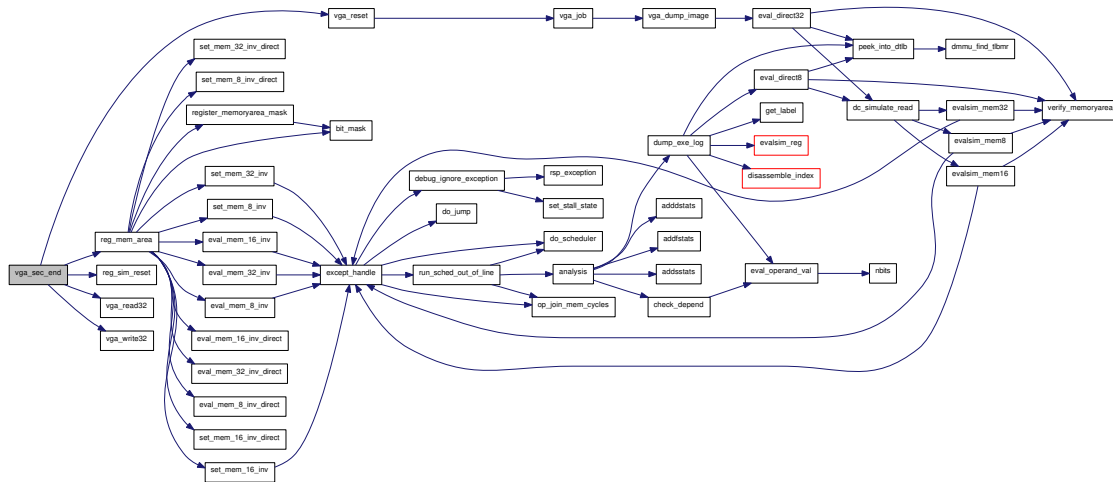
6.136.2.10 void vga_reset (void * dat)

Here is the call graph for this function:



6.136.2.11 static void vga_sec_end (void * dat) [static]

Here is the call graph for this function:



6.136.2.12 static void* vga_sec_start () [static]

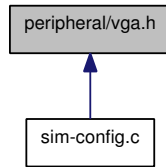
Initialize a new VGA configuration

ALL parameters are set explicitly to default values.

6.136.2.13 void vga_write32 (oraddr_t addr, uint32_t value, void * dat)

6.137 peripheral/vga.h File Reference

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



Functions

- void [reg_vga_sec](#) ()

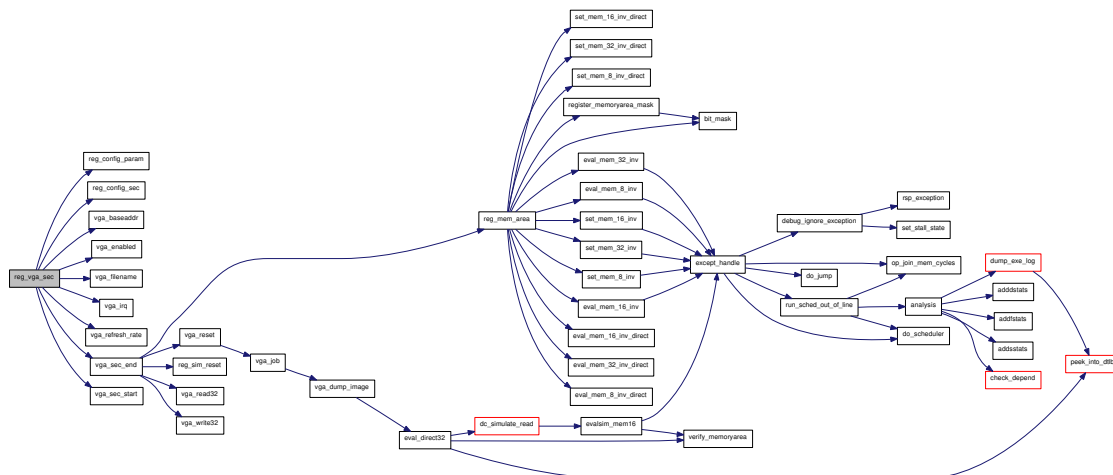
6.137.1 Function Documentation

6.137.1.1 void [reg_vga_sec](#) ()

Create a new VGA configuration

ALL parameters are set explicitly to default values. Alternative naming for file parameter supported.

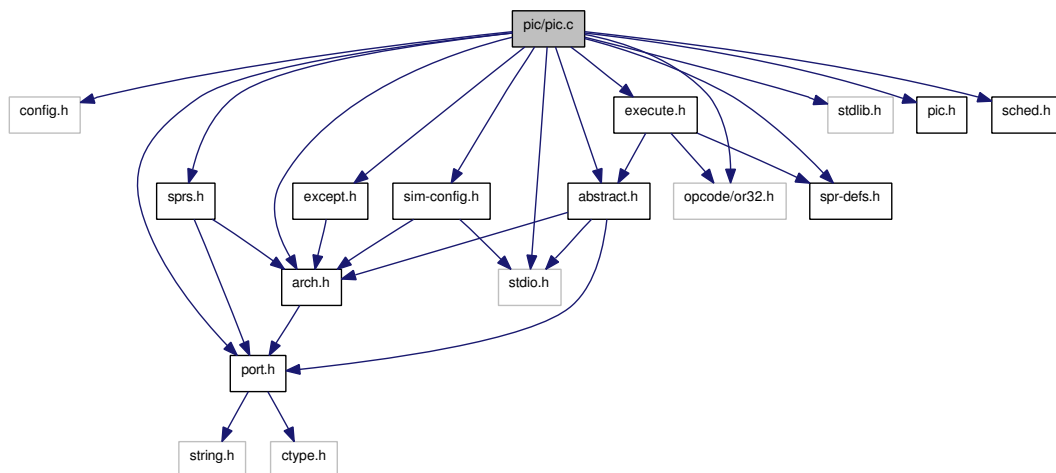
Here is the call graph for this function:



6.138 pic/pic.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include "arch.h"
#include "abstract.h"
#include "pic.h"
#include "opcode/or32.h"
#include "spr-defs.h"
#include "execute.h"
#include "except.h"
#include "sprs.h"
#include "sim-config.h"
#include "sched.h"
```

Include dependency graph for pic.c:



Functions

- void [pic_reset](#) (void)
- static void [pic_rep_int](#) (void *dat)
- void [pic_ints_en](#) (void)
- void [report_interrupt](#) (int line)
- void [clear_interrupt](#) (int line)
- static void [pic_enabled](#) (union [param_val](#) val, void *dat)
- static void [pic_edge_trigger](#) (union [param_val](#) val, void *dat)
- void [reg_pic_sec](#) ()

Variables

- struct pic `pic_state_int` = { 1, 1 }
- struct pic * `pic_state` = &`pic_state_int`

6.138.1 Function Documentation

6.138.1.1 void `clear_interrupt` (int *line*)

6.138.1.2 static void `pic_edge_trigger` (union `param_val val`, void * *dat*) [static]

Enable or disable edge triggering of interrupts

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure (not used here)

6.138.1.3 static void `pic_enabled` (union `param_val val`, void * *dat*) [static]

Enable or disable the programmable interrupt controller

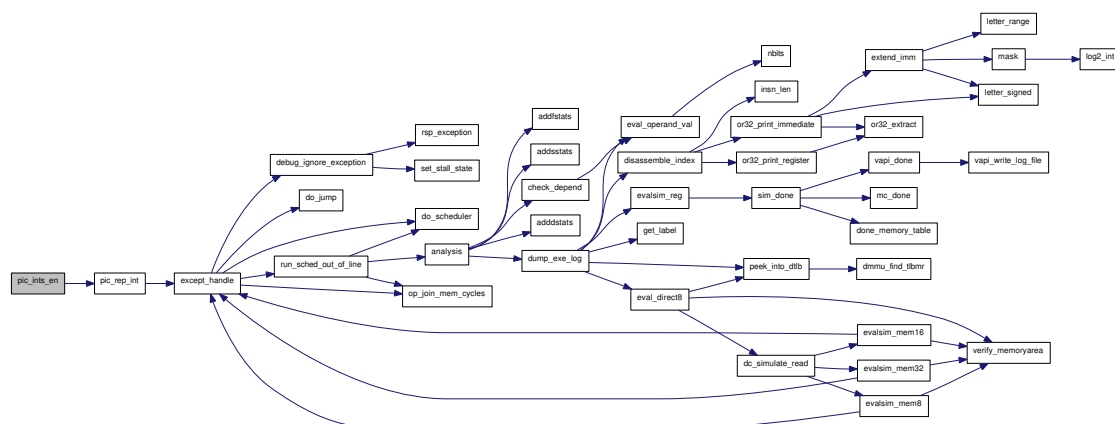
Set the corresponding field in the UPR

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure (not used here)

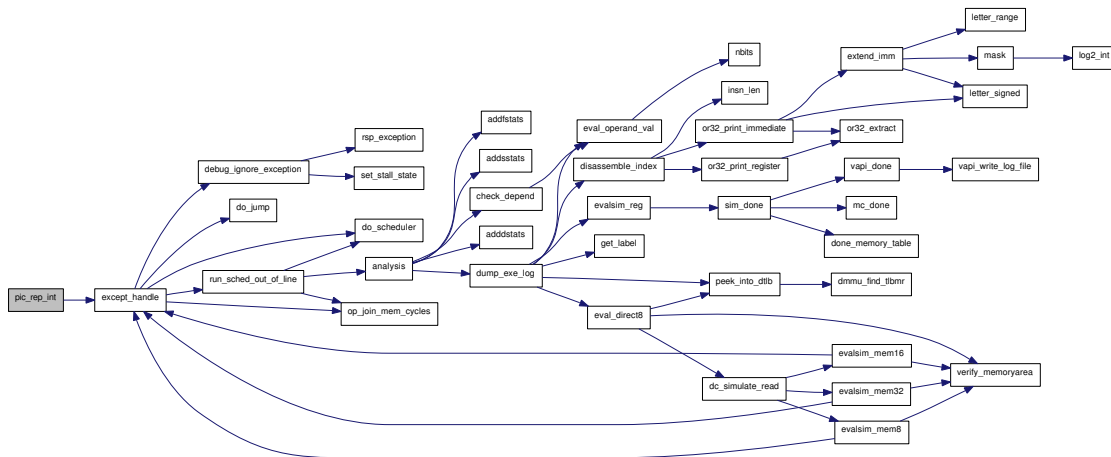
6.138.1.4 void `pic_ints_en` (void)

Here is the call graph for this function:



6.138.1.5 static void pic_rep_int (void * dat) [static]

Here is the call graph for this function:



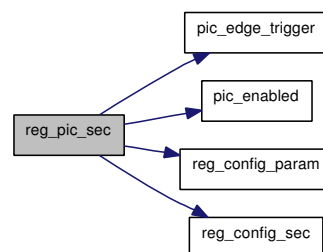
6.138.1.6 void pic_reset (void)

6.138.1.7 void reg_pic_sec ()

Initialize a new interrupt controller configuration

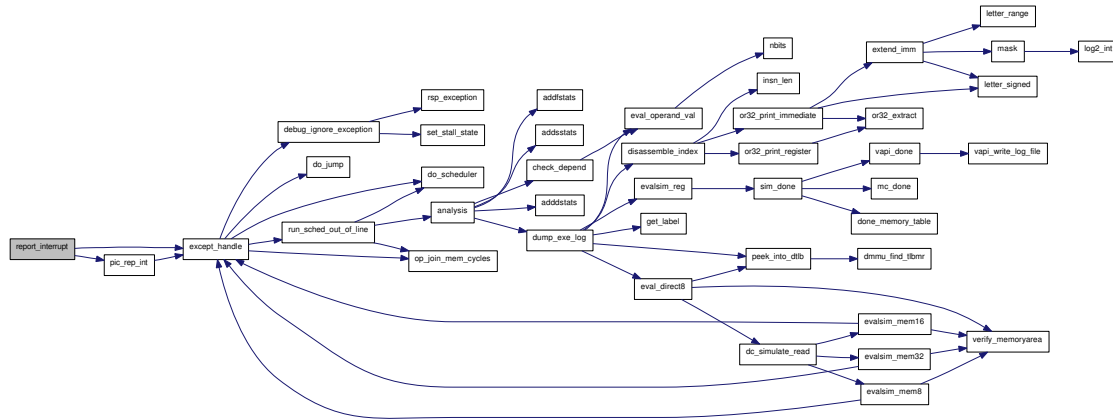
ALL parameters are set explicitly to default values in [init_defconfig\(\)](#)

Here is the call graph for this function:



6.138.1.8 void report_interrupt (int line)

Here is the call graph for this function:



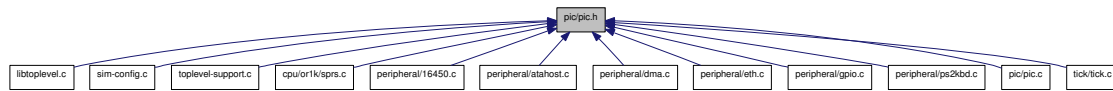
6.138.2 Variable Documentation

6.138.2.1 struct pic* pic_state = &pic_state_int

6.138.2.2 struct pic pic_state_int = { 1, 1 }

6.139 pic/pic.h File Reference

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



Functions

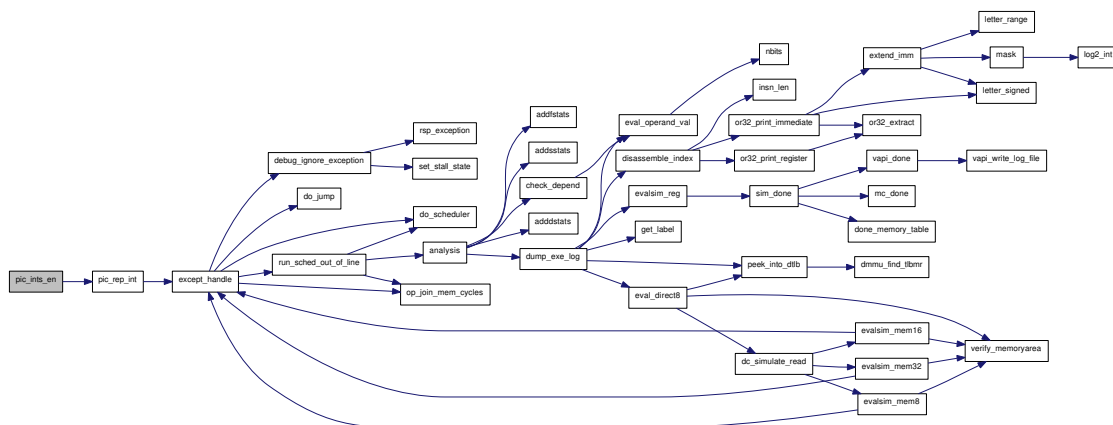
- void [pic_reset](#) ()
- void [report_interrupt](#) (int line)
- void [clear_interrupt](#) (int line)
- void [pic_ints_en](#) ()
- void [reg_pic_sec](#) ()

6.139.1 Function Documentation

6.139.1.1 void [clear_interrupt](#) (int *line*)

6.139.1.2 void [pic_ints_en](#) ()

Here is the call graph for this function:



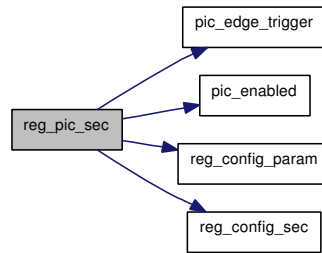
6.139.1.3 void [pic_reset](#) ()

6.139.1.4 void [reg_pic_sec](#) ()

Initialize a new interrupt controller configuration

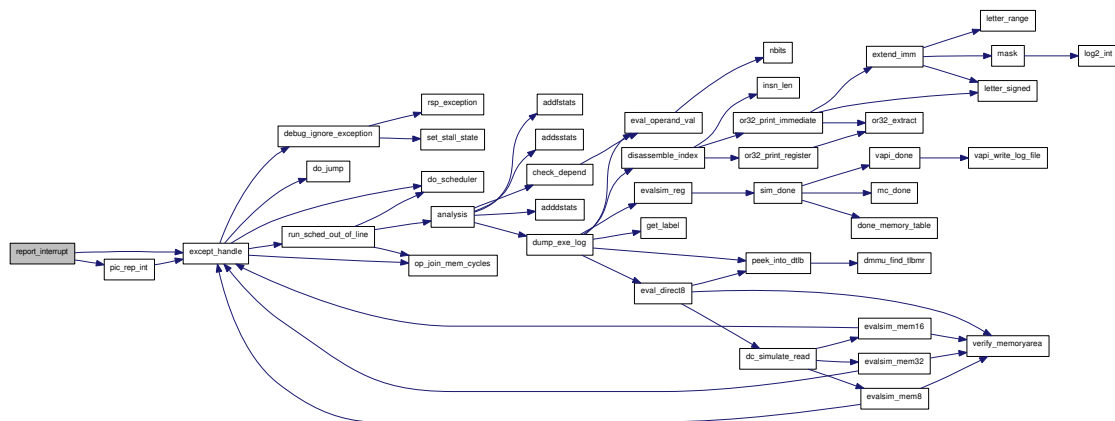
ALL parameters are set explicitly to default values in [init_defconfig\(\)](#)

Here is the call graph for this function:



6.139.1.5 void report_interrupt (int line)

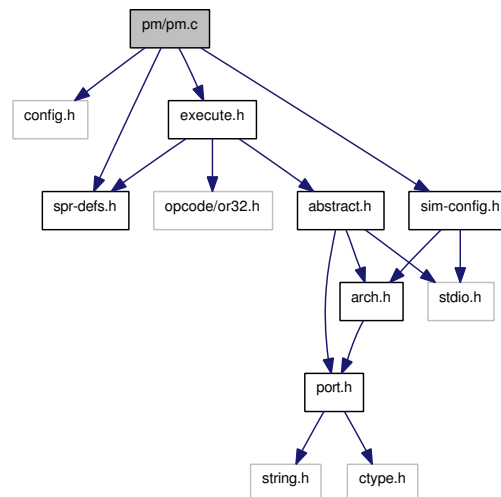
Here is the call graph for this function:



6.140 pm/pm.c File Reference

```
#include "config.h"
#include "spr-defs.h"
#include "execute.h"
#include "sim-config.h"
```

Include dependency graph for pm.c:



Functions

- void [pm_reset](#) ()
- static void [pm_enabled](#) (union [param_val](#) val, void *dat)
- void [reg_pm_sec](#) ()

6.140.1 Function Documentation

6.140.1.1 static void [pm_enabled](#) (union [param_val](#) val, void *dat) [static]

Enable or disable power management

Set the corresponding field in the UPR

Parameters:

- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

6.140.1.2 void [pm_reset](#) ()

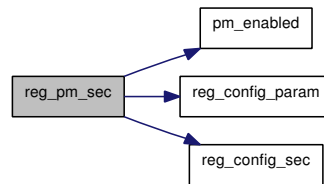
Reset power management

Initializes PMR register by clearing it.

6.140.1.3 void reg_pm_sec ()

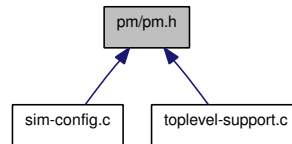
Set up a new power management configuration section

Here is the call graph for this function:



6.141 pm/pm.h File Reference

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



Functions

- void [pm_reset \(\)](#)
- void [reg_pm_sec \(\)](#)

6.141.1 Function Documentation

6.141.1.1 void pm_reset ()

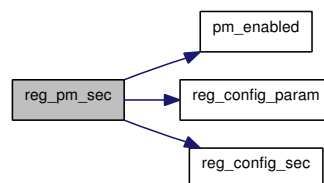
Reset power management

Initializes PMR register by clearing it.

6.141.1.2 void reg_pm_sec ()

Set up a new power management configuration section

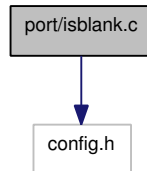
Here is the call graph for this function:



6.142 port/isblank.c File Reference

```
#include "config.h"
```

Include dependency graph for isblank.c:



Functions

- int [isblank](#) (int *c*)

6.142.1 Function Documentation

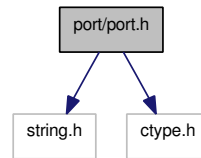
6.142.1.1 int isblank (int *c*)

6.143 port/port.h File Reference

```
#include <string.h>
```

```
#include <ctype.h>
```

Include dependency graph for port.h:



Defines

- #define [PRIx16](#) "x"
- #define [PRIx8](#) "x"

Functions

- char * [strndup](#) (const char *s, size_t n)
- int [isblank](#) (int c)

6.143.1 Define Documentation

6.143.1.1 #define [PRIx16](#) "x"

6.143.1.2 #define [PRIx8](#) "x"

6.143.2 Function Documentation

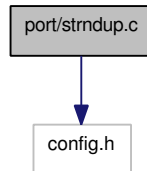
6.143.2.1 int [isblank](#) (int *c*)

6.143.2.2 char* [strndup](#) (const char * *s*, size_t *n*)

6.144 port/strndup.c File Reference

```
#include "config.h"
```

Include dependency graph for strndup.c:



Functions

- char * [strndup](#) (const char *s, size_t n)

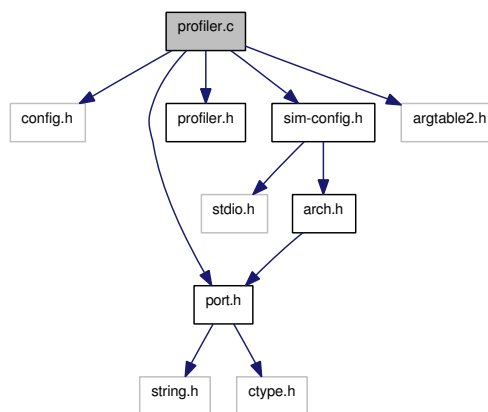
6.144.1 Function Documentation

6.144.1.1 char* strndup (const char * s, size_t n)

6.145 profiler.c File Reference

```
#include "config.h"
#include "port.h"
#include "profiler.h"
#include "sim-config.h"
#include "argtable2.h"
```

Include dependency graph for profiler.c:



Data Structures

- struct [stack_struct](#)

Defines

- #define [MAX_STACK](#) 1024

Functions

- int [prof_acquire](#) (const char *fprofname)
- static void [prof_print](#) ()
- void [prof_set](#) (int _quiet, int _cumulative)
- int [main_profiler](#) (int argc, char *argv[], int just_help)

Variables

- struct [func_struct](#) [prof_func](#) [MAX_FUNCS]
- int [prof_nfuncs](#) = 0
- int [prof_cycles](#) = 0
- static struct [stack_struct](#) [stack](#) [MAX_STACK]
- static int [nstack](#) = 0
- static int [maxstack](#) = 0

- static int `ntotcalls` = 0
- static int `nfuncalls` = 0
- static int `cumulative` = 0
- static int `quiet` = 0
- static FILE * `fprof` = 0

6.145.1 Define Documentation

6.145.1.1 #define MAX_STACK 1024

Maximum stack frames that can be profiled

6.145.2 Function Documentation

6.145.2.1 int main_profiler (int argc, char * argv[], int just_help)

Parse the arguments for the profiling utility

Updated by Jeremy Bennett to use argtable2. Also has an option just to print help, for use with the CLI.

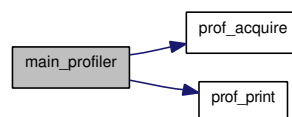
Parameters:

- ← *argc* Number of command args
- ← *argv* Vector of the command args
- ← *just_help* If 1 (true), ignore argc & argv and just print out the help message without parsing args

Returns:

0 on success, 1 on failure

Here is the call graph for this function:



6.145.2.2 int prof_acquire (const char * fprofname)

Acquire data from profiler file

Parameters:

- ← *fprofname* Data file to analyse

Returns:

0 on success, return code otherwise

6.145.2.3 `static void prof_print ()` [static]

6.145.2.4 `void prof_set (int _quiet, int _cumulative)`

6.145.3 Variable Documentation

6.145.3.1 `int cumulative = 0` [static]

Whether we are in cumulative mode

6.145.3.2 `FILE* fprof = 0` [static]

File to read from

6.145.3.3 `int maxstack = 0` [static]

Max depth

6.145.3.4 `int nfuncalls = 0` [static]

Number of covered calls

6.145.3.5 `int nstack = 0` [static]

Current depth

6.145.3.6 `int ntotcalls = 0` [static]

Number of total calls

6.145.3.7 `int prof_cycles = 0`

Global: current cycles

6.145.3.8 `struct func_struct prof_func[MAX_FUNCS]`

Global: data about functions

6.145.3.9 `int prof_nfuncs = 0`

Global: total number of functions

6.145.3.10 `int quiet = 0` [static]

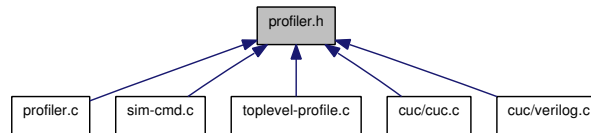
Whether we should not report warnings

6.145.3.11 `struct stack_struct stack[MAX_STACK]` `[static]`

Representation of the stack

6.146 profiler.h File Reference

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



Data Structures

- struct [func_struct](#)

Defines

- #define [MAX_FUNCS](#) 1024

Functions

- int [prof_acquire](#) (const char *fprofname)
- void [prof_set](#) (int _quiet, int _cumulative)
- int [main_profiler](#) (int argc, char *argv[], int just_help)

Variables

- struct [func_struct](#) [prof_func](#) [MAX_FUNCS]
- int [prof_nfuncs](#)
- int [prof_cycles](#)

6.146.1 Define Documentation

6.146.1.1 #define MAX_FUNCS 1024

Maximum number of functions that can be profiled

6.146.2 Function Documentation

6.146.2.1 int main_profiler (int *argc*, char * *argv*[], int *just_help*)

Parse the arguments for the profiling utility

Updated by Jeremy Bennett to use argtable2. Also has an option just to print help, for use with the CLI.

Parameters:

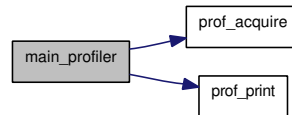
- ← *argc* Number of command args
- ← *argv* Vector of the command args

← *just_help* If 1 (true), ignore argc & argv and just print out the help message without parsing args

Returns:

0 on success, 1 on failure

Here is the call graph for this function:

**6.146.2.2 int prof_acquire (const char * *fprofname*)**

Acquire data from profiler file

Parameters:

← *fprofname* Data file to analyse

Returns:

0 on success, return code otherwise

6.146.2.3 void prof_set (int *_quiet*, int *_cumulative*)**6.146.3 Variable Documentation****6.146.3.1 int prof_cycles**

Global: current cycles

6.146.3.2 struct func_struct prof_func[MAX_FUNCS]

Global: data about functions

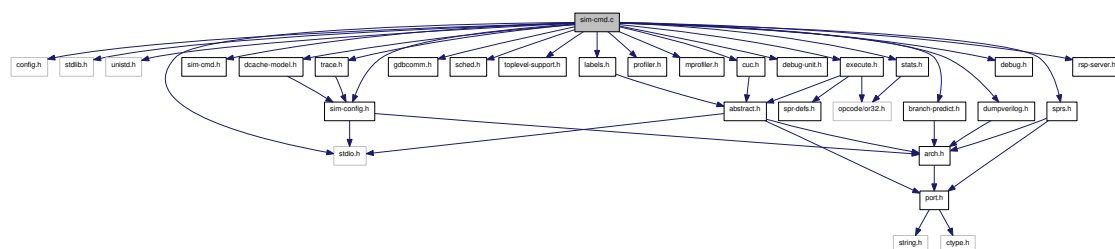
6.146.3.3 int prof_nfuncs

Global: total number of functions

6.147 sim-cmd.c File Reference

```
#include "config.h"
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include "sim-cmd.h"
#include "sim-config.h"
#include "execute.h"
#include "labels.h"
#include "gdbcomm.h"
#include "sched.h"
#include "toplevel-support.h"
#include "dumpverilog.h"
#include "profiler.h"
#include "mprofiler.h"
#include "trace.h"
#include "debug-unit.h"
#include "stats.h"
#include "sprs.h"
#include "dcache-model.h"
#include "branch-predict.h"
#include "debug.h"
#include "cuc.h"
#include "rsp-server.h"
```

Include dependency graph for sim-cmd.c:



Data Structures

- struct [sim_stat](#)
- struct [sim_command](#)

Functions

- void `reg_sim_stat` (void(*stat_func)(void *dat), void *dat)
- void `reenter_int` (void *dat)
- static int `sim_cmd_quit` (int argc, char **argv)
- static int `sim_cmd_help` (int argc, char **argv)
- static int `sim_cmd_trace` (int argc, char **argv)
- static int `sim_cmd_dm` (int argc, char **argv)
- static int `sim_cmd_dv` (int argc, char **argv)
- static int `sim_cmd_dh` (int argc, char **argv)
- static int `sim_cmd_pm` (int argc, char **argv)
- static int `sim_cmd_cm` (int argc, char **argv)
- static int `sim_cmd_pr` (int argc, char **argv)
- static int `sim_cmd_pc` (int argc, char **argv)
- static int `sim_cmd_breaks` (int argc, char **argv)
- static int `sim_cmd_break` (int argc, char **argv)
- static int `sim_cmd_r` (int argc, char **argv)
- static int `sim_cmd_de` (int argc, char **argv)
- static int `sim_cmd_reset` (int argc, char **argv)
- static int `sim_cmd_hist` (int argc, char **argv)
- void `check_insn_exec` (void *dat)
- void `print_insn_exec` (void *dat)
- static int `sim_cmd_run` (int argc, char **argv)
- static int `sim_cmd_stall` (int argc, char **argv)
- static int `sim_cmd_unstall` (int argc, char **argv)
- static int `sim_cmd_stats` (int argc, char **argv)
- static int `sim_cmd_info` (int argc, char **argv)
- static int `sim_cmd_setdbch` (int argc, char **argv)
- static int `sim_cmd_debug` (int argc, char **argv)
- static int `sim_cmd_profile` (int argc, char **argv)
- static int `sim_cmd_mprofile` (int argc, char **argv)
- static int `sim_cmd_cuc` (int argc, char **argv)
- static int `sim_cmd_set` (int argc, char **argv)
- static char * `strip_space` (char *str)
- void `handle_sim_command` (void)

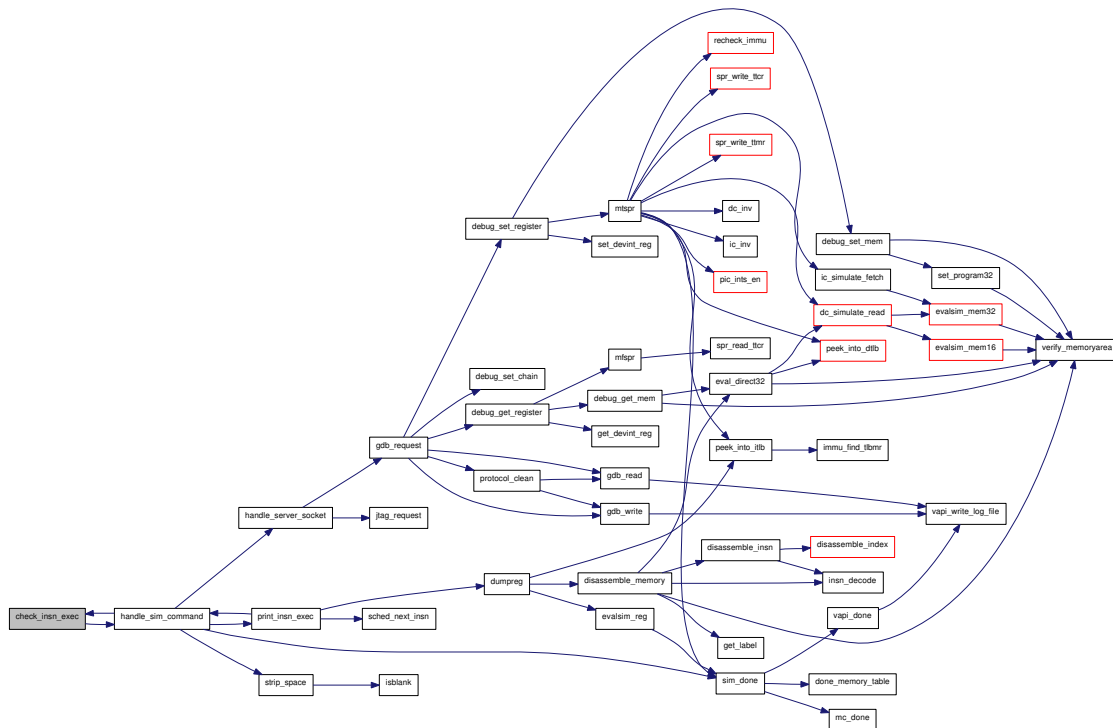
Variables

- static long long `to_insn_num`
- static struct `sim_stat` * `sim_stats` = NULL
- static struct `sim_command` `sim_commands` []

6.147.1 Function Documentation

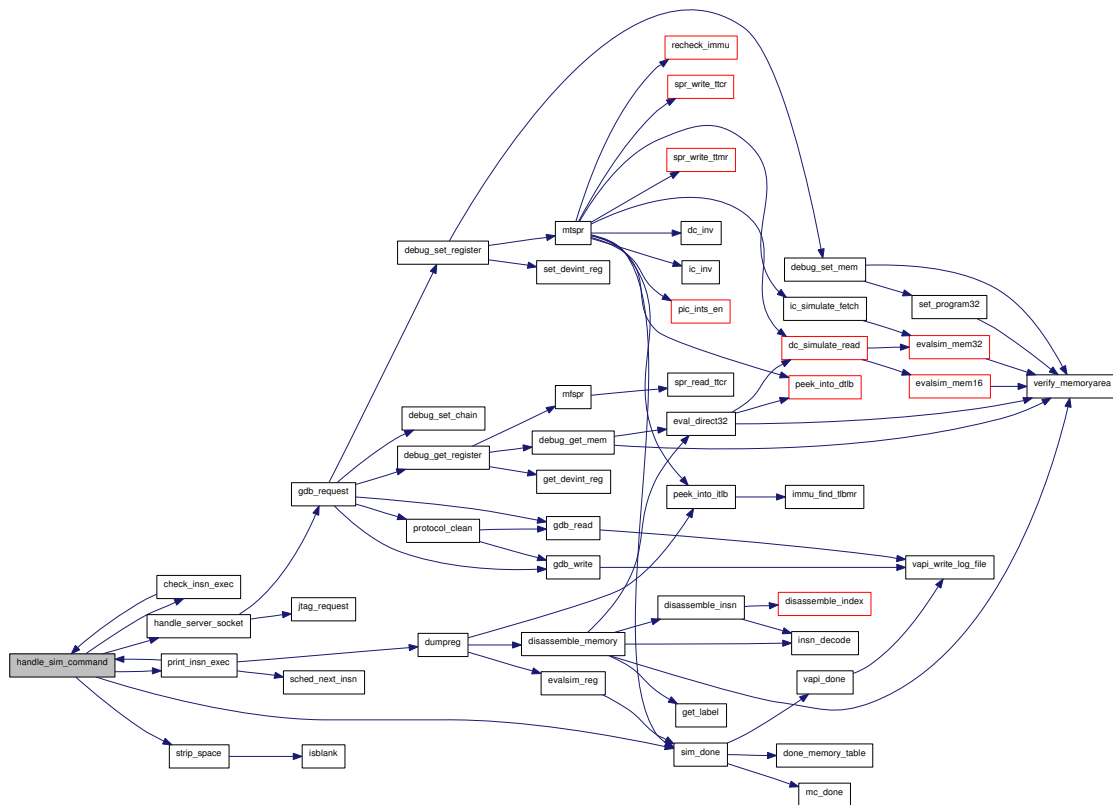
6.147.1.1 void check_insn_exec (void * dat)

Here is the call graph for this function:



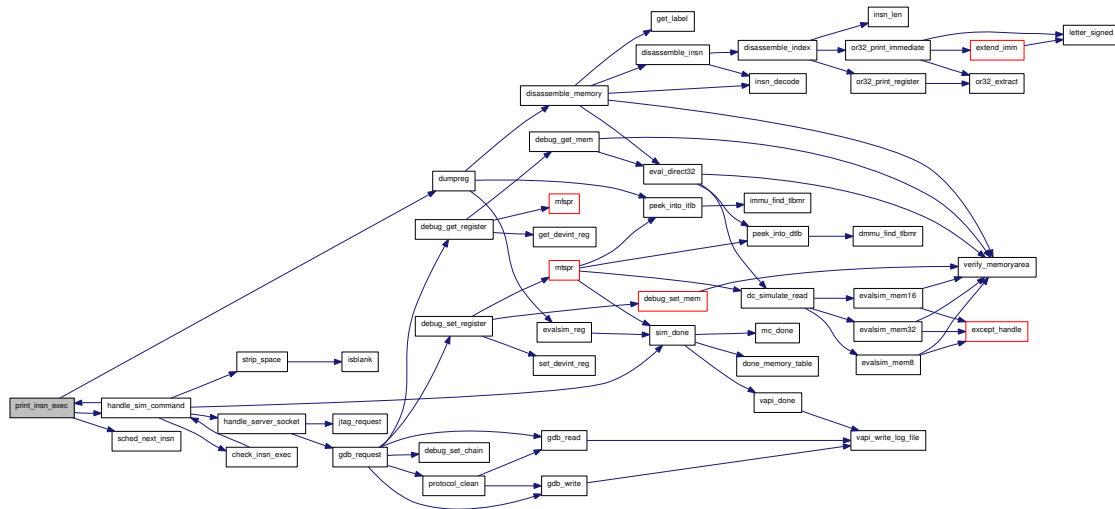
6.147.1.2 void handle_sim_command (void)

Here is the call graph for this function:



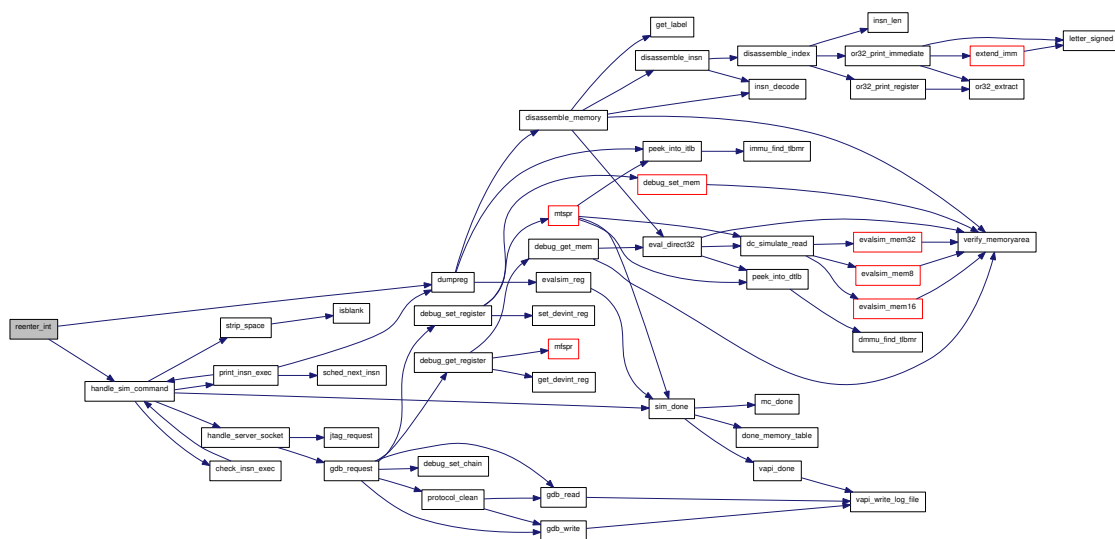
6.147.1.3 void print_insn_exec (void * dat)

Here is the call graph for this function:



6.147.1.4 void reenter_int (void * dat)

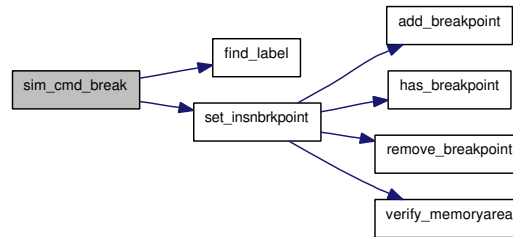
Here is the call graph for this function:



6.147.1.5 void reg_sim_stat (void*)(void *dat) stat_func, void * dat)

6.147.1.6 static int sim_cmd_break (int argc, char ** argv) [static]

Here is the call graph for this function:



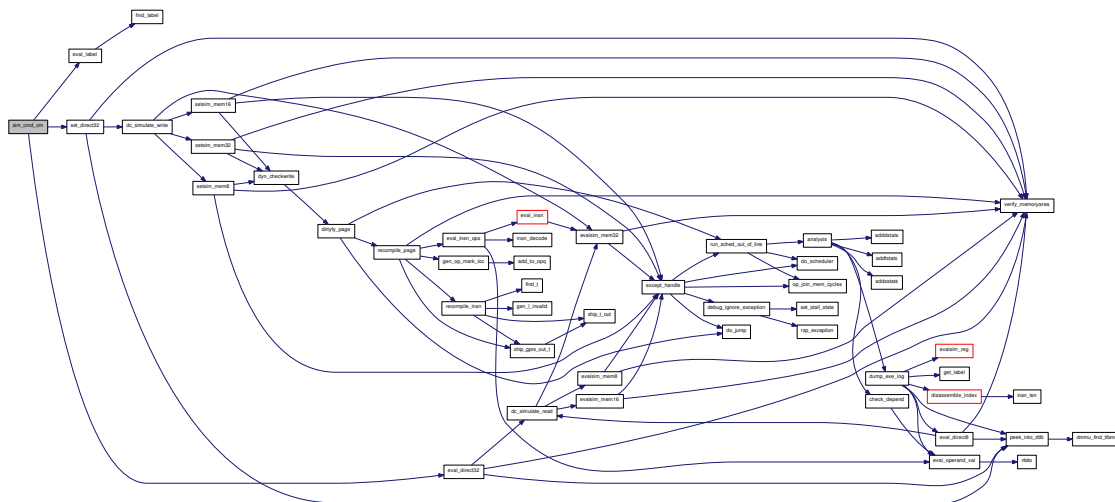
6.147.1.7 static int sim_cmd_breaks (int argc, char ** argv) [static]

Here is the call graph for this function:



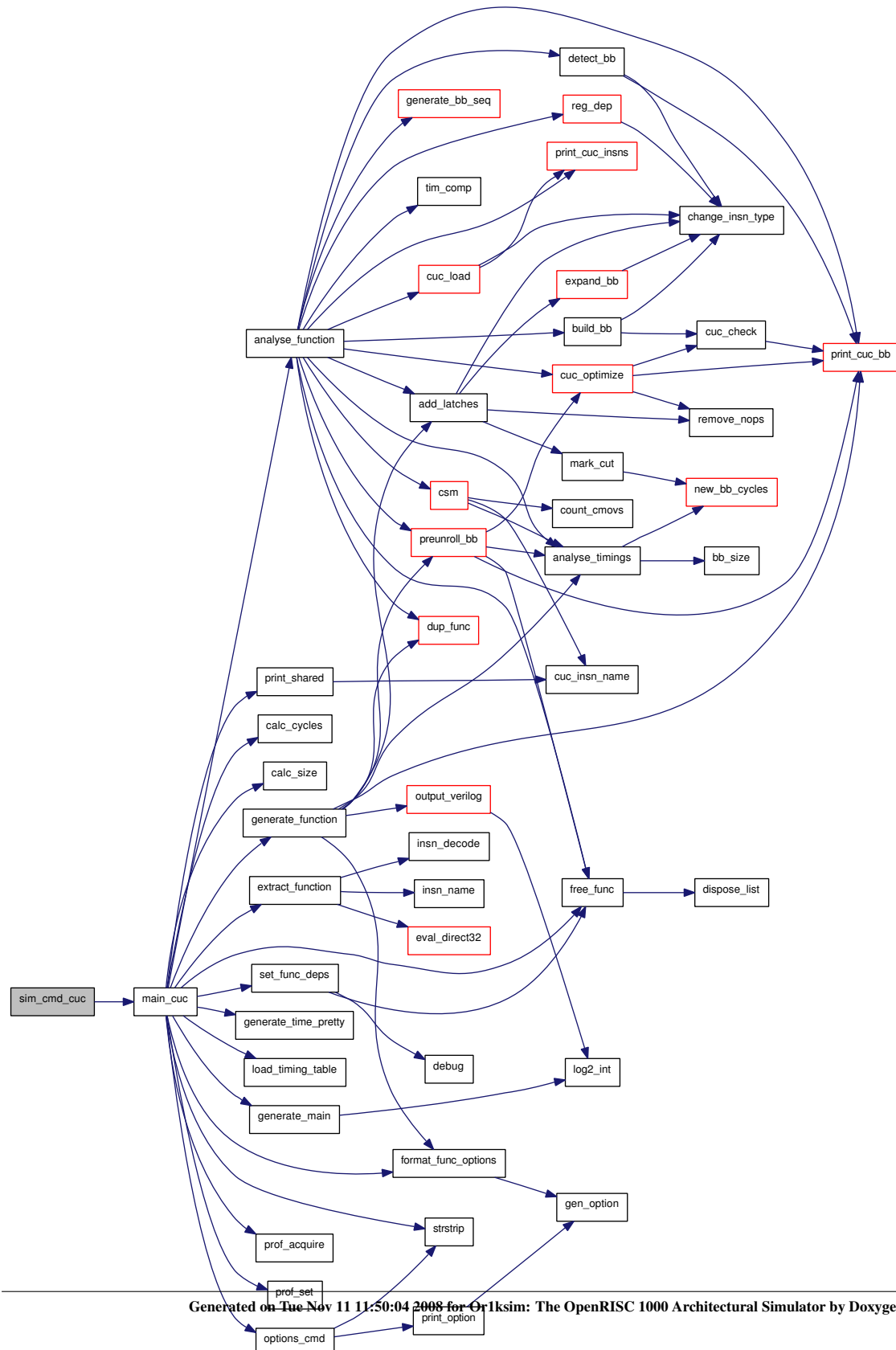
6.147.1.8 static int sim_cmd_cm (int argc, char ** argv) [static]

Here is the call graph for this function:



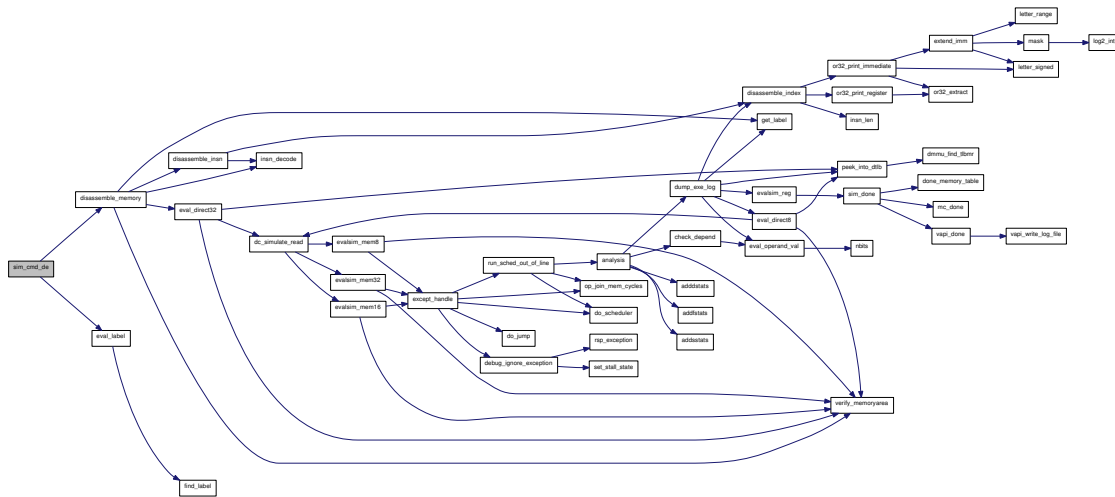
6.147.1.9 static int sim_cmd_cuc (int argc, char ** argv) [static]

Here is the call graph for this function:



6.147.1.10 static int sim_cmd_de (int argc, char ** argv) [static]

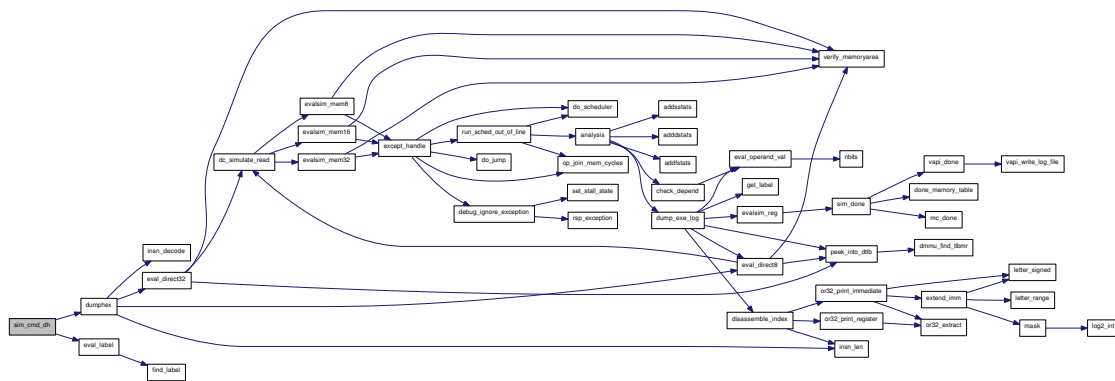
Here is the call graph for this function:



6.147.1.11 static int sim_cmd_debug (int argc, char ** argv) [static]

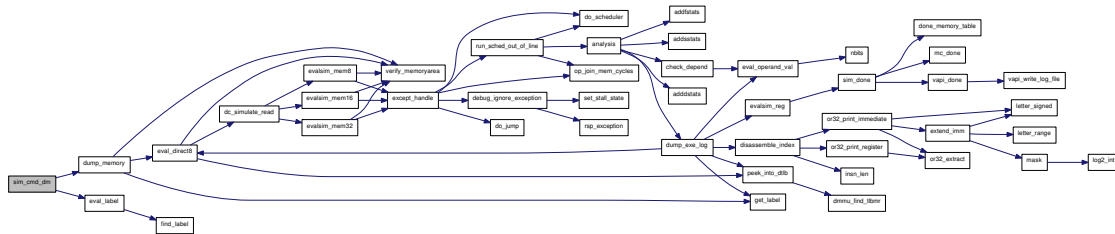
6.147.1.12 static int sim_cmd_dh (int argc, char ** argv) [static]

Here is the call graph for this function:



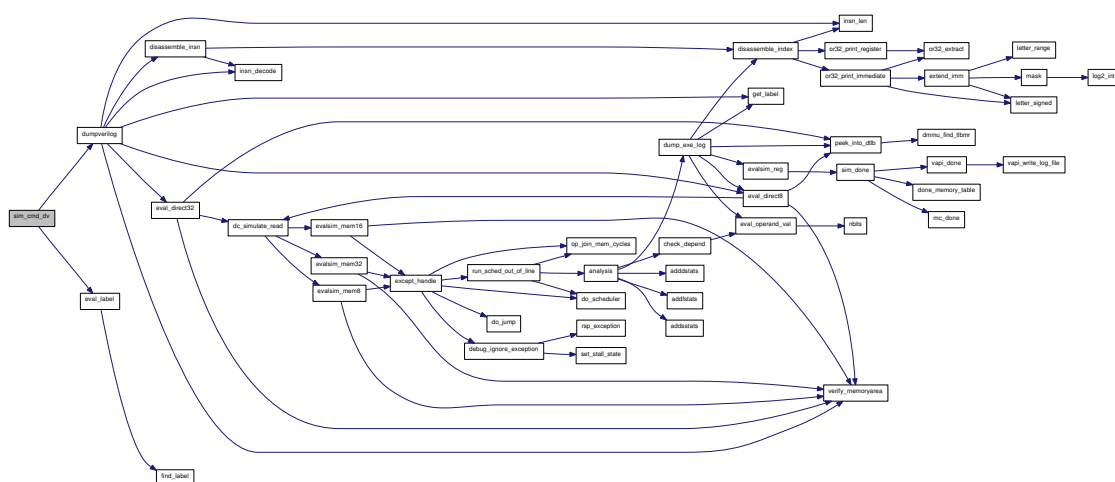
6.147.1.13 static int sim_cmd_dm (int argc, char ** argv) [static]

Here is the call graph for this function:



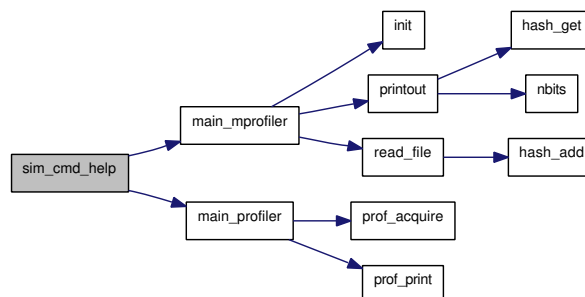
6.147.1.14 static int sim_cmd_dv (int argc, char ** argv) [static]

Here is the call graph for this function:



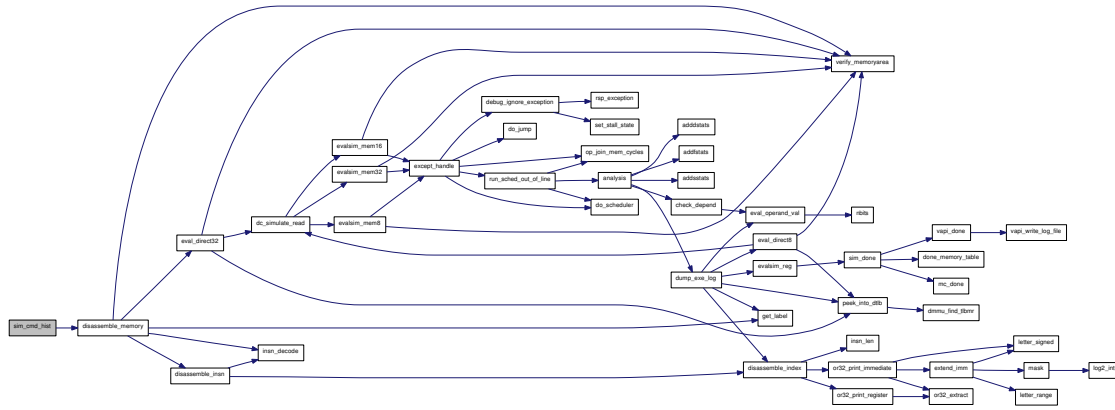
6.147.1.15 static int sim_cmd_help (int argc, char ** argv) [static]

Here is the call graph for this function:



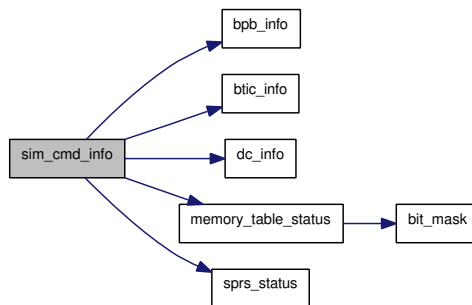
6.147.1.16 static int sim_cmd_hist (int argc, char ** argv) [static]

Here is the call graph for this function:



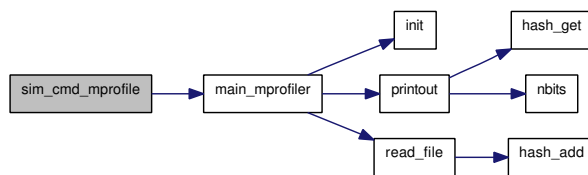
6.147.1.17 static int sim_cmd_info (int argc, char ** argv) [static]

Here is the call graph for this function:



6.147.1.18 static int sim_cmd_mprofile (int argc, char ** argv) [static]

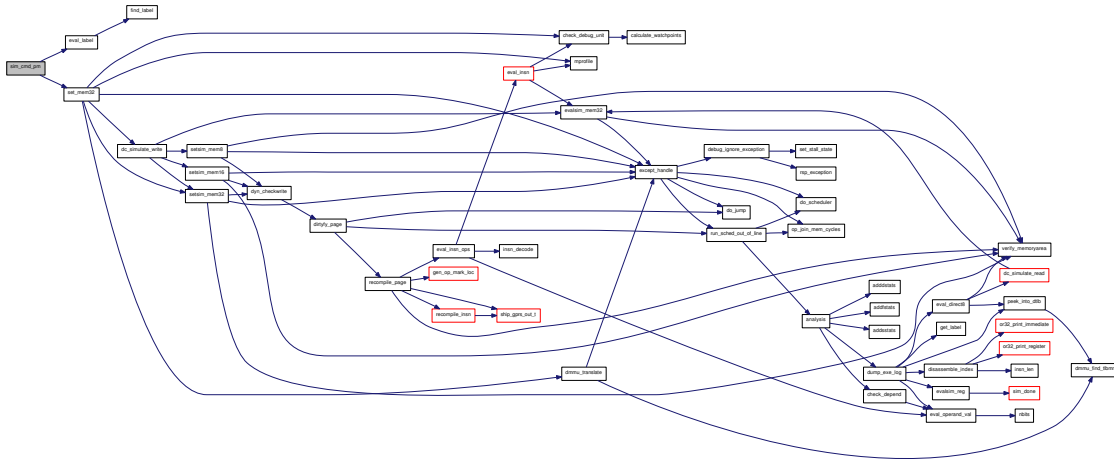
Here is the call graph for this function:



6.147.1.19 static int sim_cmd_pc (int argc, char ** argv) [static]

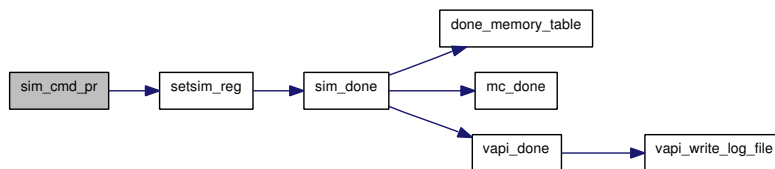
6.147.1.20 static int sim_cmd_pm (int argc, char ** argv) [static]

Here is the call graph for this function:



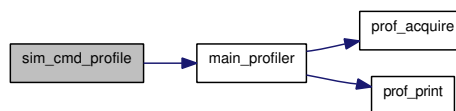
6.147.1.21 static int sim_cmd_pr (int argc, char ** argv) [static]

Here is the call graph for this function:



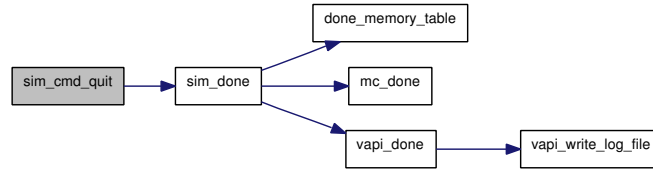
6.147.1.22 static int sim_cmd_profile (int argc, char ** argv) [static]

Here is the call graph for this function:



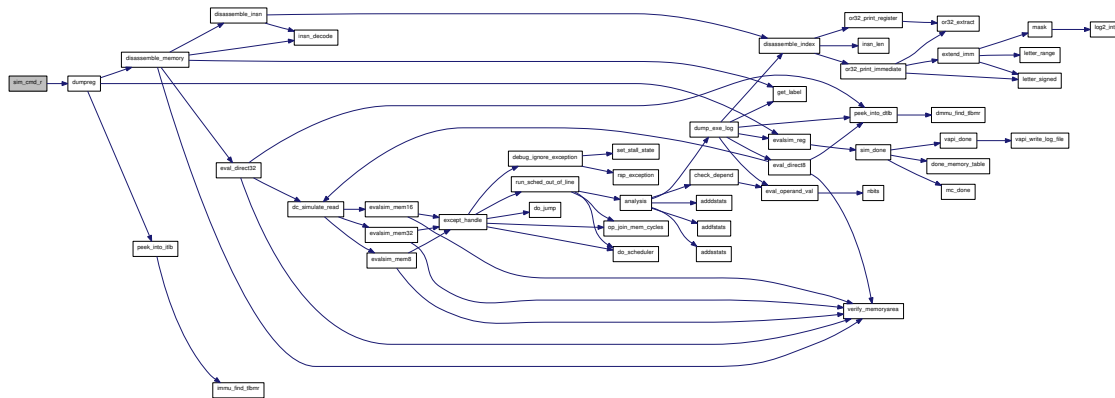
6.147.1.23 static int sim_cmd_quit (int argc, char ** argv) [static]

Here is the call graph for this function:



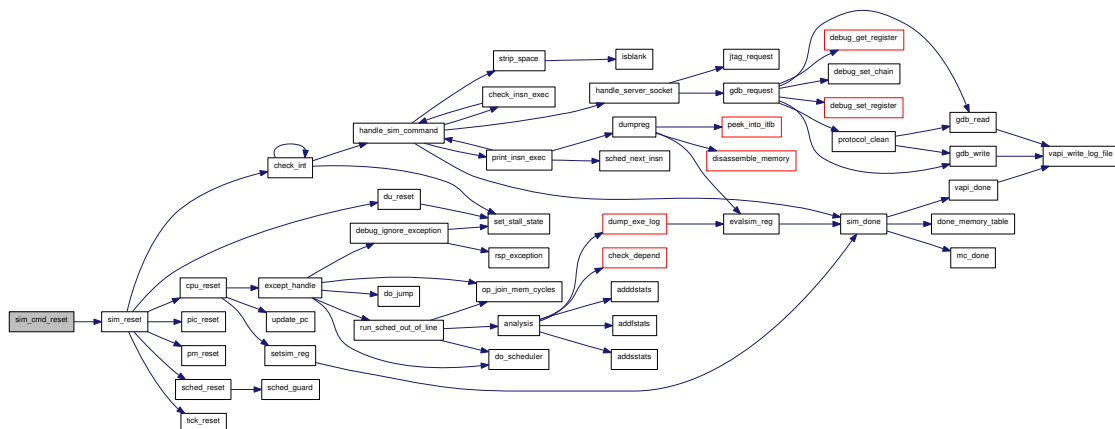
6.147.1.24 static int sim_cmd_r (int argc, char ** argv) [static]

Here is the call graph for this function:



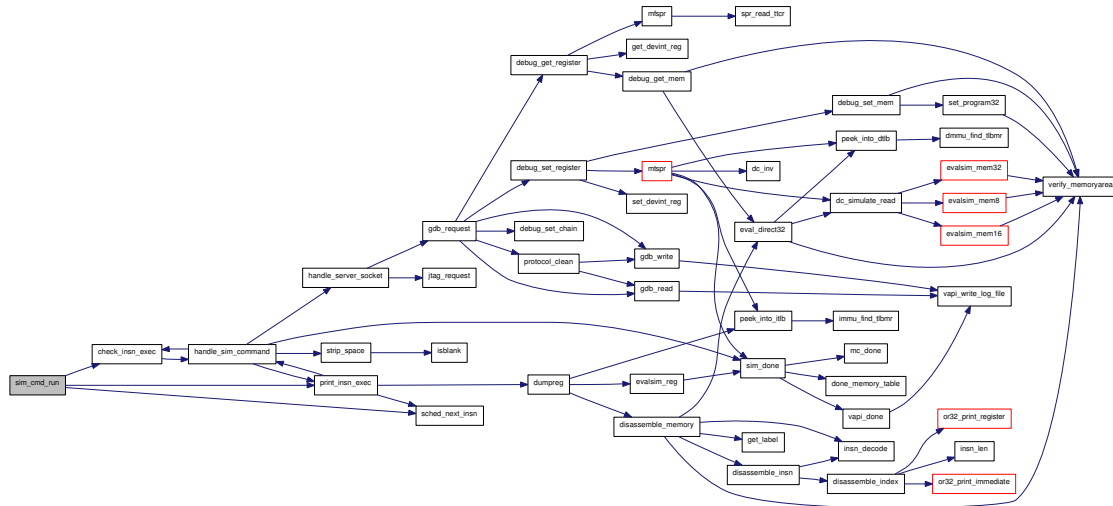
6.147.1.25 static int sim_cmd_reset (int argc, char ** argv) [static]

Here is the call graph for this function:



6.147.1.26 `static int sim_cmd_run (int argc, char ** argv) [static]`

Here is the call graph for this function:



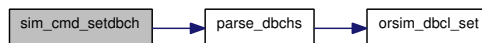
6.147.1.27 `static int sim_cmd_set (int argc, char ** argv) [static]`

Here is the call graph for this function:



6.147.1.28 `static int sim_cmd_setdbch (int argc, char ** argv) [static]`

Here is the call graph for this function:



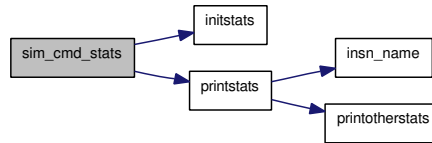
6.147.1.29 `static int sim_cmd_stall (int argc, char ** argv) [static]`

Here is the call graph for this function:

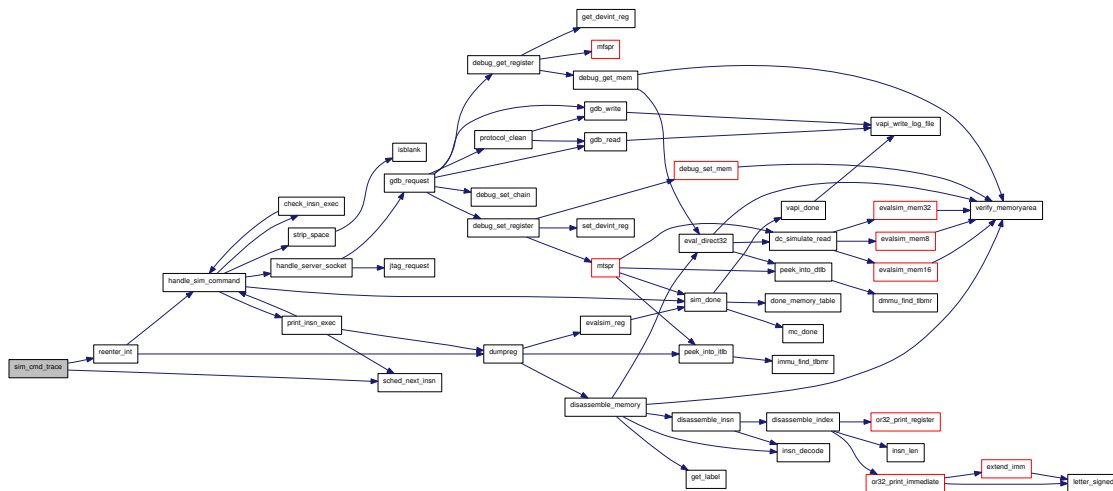


6.147.1.30 `static int sim_cmd_stats (int argc, char ** argv) [static]`

Here is the call graph for this function:

**6.147.1.31** `static int sim_cmd_trace (int argc, char ** argv) [static]`

Here is the call graph for this function:

**6.147.1.32** `static int sim_cmd_unstall (int argc, char ** argv) [static]`

Here is the call graph for this function:

**6.147.1.33** `static char* strip_space (char * str) [static]`

Here is the call graph for this function:



6.147.2 Variable Documentation

6.147.2.1 struct sim_command sim_commands[] [static]

Initial value:

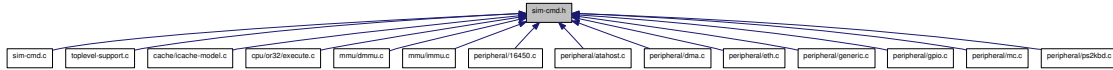
```
{
  {"q", sim_cmd_quit},
  {"help", sim_cmd_help},
  {"t", sim_cmd_trace},
  {"dm", sim_cmd_dm},
  {"dv", sim_cmd_dv},
  {"dh", sim_cmd_dh},
  {"pm", sim_cmd_pm},
  {"cm", sim_cmd_cm},
  {"pr", sim_cmd_pr},
  {"pc", sim_cmd_pc},
  {"breaks", sim_cmd_breaks},
  {"break", sim_cmd_break},
  {"r", sim_cmd_r},
  {"de", sim_cmd_de},
  {"reset", sim_cmd_reset},
  {"hist", sim_cmd_hist},
  {"stall", sim_cmd_stall},
  {"unstall", sim_cmd_unstall},
  {"stats", sim_cmd_stats},
  {"info", sim_cmd_info},
  {"run", sim_cmd_run},
  {"setdbch", sim_cmd_setdbch},
  {"debug", sim_cmd_debug},
  {"profile", sim_cmd_profile},
  {"mprofile", sim_cmd_mprofile},
  {"cuc", sim_cmd_cuc},
  {"set", sim_cmd_set},
  {NULL, NULL}
}
```

6.147.2.2 struct sim_stat* sim_stats = NULL [static]

6.147.2.3 long long to_insn_num [static]

6.148 sim-cmd.h File Reference

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



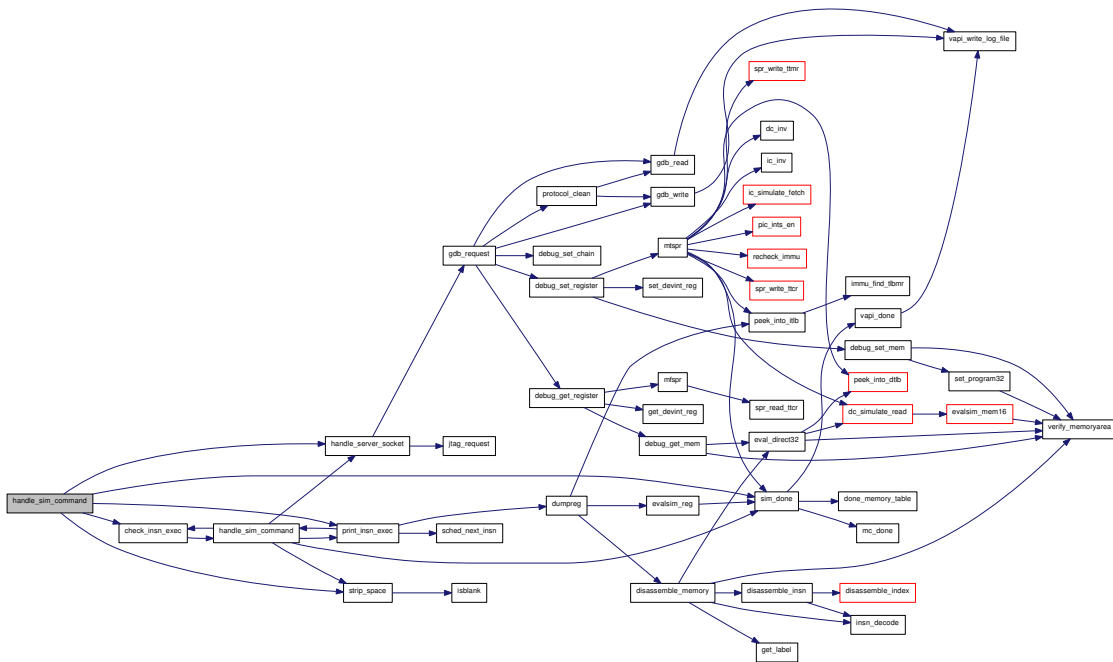
Functions

- void [handle_sim_command](#) (void)
- void [reg_sim_stat](#) (void(*stat_func)(void *dat), void *dat)

6.148.1 Function Documentation

6.148.1.1 void handle_sim_command (void)

Here is the call graph for this function:

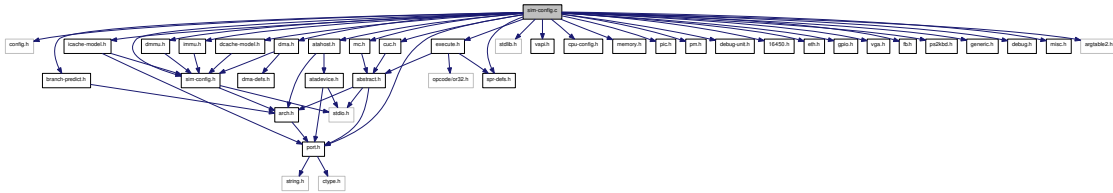


6.148.1.2 void reg_sim_stat (void(*) (void *dat) stat_func, void * dat)

6.149 sim-config.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "sim-config.h"
#include "vapi.h"
#include "cuc.h"
#include "cpu-config.h"
#include "memory.h"
#include "dmmu.h"
#include "immu.h"
#include "dcache-model.h"
#include "icache-model.h"
#include "pic.h"
#include "pm.h"
#include "branch-predict.h"
#include "debug-unit.h"
#include "mc.h"
#include "16450.h"
#include "dma.h"
#include "eth.h"
#include "gpio.h"
#include "vga.h"
#include "fb.h"
#include "ps2kbd.h"
#include "atahost.h"
#include "generic.h"
#include "execute.h"
#include "spr-defs.h"
#include "debug.h"
#include "misc.h"
#include "argtable2.h"
```


Include dependency graph for sim-config.c:



Data Structures

- struct `config_param`

Functions

- static void `read_script_file` (const char *filename)
- void `init_defconfig` ()
- int `parse_args` (int argc, char *argv[])
- void `print_config` ()
- void `base_include` (union `param_val` val, void *dat)
- void `sim_verbose` (union `param_val` val, void *dat)
- void `sim_debug` (union `param_val` val, void *dat)
- void `sim_profile` (union `param_val` val, void *dat)
- void `sim_prof_fn` (union `param_val` val, void *dat)
- void `sim_mprofile` (union `param_val` val, void *dat)
- void `sim_mprof_fn` (union `param_val` val, void *dat)
- void `sim_history` (union `param_val` val, void *dat)
- void `sim_exe_log` (union `param_val` val, void *dat)
- void `sim_exe_log_type` (union `param_val` val, void *dat)
- void `sim_exe_log_start` (union `param_val` val, void *dat)
- void `sim_exe_log_end` (union `param_val` val, void *dat)
- void `sim_exe_log_marker` (union `param_val` val, void *dat)
- void `sim_exe_log_fn` (union `param_val` val, void *dat)
- void `sim_clkcycle` (union `param_val` val, void *dat)
- static void `reg_sim_sec` ()
- void `reg_config_secs` (void)
- void `reg_config_param` (struct `config_section` *sec, const char *param, enum `param_t` type, void(*param_cb)(union `param_val`, void *))
- struct `config_section` * `reg_config_sec` (const char *section, void>(*sec_start)(void), void(*sec_end)(void *))
- static void `switch_param` (char *param, struct `config_param` *cur_param)
- static int `set_config` (int argc, char **argv)
- void `set_config_command` (int argc, char **argv)

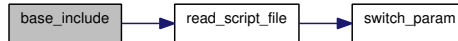
Variables

- struct `config` `config`
- struct `runtime` `runtime`
- struct `config_section` * `cur_section`
- struct `config_section` * `sections` = NULL

6.149.1 Function Documentation

6.149.1.1 void base_include (union param_val val, void * dat)

Here is the call graph for this function:



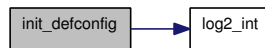
6.149.1.2 void init_defconfig (void)

Set default configuration parameters for fixed components

These values are held in the global `config` variable. Parameter orders match the order in the corresponding section registration function and documentation.

Also set some starting values for `runtime` elements.

Here is the call graph for this function:



6.149.1.3 int parse_args (int argc, char * argv[])

Parse the arguments for the standalone simulator

Updated by Jeremy Bennett to use argtable2.

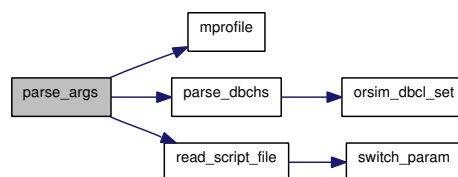
Parameters:

- ← *argc* Number of command args
- ← *argv* Vector of the command args

Returns:

- 0 on success, 1 on failure

Here is the call graph for this function:



6.149.1.4 void print_config (void)

Print the current configuration

Here is the call graph for this function:



6.149.1.5 static void read_script_file (const char *filename) [static]

Here is the call graph for this function:

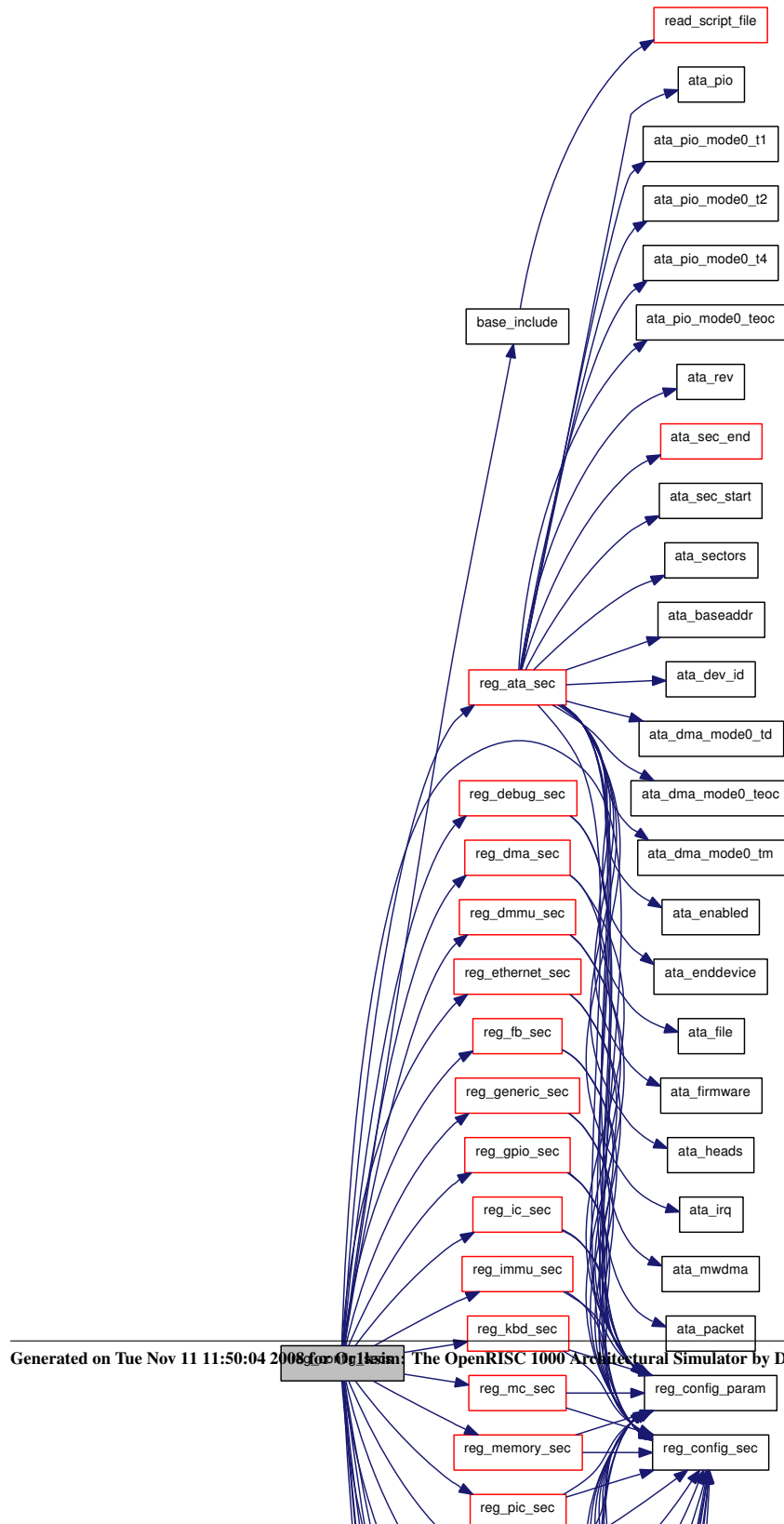


6.149.1.6 void reg_config_param (struct config_section *sec, const char *param, enum param_t type, void*(*)(union param_val, void *) param_cb)

6.149.1.7 struct config_section* reg_config_sec (const char *section, void *(*)(void) sec_start, void*(*)(void *) sec_end) [read]

6.149.1.8 void reg_config_secs (void)

Here is the call graph for this function:



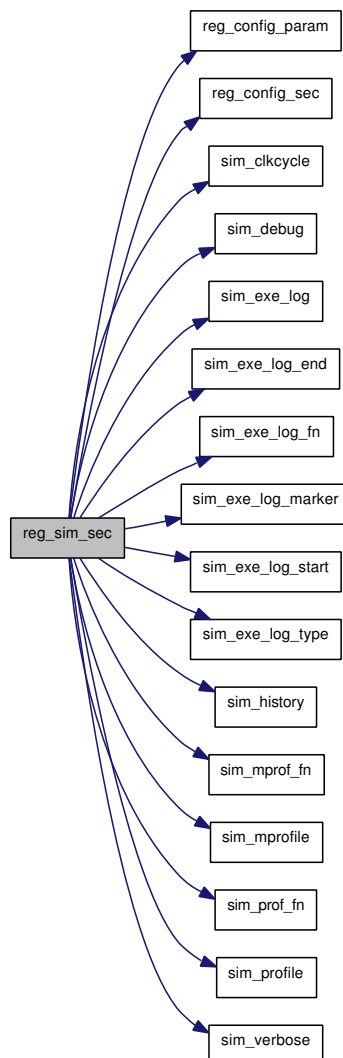
6.149.1.9 static void reg_sim_sec () [static]

Register the functions to handle a section sim

This section does not allocate dynamically a data structure holding its `config` information. It's all in the global `config.sim` data structure. Therefore it does not need a start and end function to initialize default values (although it might be clearer to do so). The default values are set in `init_defconfig()`.

New preferred parameter names are introduced (`_file` for filenames), but the legacy names (`_fn`) are also present for backwards compatibility

Here is the call graph for this function:

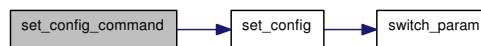


6.149.1.10 static int set_config (int argc, char ** argv) [static]

Here is the call graph for this function:

**6.149.1.11 void set_config_command (int argc, char ** argv)**

Here is the call graph for this function:

**6.149.1.12 void sim_clkcycle (union param_val val, void * dat)**

Set the clock cycle time.

Value must be an integer followed by one of ps, ns, us or ms.

If a valid time is not presented, the value is unchanged.

Parameters:

← *val* The value to use

← *dat* The `config` data structure (not used here)

6.149.1.13 void sim_debug (union param_val val, void * dat)

Set the simulator debug message level

Value must be in the range 0 (no messages) to 9. Values outside this range are converted to the nearer end of the range with a warning.

Parameters:

← *val* The value to use

← *dat* The `config` data structure (not used here)

6.149.1.14 void sim_exe_log (union param_val val, void * dat)

6.149.1.15 void sim_exe_log_end (union param_val val, void * dat)

6.149.1.16 void sim_exe_log_fn (union param_val val, void * dat)

6.149.1.17 void sim_exe_log_marker (union param_val val, void * dat)

6.149.1.18 void sim_exe_log_start (union param_val val, void * dat)

6.149.1.19 void sim_exe_log_type (union param_val val, void * dat)

Set the execution log type

Value must be one of default, hardware, simple or software. Invalid values are ignored with a warning.

Parameters:

← *val* The value to use

← *dat* The `config` data structure (not used here)

6.149.1.20 void sim_history (union param_val val, void * dat)

6.149.1.21 void sim_mprof_fn (union param_val val, void * dat)

6.149.1.22 void sim_mprofile (union param_val val, void * dat)

6.149.1.23 void sim_prof_fn (union param_val val, void * dat)

6.149.1.24 void sim_profile (union param_val val, void * dat)

6.149.1.25 void sim_verbose (union param_val val, void * dat)

6.149.1.26 static void switch_param (char * param, struct config_param * cur_param)
[static]

6.149.2 Variable Documentation

6.149.2.1 struct config config

6.149.2.2 struct config_section* cur_section

6.149.2.3 struct runtime runtime

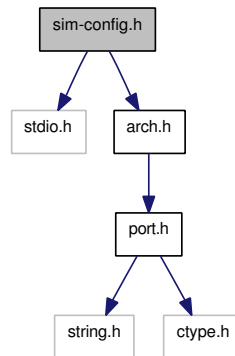
6.149.2.4 struct config_section* sections = NULL

6.150 sim-config.h File Reference

```
#include <stdio.h>
```

```
#include "arch.h"
```

Include dependency graph for sim-config.h:



Data Structures

- struct `config`
- struct `config::pic`
- struct `runtime`
- union `param_val`
- struct `config_section`

Defines

- #define `MAX_SBUF_LEN` 256
- #define `EXE_LOG_HARDWARE` 0
- #define `EXE_LOG_SIMPLE` 1
- #define `EXE_LOG_SOFTWARE` 2
- #define `STR_SIZE` 256
- #define `CHECK_INT_TIME` 100000
- #define `PRINTF(x...)` `fprintf (runtime.sim.fout, x)`

Enumerations

- enum `param_t` {
 `paramt_none` = 0, `paramt_str`, `paramt_word`, `paramt_int`,
 `paramt_longlong`, `paramt_addr` }

Functions

- void `set_config_command` (int argc, char **argv)
- void `init_defconfig` (void)

- int `parse_args` (int argc, char *argv[])
- void `print_config` (void)
- void `reg_config_param` (struct `config_section` *sec, const char *param, enum `param_t` type, void(*param_cb)(union `param_val`, void *))
- struct `config_section` * `reg_config_sec` (const char *section, void>(*sec_start)(void), void(*sec_end)(void *))
- void `reg_config_secs` ()

Variables

- struct `config config`
- struct `runtime runtime`
- struct `config_section` * `cur_section`
- int `do_stats`

6.150.1 Define Documentation

6.150.1.1 `#define CHECK_INT_TIME 100000`

6.150.1.2 `#define EXE_LOG_HARDWARE 0`

6.150.1.3 `#define EXE_LOG_SIMPLE 1`

6.150.1.4 `#define EXE_LOG_SOFTWARE 2`

6.150.1.5 `#define MAX_SBUF_LEN 256`

6.150.1.6 `#define PRINTF(x...) fprintf (runtime.sim.fout, x)`

6.150.1.7 `#define STR_SIZE 256`

6.150.2 Enumeration Type Documentation

6.150.2.1 `enum param_t`

Enum of all possible paramter types

Enumerator:

paramt_none

paramt_str

paramt_word

paramt_int

paramt_longlong

paramt_addr

6.150.3 Function Documentation

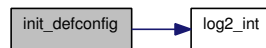
6.150.3.1 void init_defconfig (void)

Set default configuration parameters for fixed components

These values are held in the global `config` variable. Parameter orders match the order in the corresponding section registration function and documentation.

Also set some starting values for `runtime` elements.

Here is the call graph for this function:



6.150.3.2 int parse_args (int argc, char * argv[])

Parse the arguments for the standalone simulator

Updated by Jeremy Bennett to use argtable2.

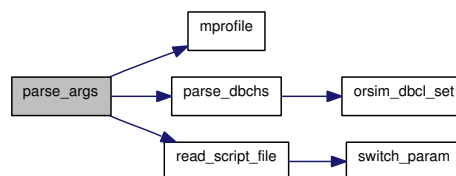
Parameters:

- ← *argc* Number of command args
- ← *argv* Vector of the command args

Returns:

- 0 on success, 1 on failure

Here is the call graph for this function:



6.150.3.3 void print_config (void)

Print the current configuration

Here is the call graph for this function:

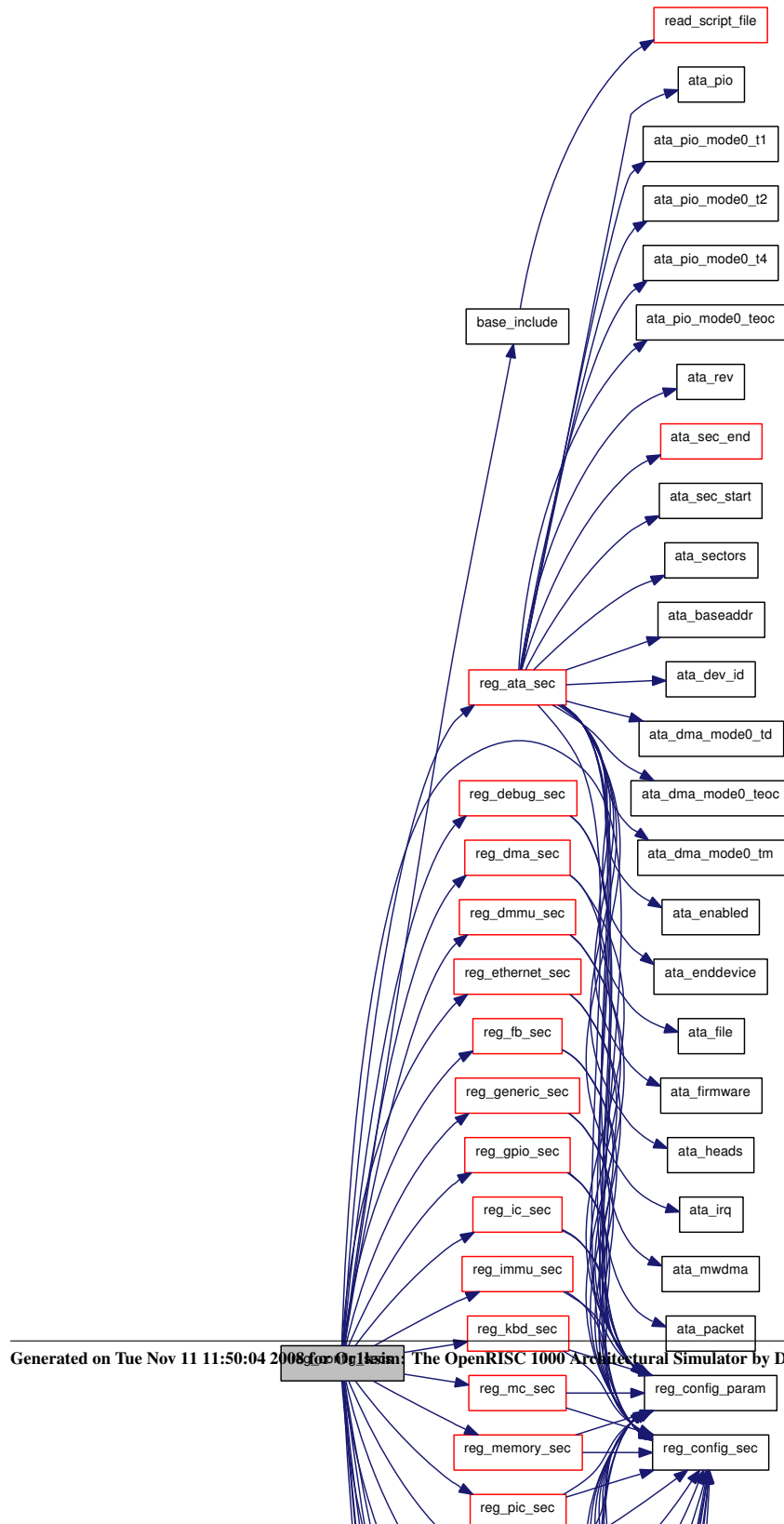


6.150.3.4 void reg_config_param (struct config_section * sec, const char * param, enum param_t type, void*(*)(union param_val, void *) param_cb)

6.150.3.5 struct config_section* reg_config_sec (const char * section, void *(*)(void) sec_start, void*(*)(void *) sec_end) [read]

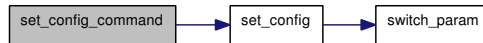
6.150.3.6 void reg_config_secs ()

Here is the call graph for this function:



6.150.3.7 void set_config_command (int argc, char ** argv)

Here is the call graph for this function:



6.150.4 Variable Documentation

6.150.4.1 struct config config

6.150.4.2 struct config_section* cur_section

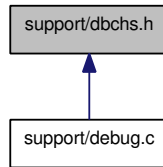
6.150.4.3 int do_stats

Whether we are doing statistical analysis. Globally available

6.150.4.4 struct runtime runtime

6.151 support/dbchs.h File Reference

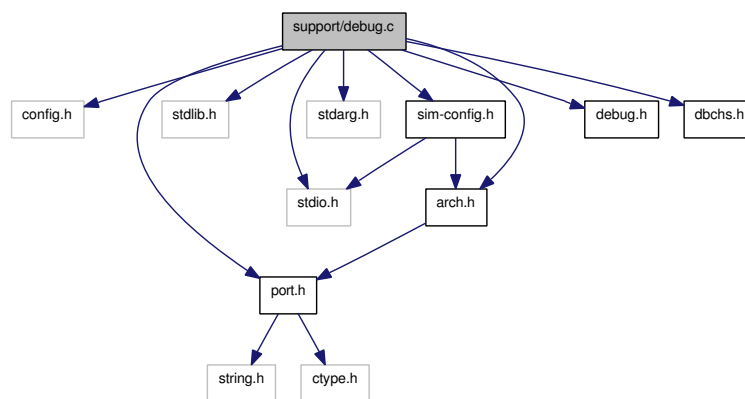
This graph shows which files directly or indirectly include this file:



6.152 support/debug.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include <stdarg.h>
#include "arch.h"
#include "sim-config.h"
#include "debug.h"
#include "dbchs.h"
```

Include dependency graph for debug.c:



Defines

- #define [__ORSIM_NO_DEC_DBCH](#)
- #define [DECLARE_DEBUG_CHANNEL](#)(dbch) char __orsim_dbch_##dbch[] = "\0"#dbch;
- #define [DECLARE_DEBUG_CHANNEL](#)(dbch) __orsim_dbch_##dbch,

Functions

- void [orsim_dbg_log](#) (enum [__ORSIM_DEBUG_CLASS](#) dbcl, const char *dbch, const char *function, const char *format,...)
- void [orsim_dbcl_set](#) (enum [__ORSIM_DEBUG_CLASS](#) dbcl, char *dbch, int on)
- void [orsim_dbcl_set_name](#) (enum [__ORSIM_DEBUG_CLASS](#) dbcl, const char *dbch, int on)
- void [parse_dbchs](#) (const char *str)
- void [debug](#) (int level, const char *format,...)

Variables

- static char * [__orsim_dbchs](#) []
- static const char * [debug_classes](#) [] = { "trace", "fixme", "warn", "err" }

6.152.1 Define Documentation

6.152.1.1 `#define __ORSIM_NO_DEC_DBCH`

6.152.1.2 `#define DECLARE_DEBUG_CHANNEL(dbch) __orsim_dbch_##dbch,`

6.152.1.3 `#define DECLARE_DEBUG_CHANNEL(dbch) char __orsim_dbch_##dbch[] =
"\0"#dbch;`

6.152.2 Function Documentation

6.152.2.1 `void debug (int level, const char *format, ...)`

Internal debug function

Print the message if the level is greater than or equal to that specified in the configuration.

Parameters:

- ← *level* The debug level of this message
- ← *format* Varargs format string
- ← ... The varargs required by the string

6.152.2.2 `void orsim_dbcl_set (enum __ORSIM_DEBUG_CLASS dbcl, char *dbch, int on)`

6.152.2.3 `void orsim_dbcl_set_name (enum __ORSIM_DEBUG_CLASS dbcl, const char *dbch,
int on)`

Here is the call graph for this function:



6.152.2.4 `void orsim_dbg_log (enum __ORSIM_DEBUG_CLASS dbcl, const char *dbch, const
char *function, const char *format, ...)`

6.152.2.5 `void parse_dbchs (const char *str)`

Here is the call graph for this function:



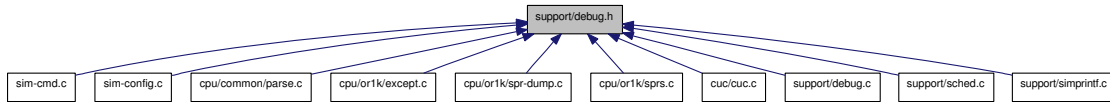
6.152.3 Variable Documentation

6.152.3.1 `char* __orsim_dbchs[] [static]`

6.152.3.2 `const char* debug_classes[] = { "trace", "fixme", "warn", "err" } [static]`

6.153 support/debug.h File Reference

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



Defines

- #define `__ORSIM_DBG_USE_FUNC` `__FUNCTION__`
- #define `__ORSIM_GET_DEBUGGING_TRACE`(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_TRACE))
- #define `__ORSIM_GET_DEBUGGING_WARN`(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_WARN))
- #define `__ORSIM_GET_DEBUGGING_FIXME`(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_FIXME))
- #define `__ORSIM_GET_DEBUGGING_ERR`(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_ERR))
- #define `__ORSIM_GET_DEBUGGING`(dbcl, dbch) `__ORSIM_GET_DEBUGGING##dbcl`(dbch)
- #define `__ORSIM_DPRINTF`(dbcl, dbch)
- #define `__ORSIM_DEBUG_LOG`(args...) `orsim_dbg_log(__dbcl, __dbch, __ORSIM_DBG_USE_FUNC, args); } }` while(0)
- #define `TRACE`(ch) `__ORSIM_DPRINTF(_TRACE, __orsim_dbch_##ch)`
- #define `FIXME`(ch) `__ORSIM_DPRINTF(_FIXME, __orsim_dbch_##ch)`
- #define `WARN`(ch) `__ORSIM_DPRINTF(_WARN, __orsim_dbch_##ch)`
- #define `ERR`(ch) `__ORSIM_DPRINTF(_ERR, __orsim_dbch_##ch)`
- #define `TRACE` `__ORSIM_DPRINTF(_TRACE, __orsim_dbch__default)`
- #define `FIXME` `__ORSIM_DPRINTF(_FIXME, __orsim_dbch__default)`
- #define `WARN` `__ORSIM_DPRINTF(_WARN, __orsim_dbch__default)`
- #define `ERR` `__ORSIM_DPRINTF(_ERR, __orsim_dbch__default)`
- #define `TRACE_ON`(ch) `__ORSIM_GET_DEBUGGING(_TRACE, __orsim_dbch_##ch)`
- #define `WARN_ON`(ch) `__ORSIM_GET_DEBUGGING(_WARN, __orsim_dbch_##ch)`
- #define `FIXME_ON`(ch) `__ORSIM_GET_DEBUGGING(_FIXME, __orsim_dbch_##ch)`
- #define `ERR_ON`(ch) `__ORSIM_GET_DEBUGGING(_ERR, __orsim_dbch_##ch)`
- #define `DEFAULT_DEBUG_CHANNEL`(dbch)
- #define `DECLARE_DEBUG_CHANNEL`(dbch) `extern char __orsim_dbch_##dbch[];`

Enumerations

- enum `__ORSIM_DEBUG_CLASS` { `__ORSIM_DBCL_TRACE`, `__ORSIM_DBCL_FIXME`, `__ORSIM_DBCL_WARN`, `__ORSIM_DBCL_ERR` }

Functions

- void `orsim_dbg_log` (enum `__ORSIM_DEBUG_CLASS` dbcl, const char *dbch, const char *function, const char *format,...) `__attribute__((format printf`
- void `orsim_dbcl_set_name` (enum `__ORSIM_DEBUG_CLASS` dbcl, const char *dbch, int on)
- void `parse_dbchs` (const char *str)
- void `debug` (int level, const char *format,...)

6.153.1 Define Documentation

6.153.1.1 `#define __ORSIM_DBG_USE_FUNC __FUNCTION__`

6.153.1.2 `#define __ORSIM_DEBUG_LOG(args...) orsim_dbg_log(__dbcl, __dbch, __ORSIM_DBG_USE_FUNC, args); } } while(0)`

6.153.1.3 `#define __ORSIM_DPRINTF(dbcl, dbch)`

Value:

```
do { if(__ORSIM_GET_DEBUGGING(dbcl, (dbch))) { \
    const char * const __dbch = dbch; \
    const enum __ORSIM_DEBUG_CLASS __dbcl = __ORSIM_DBCL##dbcl; \
    __ORSIM_DEBUG_LOG
```

6.153.1.4 `#define __ORSIM_GET_DEBUGGING(dbcl, dbch) __ORSIM_GET_DEBUGGING##dbcl(dbch)`

6.153.1.5 `#define __ORSIM_GET_DEBUGGING_ERR(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_ERR))`

6.153.1.6 `#define __ORSIM_GET_DEBUGGING_FIXME(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_FIXME))`

6.153.1.7 `#define __ORSIM_GET_DEBUGGING_TRACE(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_TRACE))`

6.153.1.8 `#define __ORSIM_GET_DEBUGGING_WARN(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_WARN))`

6.153.1.9 `#define DECLARE_DEBUG_CHANNEL(dbch) extern char __orsim_dbch_##dbch[];`

6.153.1.10 `#define DEFAULT_DEBUG_CHANNEL(dbch)`

Value:

```
extern char __orsim_dbch_##dbch[]; \
static char * const __orsim_dbch___default = __orsim_dbch_##dbch;
```

- 6.153.1.11 `#define ERR __ORSIM_DPRINTF(_ERR, __orsim_dbch__default)`
- 6.153.1.12 `#define ERR_(ch) __ORSIM_DPRINTF(_ERR, __orsim_dbch_##ch)`
- 6.153.1.13 `#define ERR_ON(ch) __ORSIM_GET_DEBUGGING(_ERR, __orsim_dbch_##ch)`
- 6.153.1.14 `#define FIXME __ORSIM_DPRINTF(_FIXME, __orsim_dbch__default)`
- 6.153.1.15 `#define FIXME_(ch) __ORSIM_DPRINTF(_FIXME, __orsim_dbch_##ch)`
- 6.153.1.16 `#define FIXME_ON(ch) __ORSIM_GET_DEBUGGING(_FIXME, __orsim_dbch_-##ch)`
- 6.153.1.17 `#define TRACE __ORSIM_DPRINTF(_TRACE, __orsim_dbch__default)`
- 6.153.1.18 `#define TRACE_(ch) __ORSIM_DPRINTF(_TRACE, __orsim_dbch_##ch)`
- 6.153.1.19 `#define TRACE_ON(ch) __ORSIM_GET_DEBUGGING(_TRACE, __orsim_dbch_-##ch)`
- 6.153.1.20 `#define WARN __ORSIM_DPRINTF(_WARN, __orsim_dbch__default)`
- 6.153.1.21 `#define WARN_(ch) __ORSIM_DPRINTF(_WARN, __orsim_dbch_##ch)`
- 6.153.1.22 `#define WARN_ON(ch) __ORSIM_GET_DEBUGGING(_WARN, __orsim_dbch_##ch)`

6.153.2 Enumeration Type Documentation

6.153.2.1 `enum __ORSIM_DEBUG_CLASS`

Enumerator:

- `__ORSIM_DBCL_TRACE`
- `__ORSIM_DBCL_FIXME`
- `__ORSIM_DBCL_WARN`
- `__ORSIM_DBCL_ERR`

6.153.3 Function Documentation

6.153.3.1 `void debug(int level, const char *format, ...)`

Internal debug function

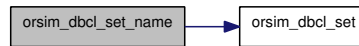
Print the message if the level is greater than or equal to that specified in the configuration.

Parameters:

- ← *level* The debug level of this message
- ← *format* Varargs format string
- ← ... The varargs required by the string

6.153.3.2 void void orsim_dbcl_set_name (enum __ORSIM_DEBUG_CLASS *dbcl*, const char * *dbch*, int *on*)

Here is the call graph for this function:



6.153.3.3 void orsim_dbg_log (enum __ORSIM_DEBUG_CLASS *dbcl*, const char * *dbch*, const char * *function*, const char * *format*, ...)

6.153.3.4 void parse_dbchs (const char * *str*)

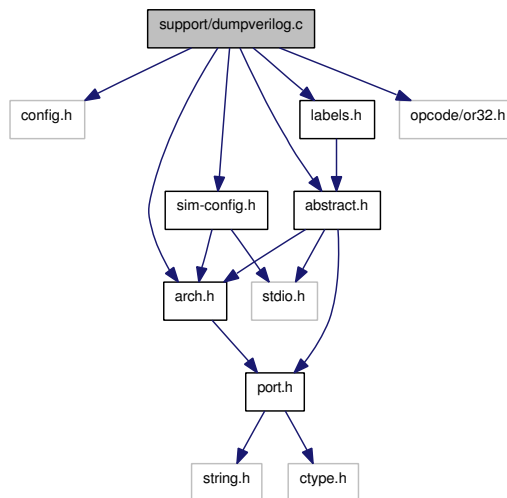
Here is the call graph for this function:



6.154 support/dumpverilog.c File Reference

```
#include "config.h"
#include "sim-config.h"
#include "arch.h"
#include "abstract.h"
#include "labels.h"
#include "opcode/or32.h"
```

Include dependency graph for dumpverilog.c:



Defines

- `#define DW 32`
- `#define DWQ (DW/8)`
- `#define DISWIDTH 25`
- `#define OR1K_MEM_VERILOG_HEADER(MODNAME, FROMADDR, TOADDR, DISWIDTH)`
- `#define OR1K_MEM_VERILOG_FOOTER "\n\nalways @(posedge clk) begin\n\n if (ce && ~we) begin\n\n dataout <= #1 mem[addr];\n\n disout <= #1 dis[addr];\n\n $display(\"or1k_mem: reading mem[%0d]:%0h dis: %0s\", addr, dataout, dis[addr]);\n\n end else\n\n if (ce && we) begin\n\n mem[addr] <= #1 data;\n\n dis[addr] <= #1 \"(data)\";\n\n $display(\"or1k_mem: writing mem[%0d]:%0h dis: %0s\", addr, mem[addr], dis[addr]);\n\n end\n\nend\n\nendmodule\n\n"`
- `#define LABELEND_CHAR ":"`

Functions

- void `dumpverilog` (char *verilog_modname, `oraddr_t` from, `oraddr_t` to)
- void `dumphex` (`oraddr_t` from, `oraddr_t` to)

6.154.1 Define Documentation

6.154.1.1 `#define DISWIDTH 25`

6.154.1.2 `#define DW 32`

6.154.1.3 `#define DWQ (DW/8)`

6.154.1.4 `#define LABELEND_CHAR ""`

6.154.1.5 `#define OR1K_MEM_VERILOG_FOOTER "\n\nend\n\nalways @(posedge clk) begin\n\n if (ce && ~we) begin\n\n dataout <= #1 mem[addr];\n\n disout <= #1 dis[addr];\n\n $display(\"or1k_mem: reading mem[%0d]:%0h dis: %0s\", addr, dataout, dis[addr]);\n\n end else\n\n if (ce && we) begin\n\n mem[addr] <= #1 data;\n\n dis[addr] <= #1 \"(data)\";\n\n $display(\"or1k_mem: writing mem[%0d]:%0h dis: %0s\", addr, mem[addr], dis[addr]);\n\n end\n\nend\n\nendmodule\n\n"`

6.154.1.6 `#define OR1K_MEM_VERILOG_HEADER(MODNAME, FROMADDR, TOADDR, DISWIDTH)`

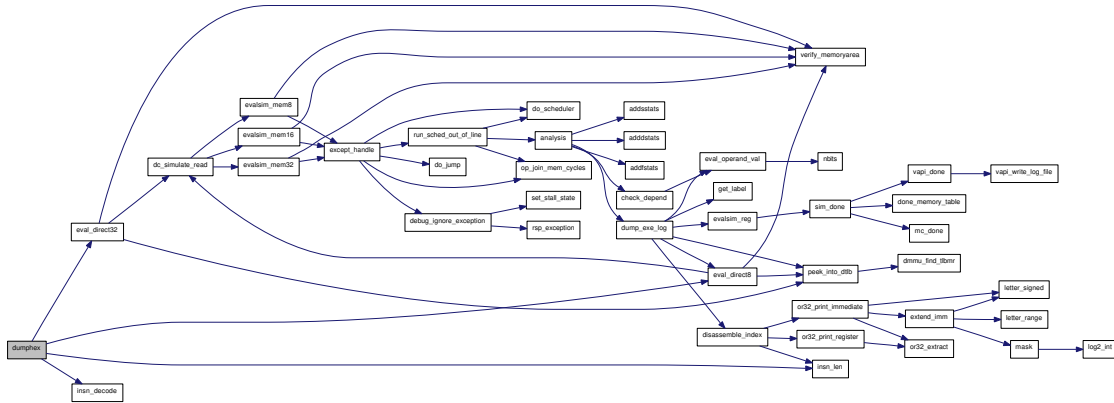
Value:

```
"\n\n"
#include "general.h"\n\n"
`timescale 1ns/100ps\n\n"
// Simple dw-wide Sync SRAM with initial content generated by orlksim.\n\n
// All control, data in and addr signals are sampled at rising clock edge \n\n
// Data out is not registered. Address bits specify dw-word (narrowest \n\n
// addressed data is not byte but dw-word !). \n\n
// There are still some bugs in generated output (dump word aligned regions)\n\n"
module %s(clk, data, addr, ce, we, disout);\n\n"
parameter dw = 32;\n\n"
parameter amin = %" PRIuREG "; \n\n"
parameter amax = %" PRIuREG "; \n\n"
input clk;\n\n"
inout [dw-1:0] data;\n\n"
input [31:0] addr;\n\n"
input ce;\n\n"
input we;\n\n"
output [%d:0] disout;\n\n"
reg [%d:0] disout;\n\n"
reg [dw-1:0] mem [amax:amin];\n\n"
reg [%d:0] dis [amax:amin];\n\n"
reg [dw-1:0] dataout;\n\n"
tri [dw-1:0] data = (ce && ~we) ? dataout : 'bz;\n\n"
initial begin\n\n", MODNAME, FROMADDR, TOADDR, DISWIDTH-1, DISWIDTH-1,
```

6.154.2 Function Documentation

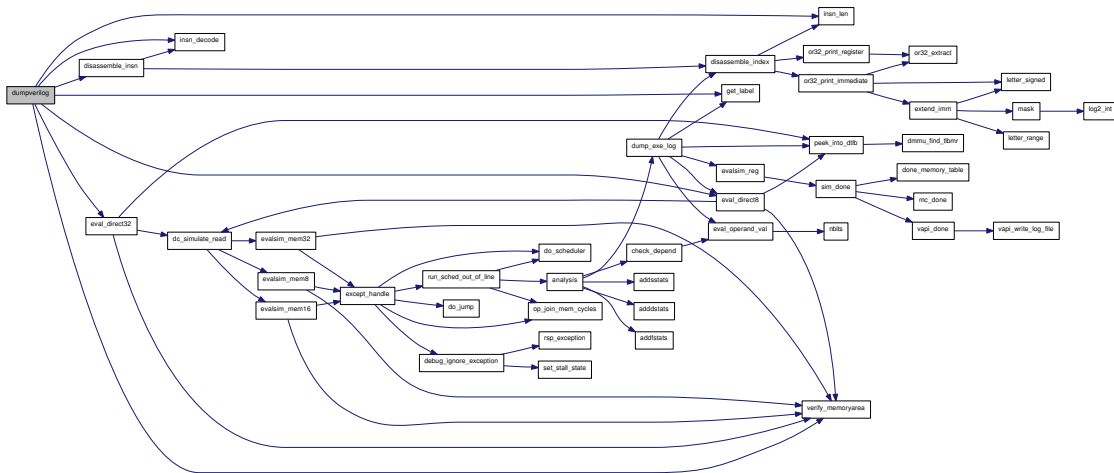
6.154.2.1 void dumphex (oraddr_t from, oraddr_t to)

Here is the call graph for this function:



6.154.2.2 void dumpverilog (char * verilog_modname, oraddr_t from, oraddr_t to)

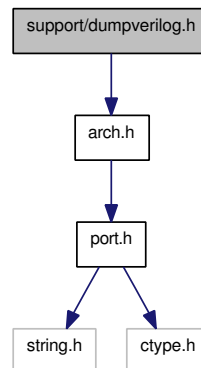
Here is the call graph for this function:



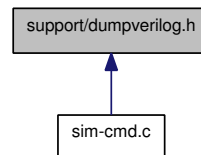
6.155 support/dumpverilog.h File Reference

```
#include "arch.h"
```

Include dependency graph for dumpverilog.h:



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



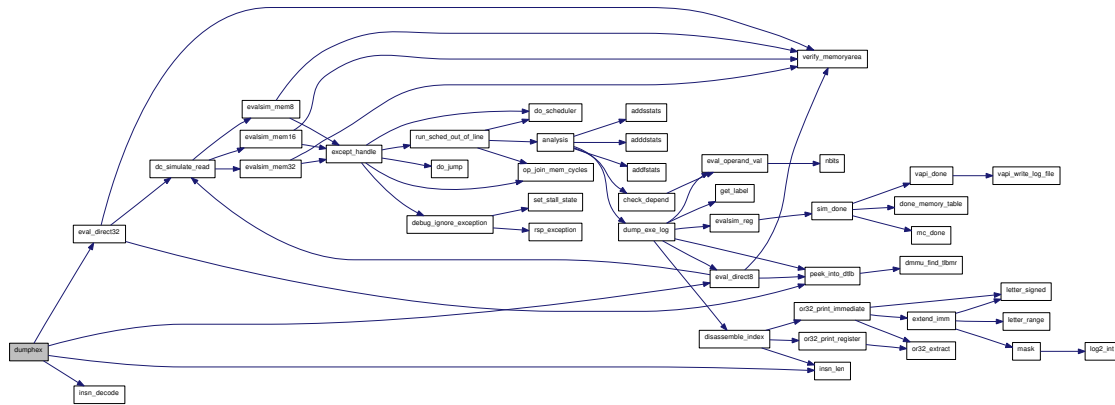
Functions

- void `dumpverilog` (char *verilog_modname, `oraddr_t` from, `oraddr_t` to)
- void `dumphex` (`oraddr_t` from, `oraddr_t` to)

6.155.1 Function Documentation

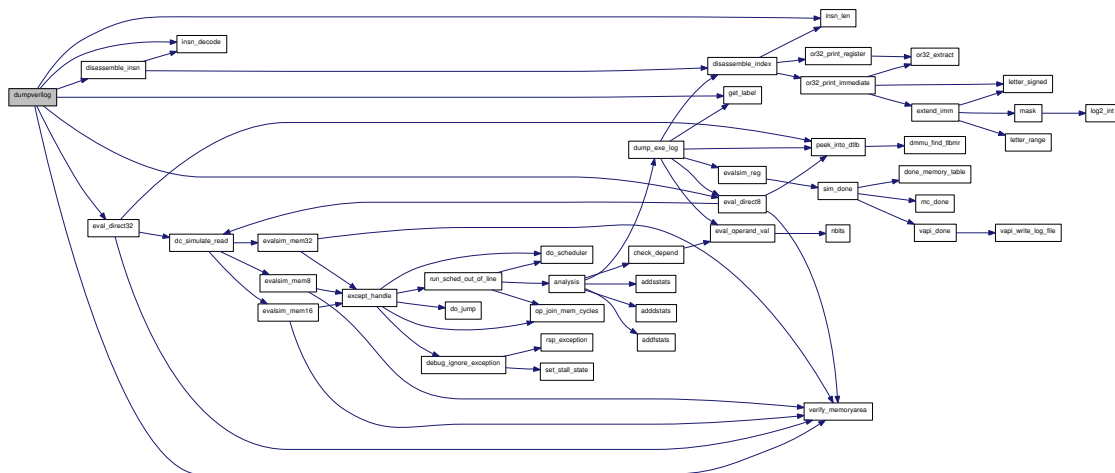
6.155.1.1 void dumphex (oraddr_t from, oraddr_t to)

Here is the call graph for this function:



6.155.1.2 void dumpverilog (char * verilog_modname, oraddr_t from, oraddr_t to)

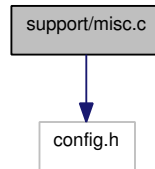
Here is the call graph for this function:



6.156 support/misc.c File Reference

```
#include "config.h"
```

Include dependency graph for misc.c:



Functions

- `int log2_int` (unsigned long *x*)
- `int is_power2` (int *x*)

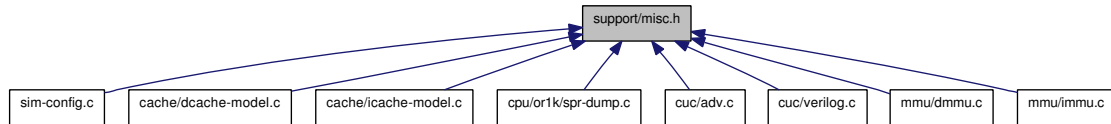
6.156.1 Function Documentation

6.156.1.1 `int is_power2` (int *x*)

6.156.1.2 `int log2_int` (unsigned long *x*)

6.157 support/misc.h File Reference

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



Functions

- [int log2_int](#) (unsigned long x)
- [int is_power2](#) (int x)

6.157.1 Function Documentation

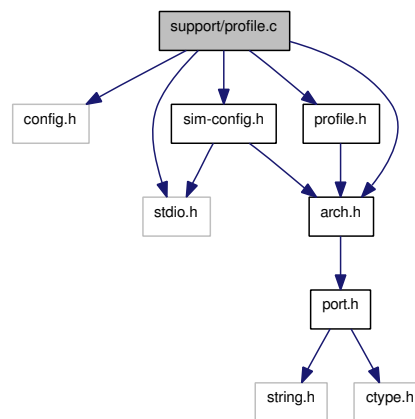
6.157.1.1 int is_power2 (int x)

6.157.1.2 int log2_int (unsigned long x)

6.158 support/profile.c File Reference

```
#include "config.h"  
#include <stdio.h>  
#include "profile.h"  
#include "sim-config.h"  
#include "arch.h"
```

Include dependency graph for profile.c:



Functions

- void `mprofile` (`oraddr_t memaddr`, unsigned char type)

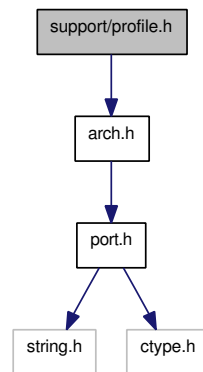
6.158.1 Function Documentation

6.158.1.1 void `mprofile` (`oraddr_t memaddr`, unsigned char type)

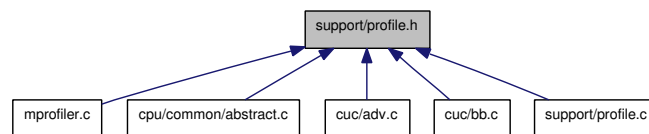
6.159 support/profile.h File Reference

```
#include "arch.h"
```

Include dependency graph for profile.h:



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



Data Structures

- struct [mprofentry_struct](#)

Defines

- #define [MPROF_READ](#) 1
- #define [MPROF_WRITE](#) 2
- #define [MPROF_FETCH](#) 4
- #define [MPROF_8](#) 8
- #define [MPROF_16](#) 16
- #define [MPROF_32](#) 32

Functions

- void [mprofile](#) ([oraddr_t](#) memaddr, unsigned char type)

6.159.1 Define Documentation

6.159.1.1 `#define MPROF_16 16`

6.159.1.2 `#define MPROF_32 32`

6.159.1.3 `#define MPROF_8 8`

6.159.1.4 `#define MPROF_FETCH 4`

6.159.1.5 `#define MPROF_READ 1`

6.159.1.6 `#define MPROF_WRITE 2`

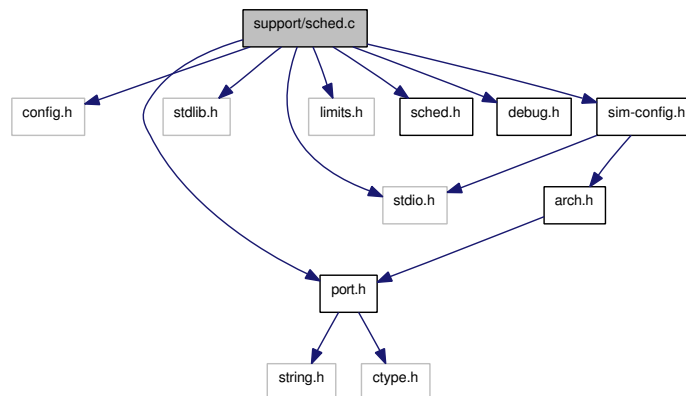
6.159.2 Function Documentation

6.159.2.1 `void mprofile (oraddr_t memaddr, unsigned char type)`

6.160 support/sched.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include "sched.h"
#include "debug.h"
#include "sim-config.h"
```

Include dependency graph for sched.c:



Defines

- #define [SCHEM_HEAP_SIZE](#) 128
- #define [SCHEM_TIME_MAX](#) INT32_MAX

Functions

- [DECLARE_DEBUG_CHANNEL](#) (sched_jobs)
- void [sched_guard](#) (void *dat)
- void [sched_reset](#) (void)
- void [sched_init](#) (void)
- void [do_scheduler](#) (void)
- void [sched_add](#) (void(*job_func)(void *), void *job_param, int32_t job_time, const char *func)
- void [sched_find_remove](#) (void(*job_func)(void *), void *dat)
- void [sched_next_insn](#) (void(*func)(void *), void *dat)

Variables

- struct [scheduler_struct](#) scheduler

6.160.1 Define Documentation

6.160.1.1 `#define SCHED_HEAP_SIZE 128`

6.160.1.2 `#define SCHED_TIME_MAX INT32_MAX`

6.160.2 Function Documentation

6.160.2.1 `DECLARE_DEBUG_CHANNEL (sched_jobs)`

6.160.2.2 `void do_scheduler (void)`

6.160.2.3 `void sched_add (void(*) (void *) job_func, void * job_param, int32_t job_time, const char * func)`

6.160.2.4 `void sched_find_remove (void(*) (void *) job_func, void * dat)`

6.160.2.5 `void sched_guard (void * dat)`

6.160.2.6 `void sched_init (void)`

Here is the call graph for this function:



6.160.2.7 `void sched_next_insn (void(*) (void *) func, void * dat)`

6.160.2.8 `void sched_reset (void)`

Here is the call graph for this function:

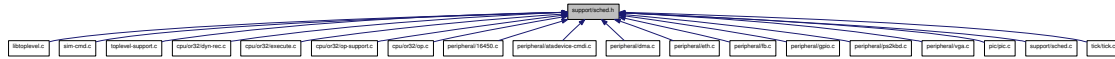


6.160.3 Variable Documentation

6.160.3.1 `struct scheduler_struct scheduler`

6.161 support/sched.h File Reference

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



Data Structures

- struct [sched_entry](#)
- struct [scheduler_struct](#)

Defines

- #define [SCHED_ADD](#)(job_func, job_param, job_time) `sched_add(job_func, job_param, job_time, #job_func)`
- #define [SCHED_FIND_REMOVE](#)(f, p) `sched_find_remove(f, p)`

Functions

- void [sched_init](#) ()
- void [sched_reset](#) ()
- void [sched_next_insn](#) (void(*func)(void *), void *dat)
- void [sched_find_remove](#) (void(*job_func)(void *), void *dat)
- void [sched_add](#) (void(*job_func)(void *), void *job_param, int32_t job_time, const char *func)
- void [do_scheduler](#) ()

Variables

- struct [scheduler_struct](#) `scheduler`

6.161.1 Define Documentation

6.161.1.1 #define [SCHED_ADD](#)(job_func, job_param, job_time) `sched_add(job_func, job_param, job_time, #job_func)`

Macro to add a job to the scheduler

6.161.1.2 #define [SCHED_FIND_REMOVE](#)(f, p) `sched_find_remove(f, p)`

Macro to remove a job from the scheduler

6.161.2 Function Documentation

6.161.2.1 void do_scheduler ()

6.161.2.2 void sched_add (void(*) (void *) *job_func*, void * *job_param*, int32_t *job_time*, const char * *func*)

6.161.2.3 void sched_find_remove (void(*) (void *) *job_func*, void * *dat*)

6.161.2.4 void sched_init ()

Here is the call graph for this function:



6.161.2.5 void sched_next_insn (void(*) (void *) *func*, void * *dat*)

6.161.2.6 void sched_reset ()

Here is the call graph for this function:



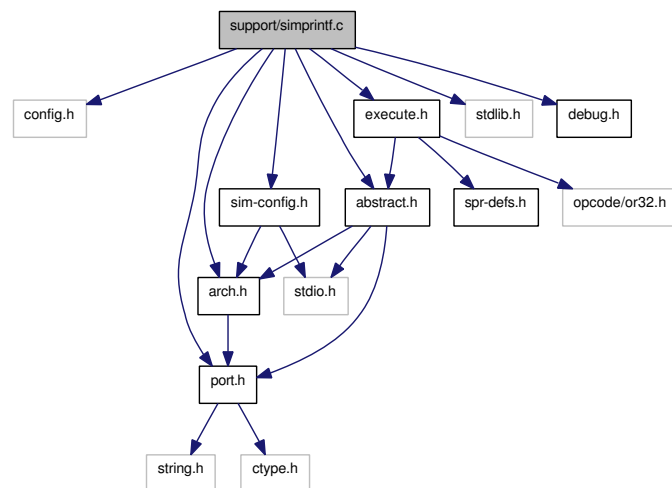
6.161.3 Variable Documentation

6.161.3.1 struct scheduler_struct scheduler

6.162 support/simprintf.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "sim-config.h"
#include "arch.h"
#include "debug.h"
#include "abstract.h"
#include "execute.h"
```

Include dependency graph for simprintf.c:



Defines

- #define [STACK_ARGS](#) 0
- #define [FMTLEN](#) 2000

Functions

- static char * [simgetstr](#) ([oraddr_t](#) stackaddr, unsigned long regparam)
- void [simprintf](#) ([oraddr_t](#) stackaddr, unsigned long regparam)

Variables

- char [fmtstr](#) [[FMTLEN](#)]

6.162.1 Define Documentation

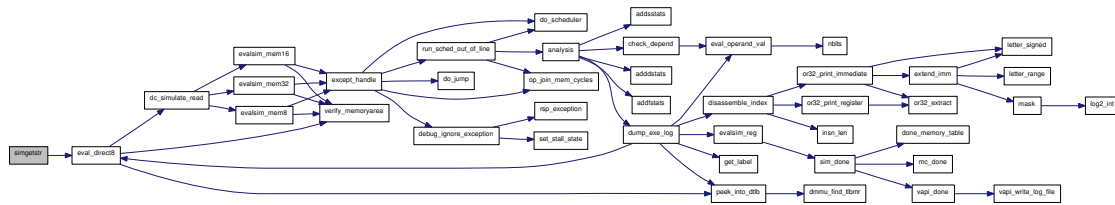
6.162.1.1 `#define FMTLEN 2000`

6.162.1.2 `#define STACK_ARGS 0`

6.162.2 Function Documentation

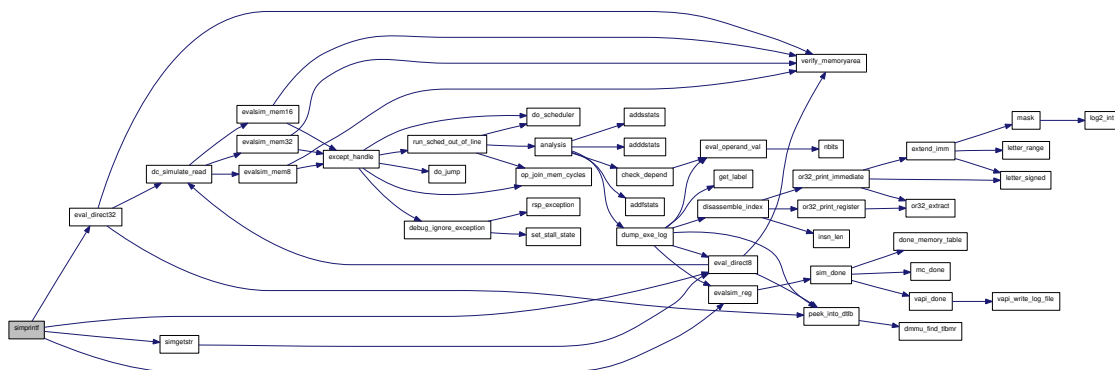
6.162.2.1 `static char* simgetstr (oraddr_t stackaddr, unsigned long regparam) [static]`

Here is the call graph for this function:



6.162.2.2 `void simprintf (oraddr_t stackaddr, unsigned long regparam)`

Here is the call graph for this function:



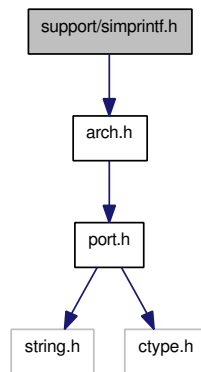
6.162.3 Variable Documentation

6.162.3.1 `char fmtstr[FMTLEN]`

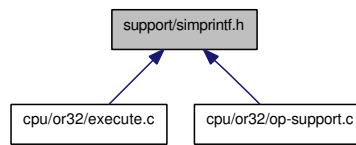
6.163 support/simprintf.h File Reference

```
#include "arch.h"
```

Include dependency graph for simprintf.h:



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



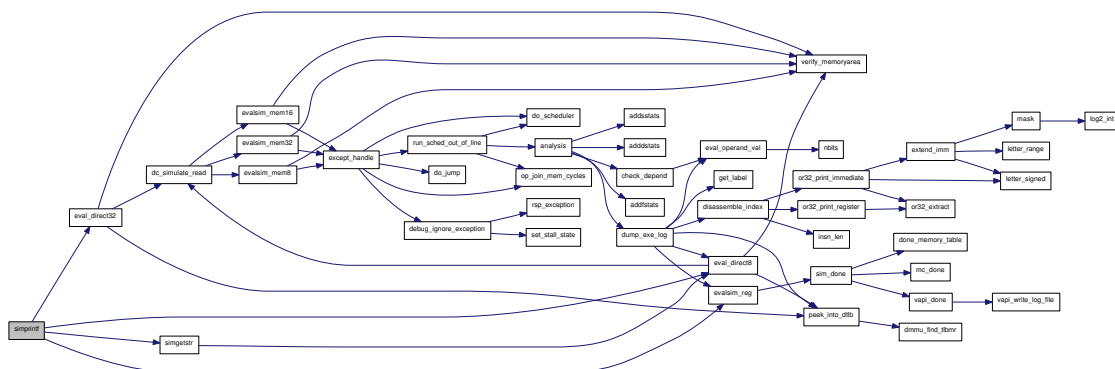
Functions

- void `simprintf` (`oraddr_t` stackaddr, unsigned long regparam)

6.163.1 Function Documentation

6.163.1.1 void `simprintf` (`oraddr_t` stackaddr, unsigned long regparam)

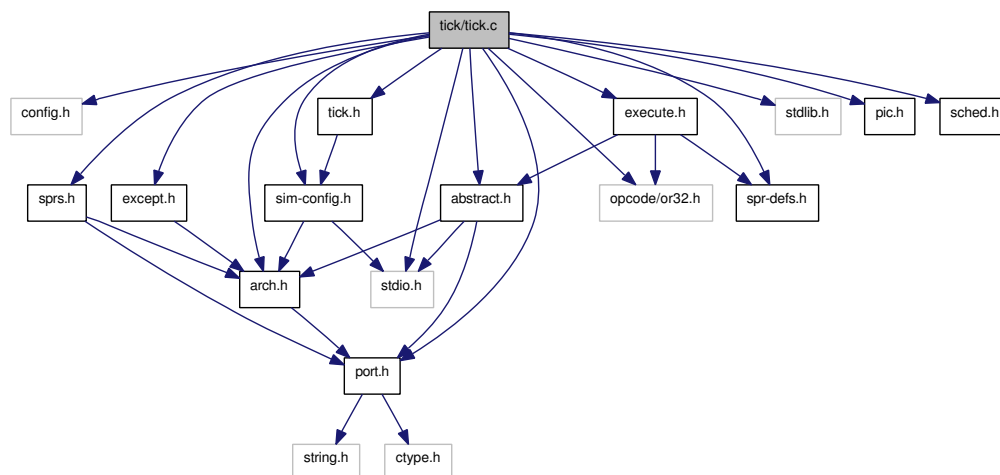
Here is the call graph for this function:



6.164 tick/tick.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <stdio.h>
#include "arch.h"
#include "abstract.h"
#include "except.h"
#include "tick.h"
#include "opcode/or32.h"
#include "spr-defs.h"
#include "execute.h"
#include "pic.h"
#include "sprs.h"
#include "sim-config.h"
#include "sched.h"
```

Include dependency graph for tick.c:



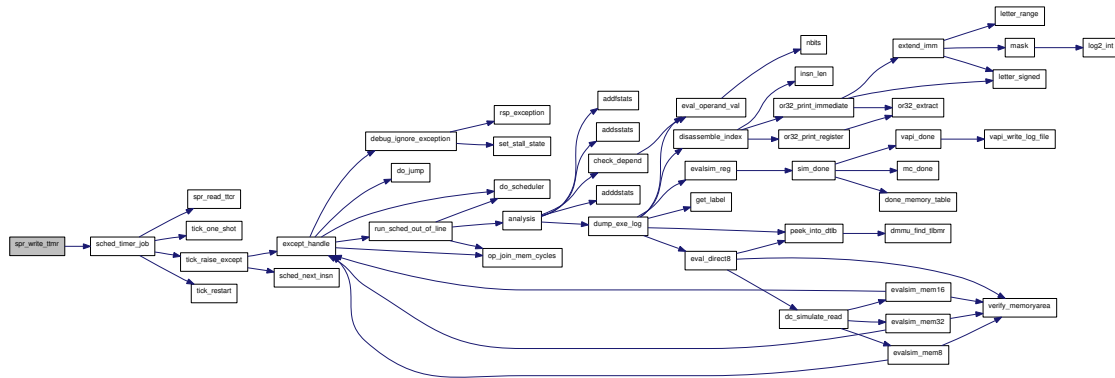
Functions

- void [tick_reset](#) (void)
- static void [tick_raise_except](#) (void *dat)
- static void [tick_restart](#) (void *dat)
- static void [tick_one_shot](#) (void *dat)
- static void [sched_timer_job](#) (uorreg_t prev_ttmr)
- void [spr_write_ttc](#) (uorreg_t value)
- void [spr_write_ttmr](#) (uorreg_t prev_val)

6.164.1.4 void spr_write_ttmr (uorreg_t prev_val)

Value is the *previous* value of SPR_TTMR. The new one can be found in `cpu_state.sprs[SPR_TTMR]`

Here is the call graph for this function:



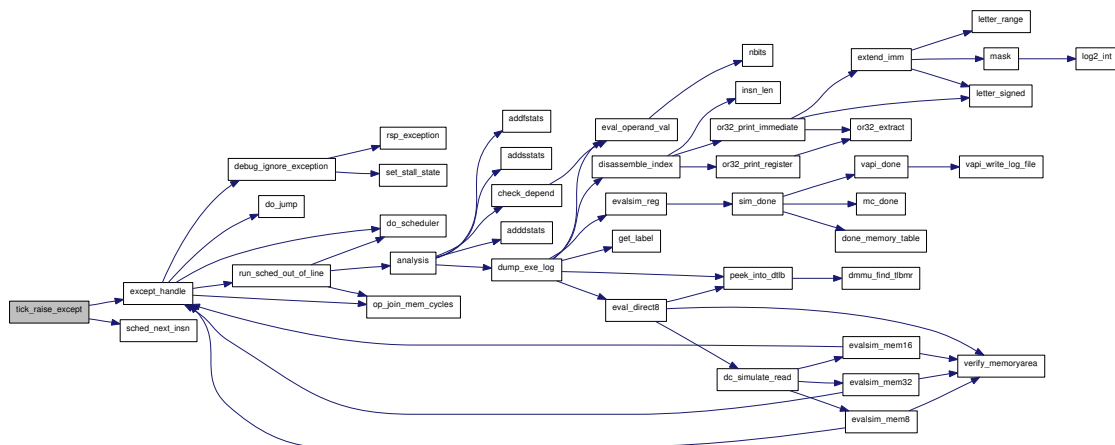
6.164.1.5 static void tick_one_shot (void * dat) [static]

Stops the timer

6.164.1.6 static void tick_raise_except (void * dat) [static]

Raises a timer exception

Here is the call graph for this function:



6.164.1.7 void tick_reset (void)

Reset. It initializes TTCR register.

6.164.1.8 `static void tick_restart (void * dat)` `[static]`

Restarts the tick timer

6.164.2 Variable Documentation

6.164.2.1 `long long cycles_start = 0` `[static]`

When did the timer start to count

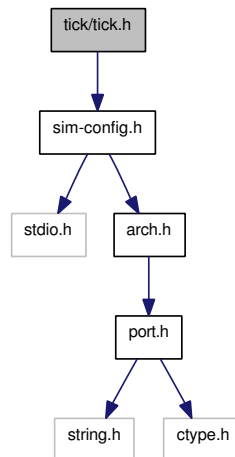
6.164.2.2 `int tick_count`

Indicates if the timer is actually counting. Needed to simulate one-shot mode correctly

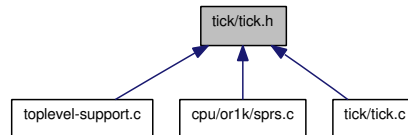
6.165 tick/tick.h File Reference

```
#include "sim-config.h"
```

Include dependency graph for tick.h:



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



Functions

- void [tick_reset \(\)](#)
- void [spr_write_ttc](#) ([uorreg_t value](#))
- void [spr_write_ttmr](#) ([uorreg_t value](#))
- [uorreg_t spr_read_ttc](#) ()

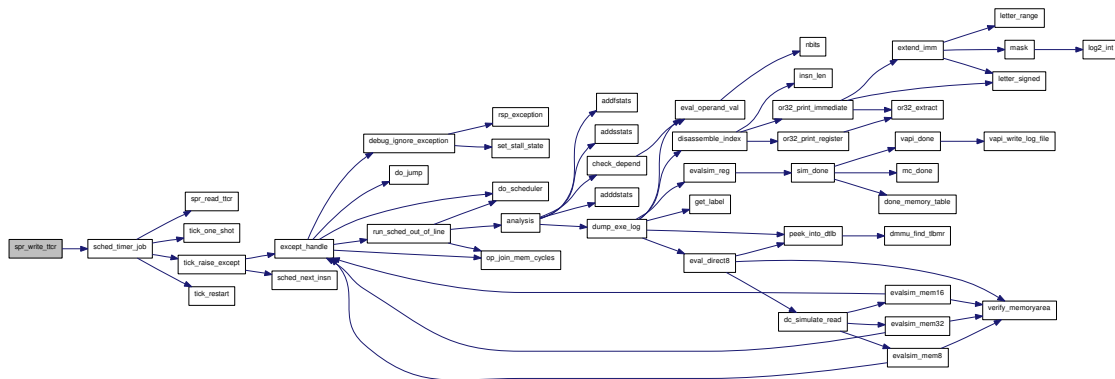
6.165.1 Function Documentation

6.165.1.1 uorreg_t spr_read_ttc ()

6.165.1.2 void spr_write_ttc (uorreg_t value)

Handles a write to the ttc spr

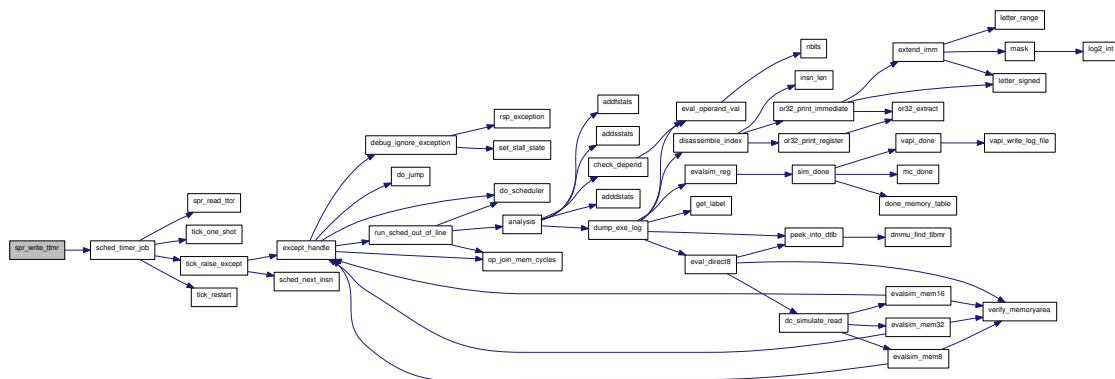
Here is the call graph for this function:



6.165.1.3 void spr_write_tmr (uorreg_t prev_val)

Value is the *previous* value of SPR_TTMR. The new one can be found in `cpu_state.sprs[SPR_TTMR]`

Here is the call graph for this function:



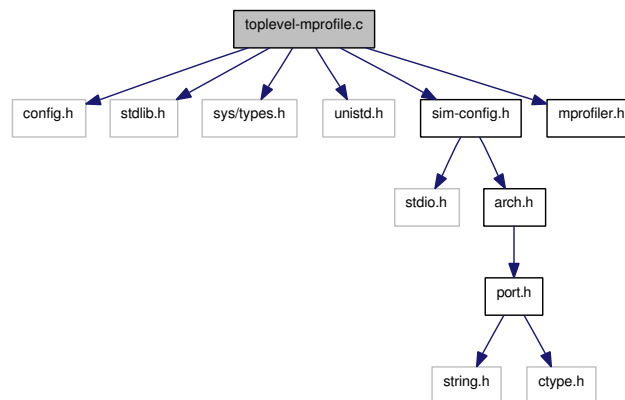
6.165.1.4 void tick_reset (void)

Reset. It initializes TTCR register.

6.166 toplevel-mprofile.c File Reference

```
#include "config.h"  
#include <stdlib.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include "sim-config.h"  
#include "mprofiler.h"
```

Include dependency graph for toplevel-mprofile.c:



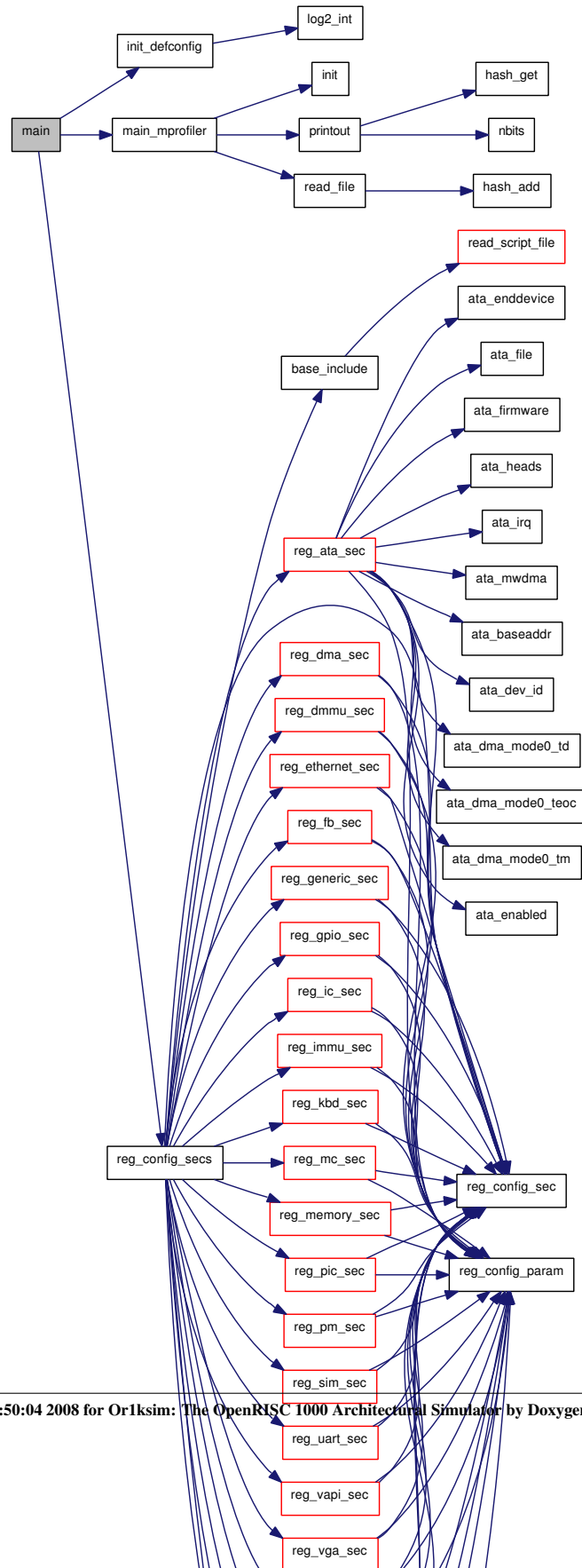
Functions

- `int main (int argc, char *argv[])`

6.166.1 Function Documentation

6.166.1.1 `int main (int argc, char * argv[])`

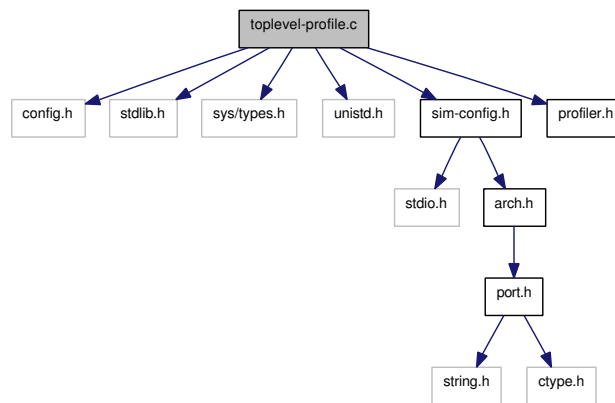
Here is the call graph for this function:



6.167 toplevel-profile.c File Reference

```
#include "config.h"  
#include <stdlib.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include "sim-config.h"  
#include "profiler.h"
```

Include dependency graph for toplevel-profile.c:



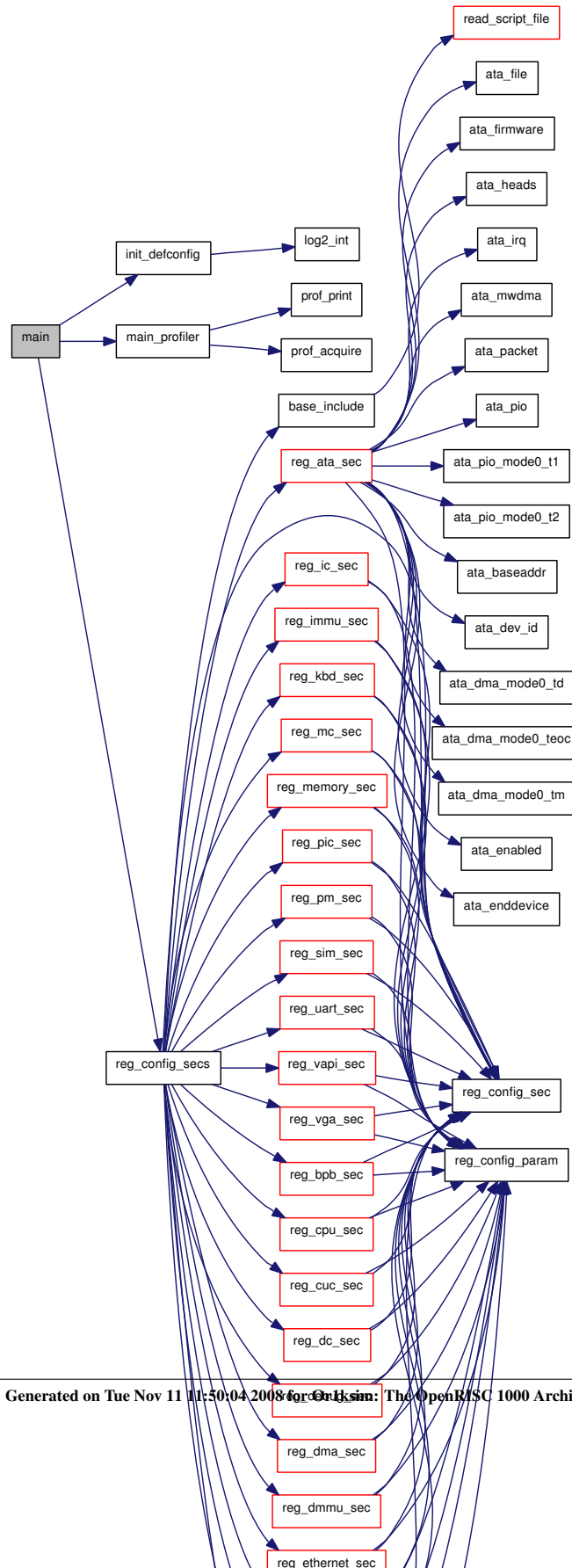
Functions

- `int main (int argc, char *argv[])`

6.167.1 Function Documentation

6.167.1.1 `int main (int argc, char * argv[])`

Here is the call graph for this function:



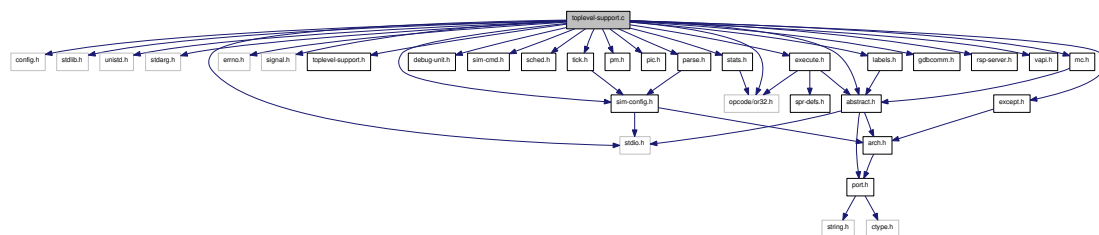
6.168 toplevel-support.c File Reference

```

#include "config.h"
#include <stdlib.h>
#include <unistd.h>
#include <stdarg.h>
#include <stdio.h>
#include <errno.h>
#include <signal.h>
#include "toplevel-support.h"
#include "sim-config.h"
#include "debug-unit.h"
#include "sim-cmd.h"
#include "sched.h"
#include "tick.h"
#include "pm.h"
#include "pic.h"
#include "execute.h"
#include "labels.h"
#include "stats.h"
#include "opcode/or32.h"
#include "parse.h"
#include "gdbcomm.h"
#include "rsp-server.h"
#include "vapi.h"
#include "abstract.h"
#include "mc.h"
#include "except.h"

```

Include dependency graph for toplevel-support.c:



Data Structures

- struct [sim_reset_hook](#)

Functions

- void [ctrl_c](#) (int signum)
- void [check_int](#) (void *dat)
- void [reg_sim_reset](#) (void(*reset_hook)(void *), void *dat)
- void [sim_reset](#) ()
- void [sim_init](#) ()
- void [sim_done](#) ()

Variables

- static struct [sim_reset_hook](#) * [sim_reset_hooks](#) = NULL

6.168.1 Function Documentation

6.168.1.1 void [check_int](#) (void * *dat*)

Routine poll to see if interaction is needed

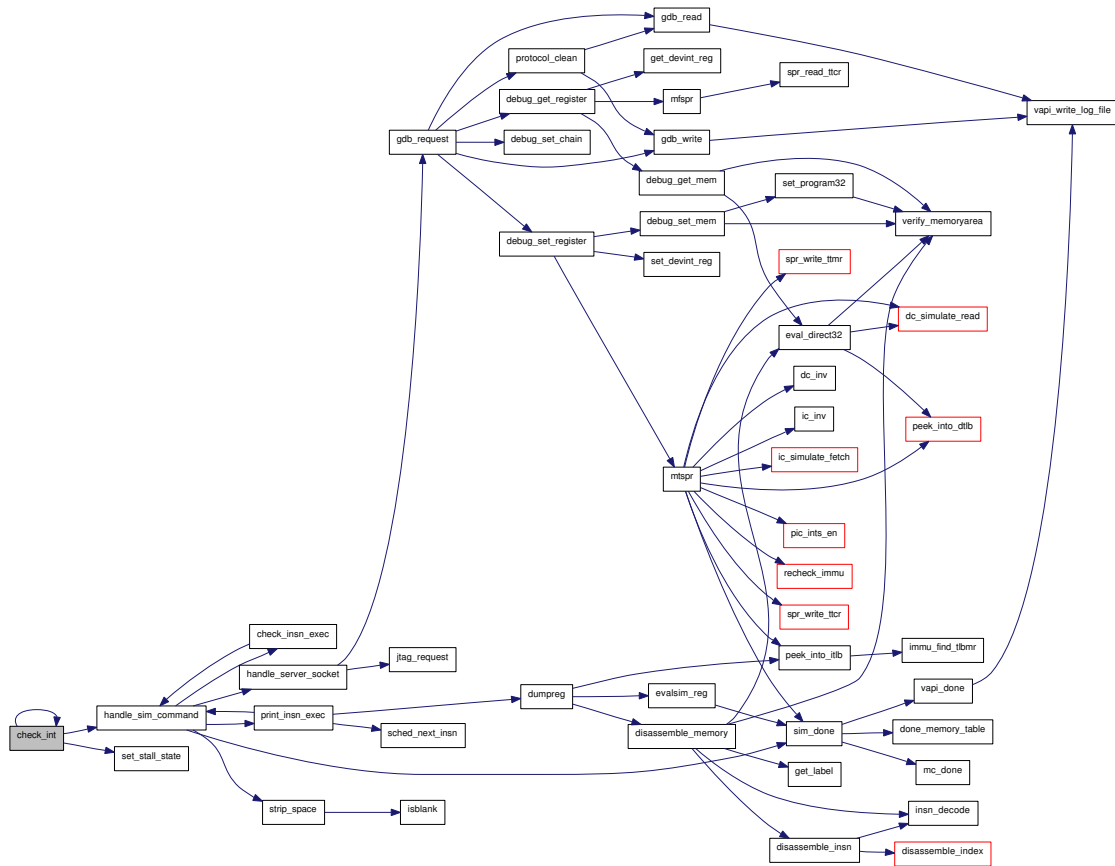
This is most likely to happen due to a ctrl-C. However when the -i flag is specified, the simulator starts up ready for interaction.

The main simulator loop will stop for interaction if it hits a breakpoint.

Parameters:

- ← *dat* Data passed in by the Or1ksim scheduler. Not needed by this function.

Here is the call graph for this function:



6.168.1.2 void ctrl_c(int signum)

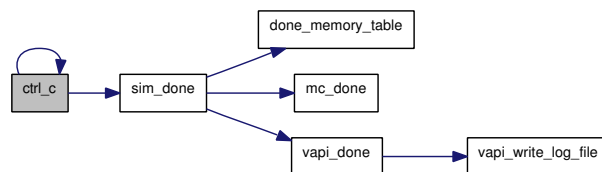
Signal handler for ctrl-C

Sets the iprompt flag, so the simulator will stop next time round the loop. If the iprompt flag is set when we enter here, that means the simulator has not reacted since the last ctrl-C, so we kill the simulation.

Parameters:

← *signum* The signal which triggered this handler

Here is the call graph for this function:



6.168.1.3 void reg_sim_reset (void(*) (void *) reset_hook, void * dat)

Register a new reset hook

The registered functions will be called in turn, whenever the simulation is reset by calling [sim_reset\(\)](#).

Parameters:

← *reset_hook* The function to be called on reset

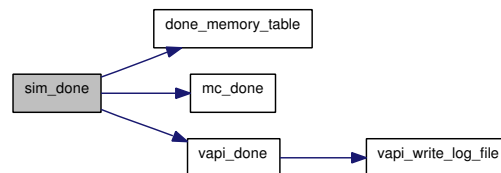
← *dat* The data structure to be passed as argument when the reset_hook function is called.

6.168.1.4 void sim_done ()

Clean up

Close an profile or log files, disconnect VAPI. Call any memory mapped peripheral close down function. Exit with rc 0.

Here is the call graph for this function:



6.168.1.5 void sim_init ()

Initialize the simulator

Reset internal data: symbol table (aka labels), breakpoints and stats. Rebuild the FSA's used for disassembly.

Initialize the [dynamic](#) execution system if required.

Initialize the scheduler.

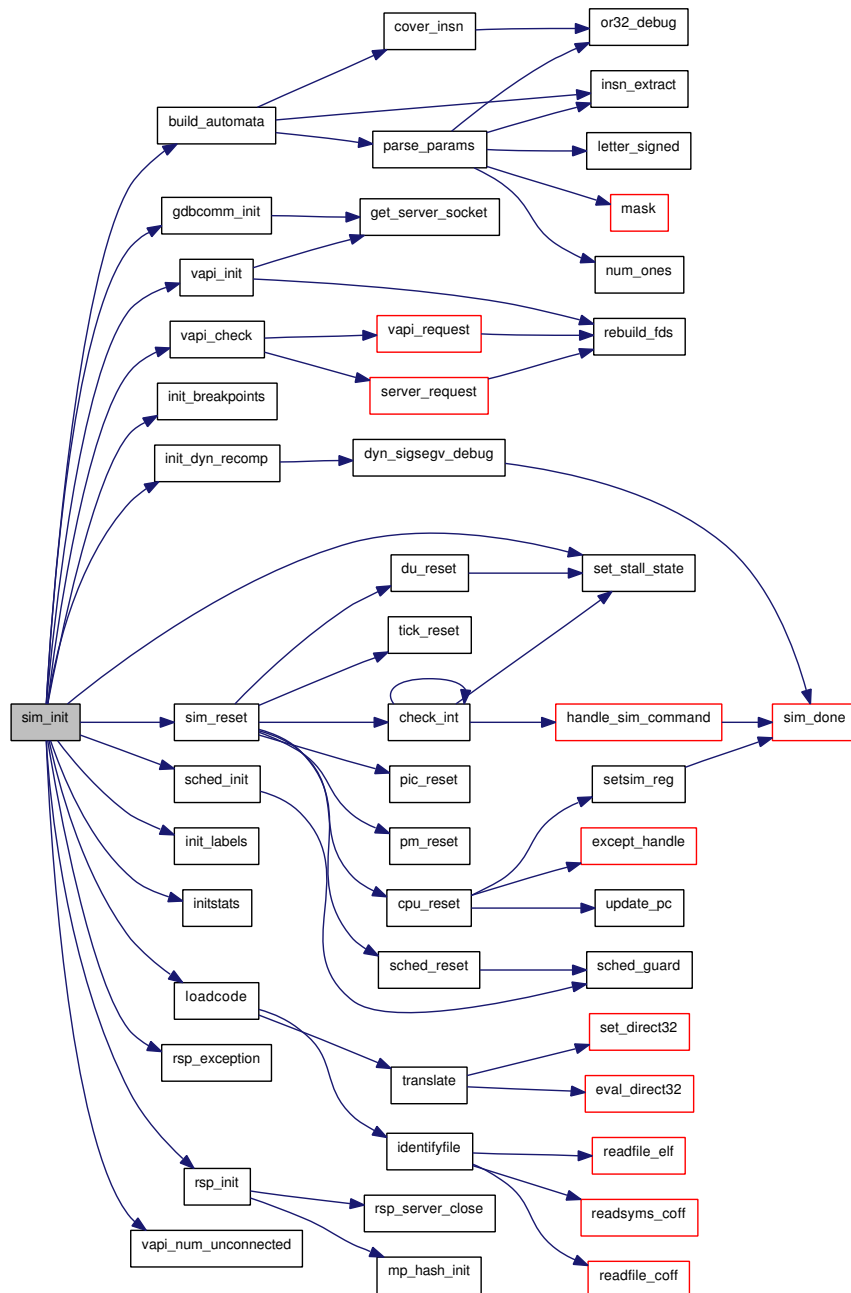
Open the various logs and statistics files requested by the configuration and/or command arguments.

Initialize GDB and VAPI connections.

Reset the simulator.

Wait for VAPI to connect if configured.

Here is the call graph for this function:



6.168.1.6 void sim_reset ()

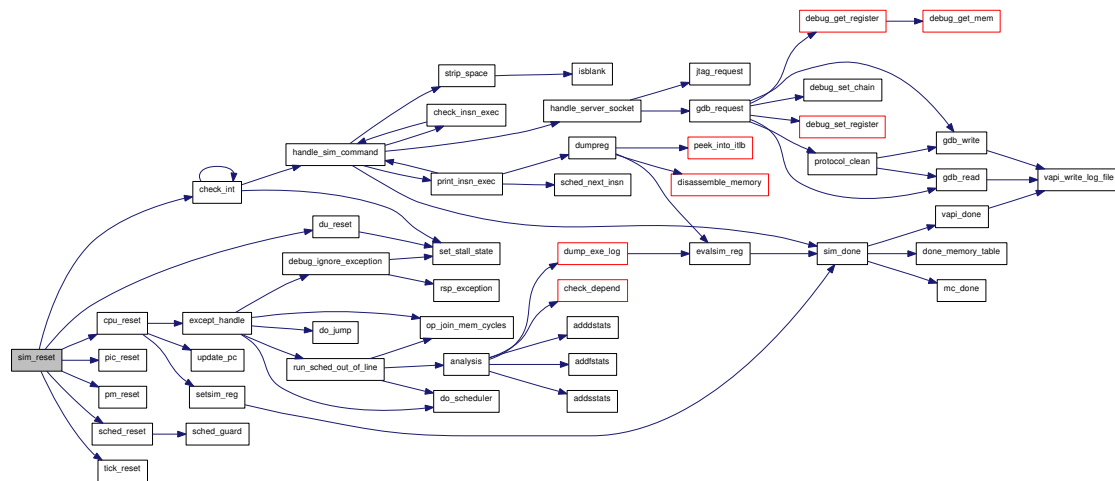
Reset the simulator

The scheduler is reset, then all reset functions on the reset hook list (i.e. peripherals) are reset. Then standard core functions (which do not use reset hooks) are reset: tick timer, power management, programmable interrupt controller and debug unit.

The scheduler queue is reinitialized with an immediate check for ctrl-C on its queue.

Finally the count of simulated cycles is set to zero, and the CPU itself is reset.

Here is the call graph for this function:



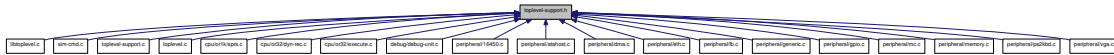
6.168.2 Variable Documentation

6.168.2.1 struct sim_reset_hook* sim_reset_hooks = NULL [static]

The list of reset hooks. Local to this source file

6.169 toplevel-support.h File Reference

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



Functions

- void [ctrl_c](#) (int signum)
- void [reg_sim_reset](#) (void(*reset_hook)(void *), void *dat)
- void [sim_done](#) ()
- void [check_int](#) (void *dat)
- void [sim_reset](#) ()
- void [sim_init](#) ()

6.169.1 Function Documentation

6.169.1.1 void [check_int](#) (void * *dat*)

Routine poll to see if interaction is needed

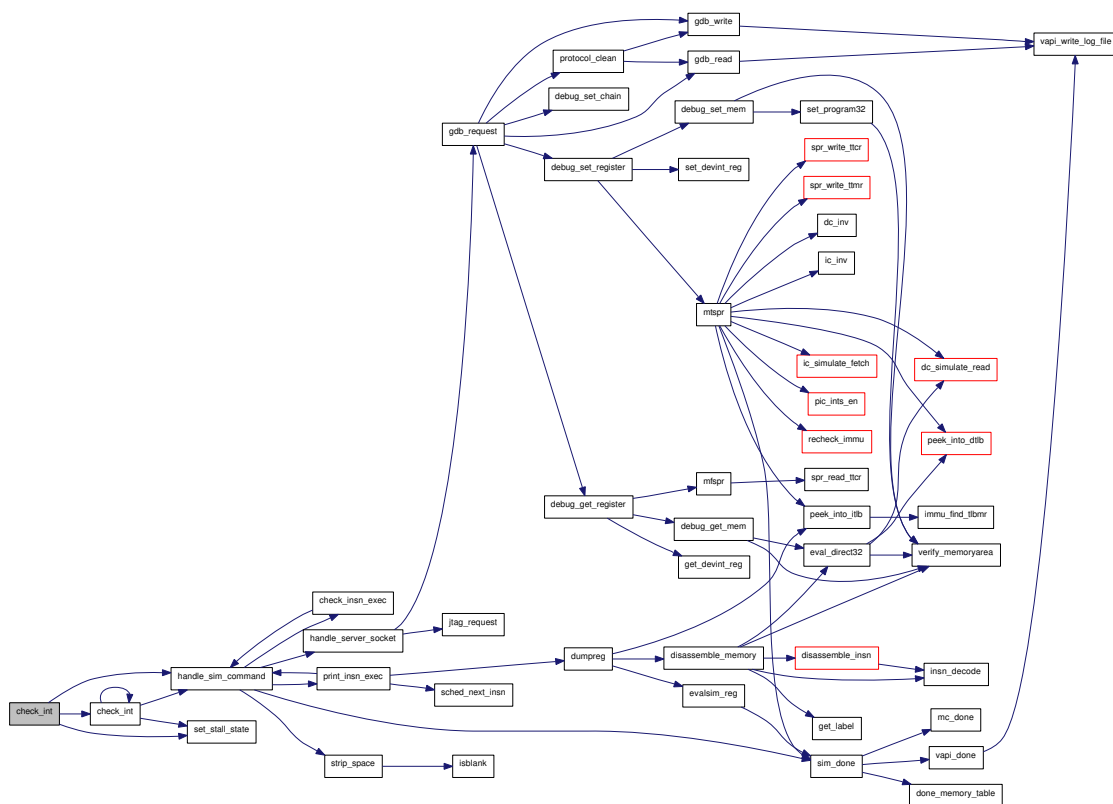
This is most likely to happen due to a ctrl-C. However when the `-i` flag is specified, the simulator starts up ready for interaction.

The main simulator loop will stop for interaction if it hits a breakpoint.

Parameters:

- ← *dat* Data passed in by the Or1ksim scheduler. Not needed by this function.

Here is the call graph for this function:



6.169.1.2 void ctrl_c (int *signum*)

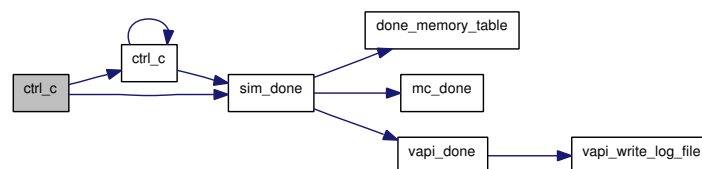
Signal handler for ctrl-C

Sets the iprompt flag, so the simulator will stop next time round the loop. If the iprompt flag is set when we enter here, that means the simulator has not reacted since the last ctrl-C, so we kill the simulation.

Parameters:

← *signum* The signal which triggered this handler

Here is the call graph for this function:



6.169.1.3 void reg_sim_reset (void(*) (void *) *reset_hook*, void * *dat*)

Register a new reset hook

The registered functions will be called in turn, whenever the simulation is reset by calling `sim_reset()`.

Parameters:

← *reset_hook* The function to be called on reset

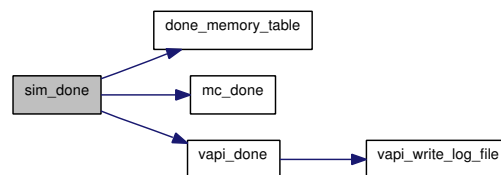
← *dat* The data structure to be passed as argument when the `reset_hook` function is called.

6.169.1.4 void sim_done ()

Clean up

Close an profile or log files, disconnect VAPI. Call any memory mapped peripheral close down function. Exit with rc 0.

Here is the call graph for this function:

**6.169.1.5 void sim_init ()**

Initialize the simulator

Reset internal data: symbol table (aka labels), breakpoints and stats. Rebuild the FSA's used for disassembly.

Initialize the `dynamic` execution system if required.

Initialize the scheduler.

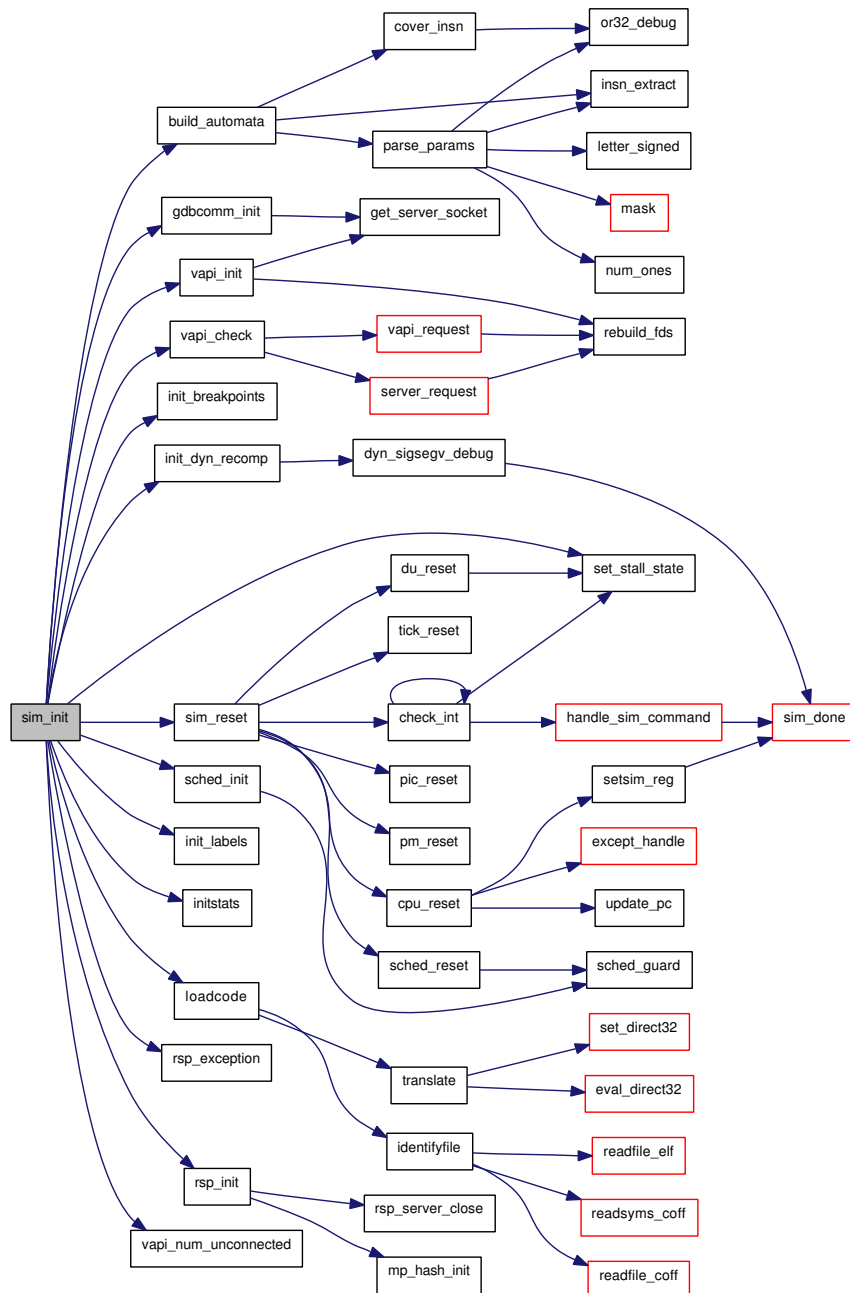
Open the various logs and statistics files requested by the configuration and/or command arguments.

Initialize GDB and VAPI connections.

Reset the simulator.

Wait for VAPI to connect if configured.

Here is the call graph for this function:



6.169.1.6 void sim_reset ()

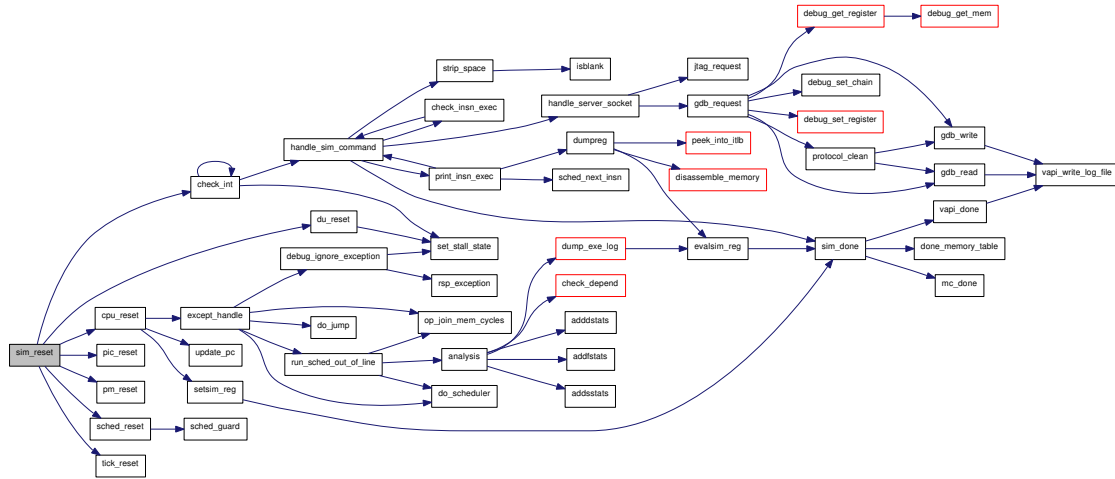
Reset the simulator

The scheduler is reset, then all reset functions on the reset hook list (i.e. peripherals) are reset. Then standard core functions (which do not use reset hooks) are reset: tick timer, power management, programmable interrupt controller and debug unit.

The scheduler queue is reinitialized with an immediate check for ctrl-C on its queue.

Finally the count of simulated cycles is set to zero, and the CPU itself is reset.

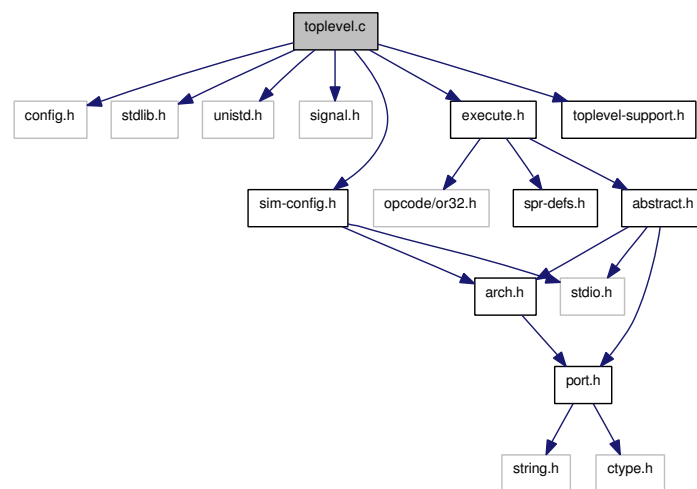
Here is the call graph for this function:



6.170 toplevel.c File Reference

```
#include "config.h"
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include "sim-config.h"
#include "toplevel-support.h"
#include "execute.h"
```

Include dependency graph for toplevel.c:



Functions

- `int main (int argc, char *argv[])`

6.170.1 Function Documentation

6.170.1.1 `int main (int argc, char * argv[])`

Main function

Set up the standalone simulation. Initialize the default configuration and register all the sections that may appear in a user configuration.

Then attempt to parse the args, configure the system from any configuration file specified and print out the configuration used.

Add a signal handler, so ctrl-C will drop the user into the CLI.

The initialize the simulator, call the appropriate main simulator function and when it returns tidy up.

Parameters:

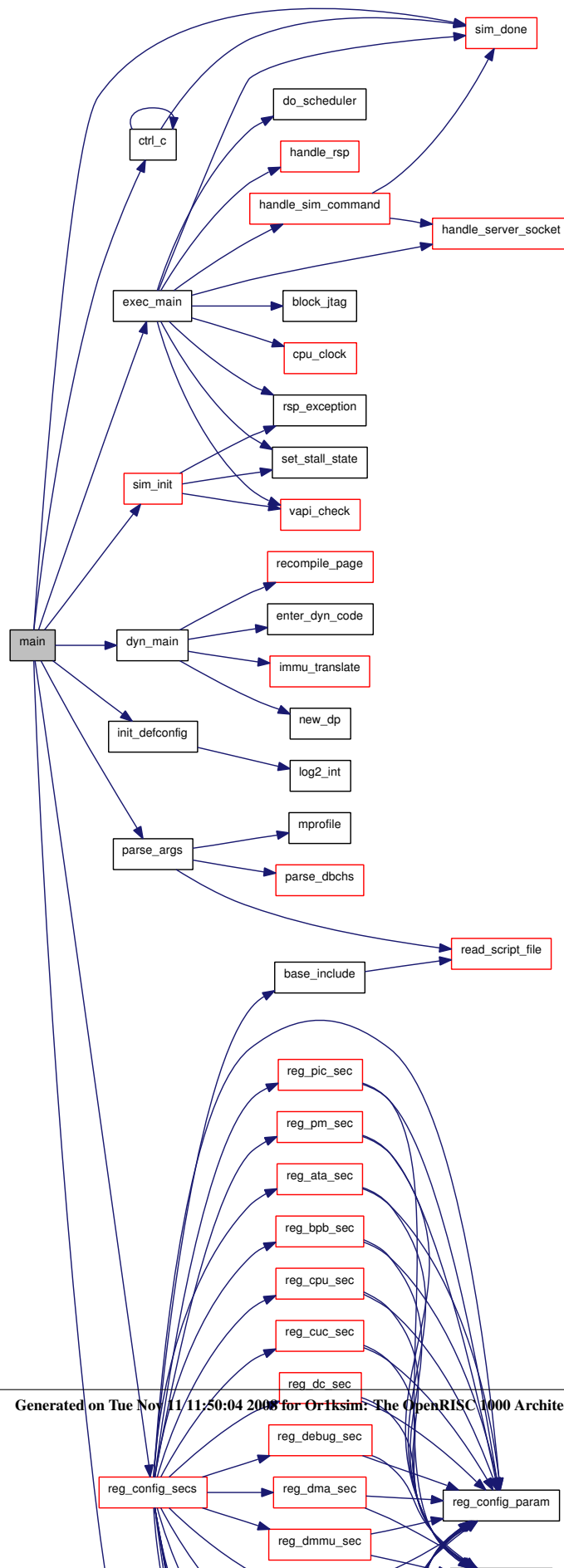
- ← *argc* The number of arguments to the command

← *argv* The vector of argument strings

Returns:

The return code required from the simulator. This is actually achieved by calling `exit()` with the return code, rather than returning an explicit value.

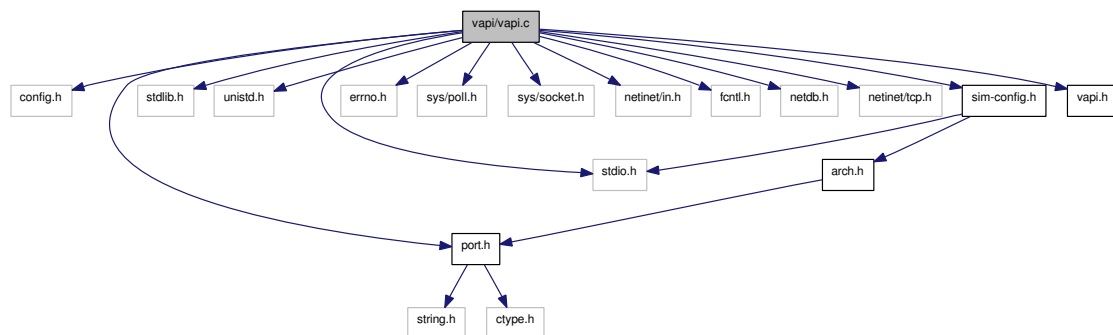
Here is the call graph for this function:



6.171 vapi/vapi.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <sys/poll.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <netdb.h>
#include <netinet/tcp.h>
#include "sim-config.h"
#include "vapi.h"
```

Include dependency graph for vapi.c:



Data Structures

- struct [vapi_handler](#)

Functions

- void [rebuild_fds](#) ()
- static int [handler_fits_id](#) (const struct [vapi_handler](#) *t, unsigned long id)
- static struct [vapi_handler](#) * [find_handler](#) (unsigned long id)
- static struct [vapi_handler](#) * [add_handler](#) (unsigned long base_id, unsigned long num_ids)
- void [vapi_write_log_file](#) (VAPI_COMMAND command, unsigned long devid, unsigned long data)
- static int [vapi_write_stream](#) (int fd, void *buf, int len)
- static int [vapi_read_stream](#) (int fd, void *buf, int len)
- int [get_server_socket](#) (const char *name, const char *proto, int port)

- static void `server_request` ()
- static int `write_packet` (unsigned long id, unsigned long data)
- static int `read_packet` (int fd, unsigned long *id, unsigned long *data)
- static void `vapi_request` (struct `vapi_handler` *t)
- void `vapi_check` ()
- int `vapi_init` ()
- void `vapi_done` ()
- void `vapi_install_handler` (unsigned long id, void(*read_func)(unsigned long, unsigned long, void *), void *dat)
- void `vapi_install_multi_handler` (unsigned long base_id, unsigned long num_ids, void(*read_func)(unsigned long, unsigned long, void *), void *dat)
- int `vapi_num_unconnected` (int printout)
- void `vapi_send` (unsigned long id, unsigned long data)
- static void `vapi_enabled` (union `param_val` val, void *dat)
- static void `vapi_server_port` (union `param_val` val, void *dat)
- static void `vapi_log_enabled` (union `param_val` val, void *dat)
- static void `vapi_hide_device_id` (union `param_val` val, void *dat)
- static void `vapi_log_fn` (union `param_val` val, void *dat)
- void `reg_vapi_sec` (void)

Variables

- static unsigned int `serverIP` = 0
- static unsigned int `server_fd` = 0
- static unsigned int `nhandlers` = 0
- static int `tcp_level` = 0
- static struct pollfd * `fds` = NULL
- static int `nfds` = 0

6.171.1 Function Documentation

6.171.1.1 static struct `vapi_handler`* `add_handler` (unsigned long *base_id*, unsigned long *num_ids*)
[static, read]

Here is the call graph for this function:



6.171.1.2 static struct `vapi_handler`* `find_handler` (unsigned long *id*) [static, read]

Here is the call graph for this function:



6.171.1.3 `int get_server_socket (const char * name, const char * proto, int port)`

6.171.1.4 `static int handler_fits_id (const struct vapi_handler * t, unsigned long id)` [static]

6.171.1.5 `static int read_packet (int fd, unsigned long * id, unsigned long * data)` [static]

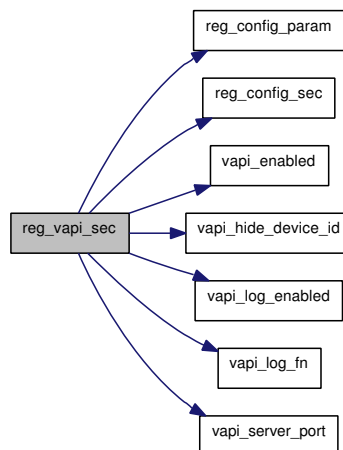
Here is the call graph for this function:



6.171.1.6 `void rebuild_fds ()`

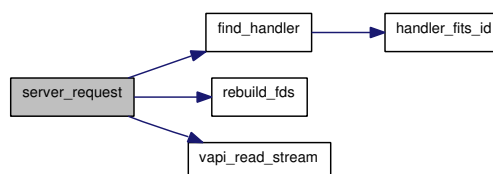
6.171.1.7 `void reg_vapi_sec (void)`

Here is the call graph for this function:



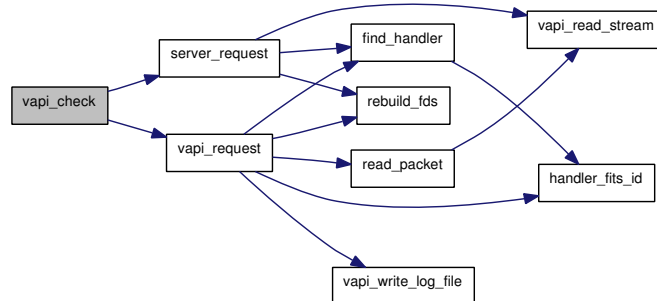
6.171.1.8 `static void server_request ()` [static]

Here is the call graph for this function:



6.171.1.9 void vapi_check ()

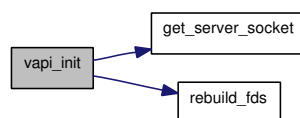
Here is the call graph for this function:

**6.171.1.10 void vapi_done ()**

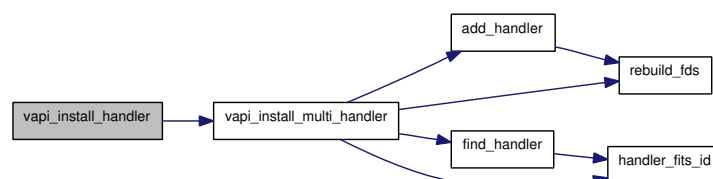
Here is the call graph for this function:

**6.171.1.11 static void vapi_enabled (union param_val val, void * dat) [static]****6.171.1.12 static void vapi_hide_device_id (union param_val val, void * dat) [static]****6.171.1.13 int vapi_init ()**

Here is the call graph for this function:

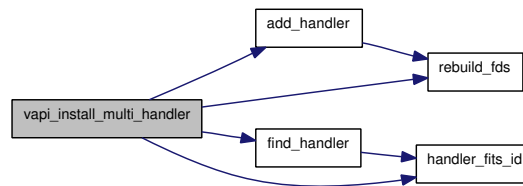
**6.171.1.14 void vapi_install_handler (unsigned long id, void(*) (unsigned long, unsigned long, void *) read_func, void * dat)**

Here is the call graph for this function:



6.171.1.15 void vapi_install_multi_handler (unsigned long *base_id*, unsigned long *num_ids*, void(*) (unsigned long, unsigned long, void *) *read_func*, void * *dat*)

Here is the call graph for this function:



6.171.1.16 static void vapi_log_enabled (union param_val *val*, void * *dat*) [static]

6.171.1.17 static void vapi_log_fn (union param_val *val*, void * *dat*) [static]

Set the log file

Free any existing string.

Parameters:

← *val* The value to use

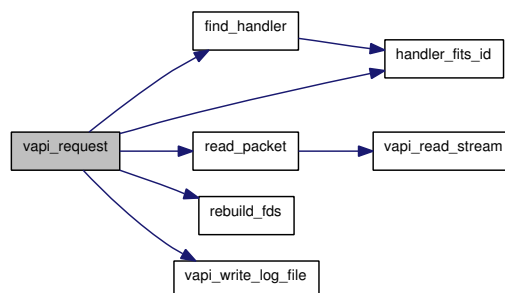
← *dat* The [config](#) data structure (not used here)

6.171.1.18 int vapi_num_unconnected (int *printout*)

6.171.1.19 static int vapi_read_stream (int *fd*, void * *buf*, int *len*) [static]

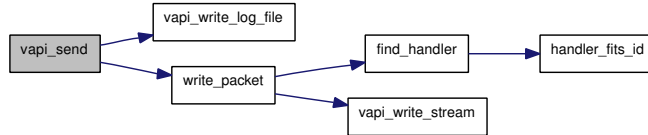
6.171.1.20 static void vapi_request (struct vapi_handler * *t*) [static]

Here is the call graph for this function:



6.171.1.21 void vapi_send (unsigned long *id*, unsigned long *data*)

Here is the call graph for this function:



6.171.1.22 static void vapi_server_port (union param_val *val*, void * *dat*) [static]

Set the VAPI server port

Ensure the value chosen is valid

Parameters:

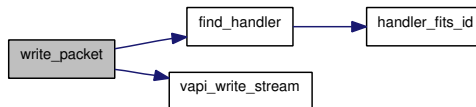
- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

6.171.1.23 void vapi_write_log_file (VAPI_COMMAND *command*, unsigned long *devid*, unsigned long *data*)

6.171.1.24 static int vapi_write_stream (int *fd*, void * *buf*, int *len*) [static]

6.171.1.25 static int write_packet (unsigned long *id*, unsigned long *data*) [static]

Here is the call graph for this function:



6.171.2 Variable Documentation

6.171.2.1 struct pollfd* fds = NULL [static]

6.171.2.2 int nfds = 0 [static]

6.171.2.3 unsigned int nhandlers = 0 [static]

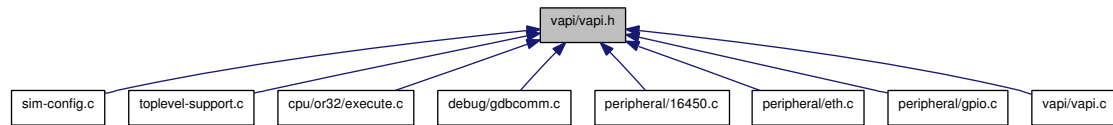
6.171.2.4 unsigned int server_fd = 0 [static]

6.171.2.5 unsigned int serverIP = 0 [static]

6.171.2.6 int tcp_level = 0 [static]

6.172 vapi/vapi.h File Reference

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



Defines

- #define `VAPI_MAX_DEVID` 0xFFFF

Enumerations

- enum `VAPI_COMMAND` { `VAPI_COMMAND_REQUEST` = 0, `VAPI_COMMAND_SEND` = 1, `VAPI_COMMAND_END` = 2 }

Functions

- int `vapi_init` ()
- void `vapi_done` ()
- void `vapi_install_handler` (unsigned long id, void(*read_func)(unsigned long, unsigned long, void *), void *dat)
- void `vapi_install_multi_handler` (unsigned long base_id, unsigned long num_ids, void(*read_func)(unsigned long, unsigned long, void *), void *dat)
- void `vapi_check` ()
- int `vapi_num_unconnected` (int printout)
- void `vapi_send` (unsigned long id, unsigned long data)
- void `vapi_write_log_file` (`VAPI_COMMAND` command, unsigned long device_id, unsigned long data)
- void `reg_vapi_sec` ()

6.172.1 Define Documentation

6.172.1.1 #define `VAPI_MAX_DEVID` 0xFFFF

Maximum value for VAPI device id

6.172.2 Enumeration Type Documentation

6.172.2.1 enum `VAPI_COMMAND`

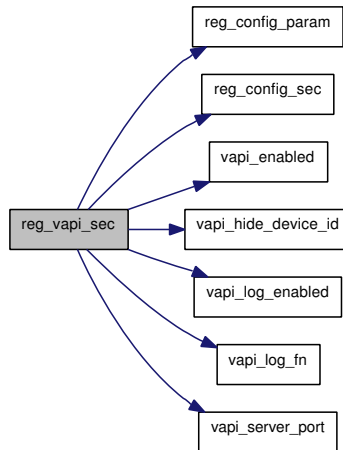
Enumerator:

`VAPI_COMMAND_REQUEST`
`VAPI_COMMAND_SEND`
`VAPI_COMMAND_END`

6.172.3 Function Documentation

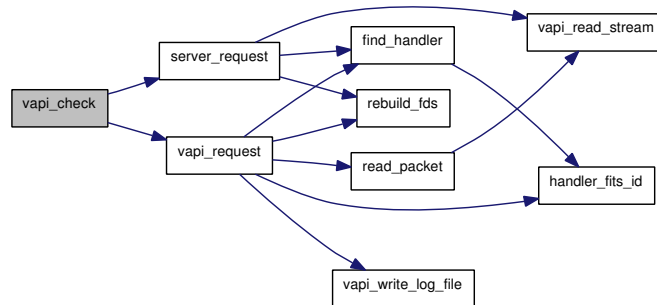
6.172.3.1 void reg_vapi_sec ()

Here is the call graph for this function:



6.172.3.2 void vapi_check ()

Here is the call graph for this function:



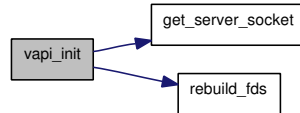
6.172.3.3 void vapi_done ()

Here is the call graph for this function:

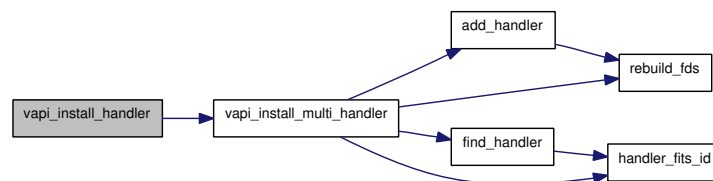


6.172.3.4 int vapi_init ()

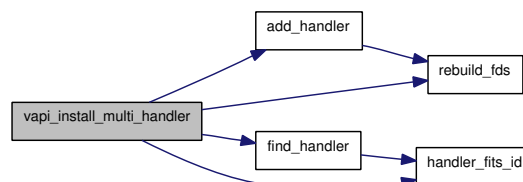
Here is the call graph for this function:

**6.172.3.5 void vapi_install_handler (unsigned long id, void(*) (unsigned long, unsigned long, void *) read_func, void * dat)**

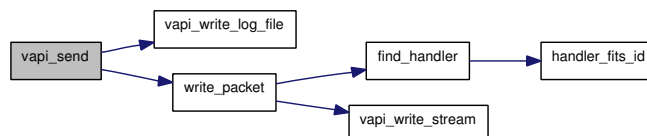
Here is the call graph for this function:

**6.172.3.6 void vapi_install_multi_handler (unsigned long base_id, unsigned long num_ids, void(*) (unsigned long, unsigned long, void *) read_func, void * dat)**

Here is the call graph for this function:

**6.172.3.7 int vapi_num_unconnected (int printout)****6.172.3.8 void vapi_send (unsigned long id, unsigned long data)**

Here is the call graph for this function:



6.172.3.9 void vapi_write_log_file (VAPI_COMMAND *command*, unsigned long *device_id*, unsigned long *data*)

Index

- `__ORSIM_DBCL_ERR`
 - debug.h, 820
- `__ORSIM_DBCL_FIXME`
 - debug.h, 820
- `__ORSIM_DBCL_TRACE`
 - debug.h, 820
- `__ORSIM_DBCL_WARN`
 - debug.h, 820
- `__DYNAMIC`
 - elf.h, 260
- `__ORSIM_DBG_USE_FUNC`
 - debug.h, 819
- `__ORSIM_DEBUG_CLASS`
 - debug.h, 820
- `__ORSIM_DEBUG_LOG`
 - debug.h, 819
- `__ORSIM_DPRINTF`
 - debug.h, 819
- `__ORSIM_GET_DEBUGGING`
 - debug.h, 819
- `__ORSIM_GET_DEBUGGING_ERR`
 - debug.h, 819
- `__ORSIM_GET_DEBUGGING_FIXME`
 - debug.h, 819
- `__ORSIM_GET_DEBUGGING_TRACE`
 - debug.h, 819
- `__ORSIM_GET_DEBUGGING_WARN`
 - debug.h, 819
- `__ORSIM_NO_DEC_DBCH`
 - debug.c, 817
- `__op_param1`
 - dyn-rec.c, 350
 - op.c, 429
- `__op_param2`
 - dyn-rec.c, 350
 - op.c, 429
- `__op_param3`
 - dyn-rec.c, 350
 - op.c, 429
- `__or_dynop`
 - op.c, 423
- `__orsim_dbchs`
 - debug.c, 817
- `_csm_list`, 13
 - cmatch, 14
- cmovs, 14
- cnt, 14
- dead, 14
- from, 14
- next, 14
- ninsn, 14
- osize, 14
- ref, 14
- size, 14
- `_cuc_func`, 15
 - bb, 16
 - end_addr, 16
 - fdeps, 16
 - init_bb_reloc, 16
 - lur, 16
 - memory_order, 16
 - msched, 16
 - mtype, 16
 - nfdeps, 16
 - nmsched, 16
 - num_bb, 16
 - num_init_bb, 16
 - num_runs, 16
 - orig_time, 16
 - saved_regs, 16
 - start_addr, 16
 - timings, 16
 - tmp, 16
 - used_regs, 16
- `_dep_list_t`, 17
 - next, 17
 - ref, 17
- 16450.c
 - char_clks, 604
 - MAX_SKEW, 598
 - MIN, 598
 - reg_uart_sec, 604
 - send_char, 605
 - uart_16550, 605
 - uart_add_char, 605
 - UART_ADDR_SPACE, 598
 - uart_baseaddr, 605
 - UART_BREAK_COUNT, 598
 - uart_channel, 605
 - uart_char_clock, 605

- UART_CHAR_TIMEOUT, 598
- uart_check_char, 606
- uart_check_rdi, 606
- uart_check_rlsi, 606
- uart_clear_int, 607
- UART_CLOCK_DIVIDER, 598
- UART_DLH, 598
- UART_DLL, 598
- uart_enabled, 607
- UART_FCR, 598
- UART_FCR_FIE, 598
- UART_FCR_RRXFI, 598
- UART_FCR_RTXFI, 598
- UART_FGETC_SLOWDOWN, 598
- UART_FIFO_TRIGGER, 598
- UART_IER, 599
- UART_IER_MSI, 601
- UART_IER_RDI, 601
- UART_IER_RLSI, 601
- UART_IER_THRI, 601
- UART_IIR, 601
- UART_IIR_CTI, 601
- UART_IIR_ID, 601
- UART_IIR_MSI, 601
- UART_IIR_NO_INT, 601
- UART_IIR_RDI, 601
- UART_IIR_RLSI, 601
- UART_IIR_THRI, 601
- uart_int_cti, 608
- uart_int_msi, 608
- uart_int_rdi, 608
- uart_int_rlsi, 609
- uart_int_thri, 609
- uart_irq, 610
- uart_jitter, 610
- UART_LCR, 601
- UART_LCR_DLAB, 601
- UART_LCR_EPAR, 601
- UART_LCR_PARITY, 601
- UART_LCR_RESET, 601
- UART_LCR_SBC, 601
- UART_LCR_SPAR, 601
- UART_LCR_STOP, 601
- UART_LCR_WLEN5, 601
- UART_LCR_WLEN6, 601
- UART_LCR_WLEN7, 601
- UART_LCR_WLEN8, 601
- uart_loopback, 610
- UART_LSR, 601
- UART_LSR_BREAK, 601
- UART_LSR_FRAME, 601
- UART_LSR_OVRRUN, 601
- UART_LSR_PARITY, 601
- UART_LSR_RDRDY, 601
- UART_LSR_RXERR, 601
- UART_LSR_TXBUFE, 601
- UART_LSR_TXSERE, 601
- UART_MAX_FIFO_LEN, 601
- UART_MCR, 601
- UART_MCR_AUX1, 602
- UART_MCR_AUX2, 602
- UART_MCR_DTR, 602
- UART_MCR_LOOP, 602
- UART_MCR_RTS, 602
- UART_MSR, 602
- UART_MSR_CTS, 602
- UART_MSR_DCD, 602
- UART_MSR_DCTS, 602
- UART_MSR_DDCD, 602
- UART_MSR_DDSR, 602
- UART_MSR_DSR, 602
- UART_MSR_RI, 602
- UART_MSR_TERI, 602
- uart_newway, 610
- uart_next_int, 611
- uart_read_byte, 611
- uart_recv_break, 611
- uart_recv_break_start, 611
- uart_recv_break_stop, 612
- uart_recv_char, 612
- uart_reset, 612
- UART_RXBUF, 602
- uart_sched_recv_check, 613
- UART_SCR, 602
- uart_sec_end, 613
- uart_sec_start, 614
- uart_send_break, 614
- uart_status, 615
- uart_tx_send, 615
- UART_TXBUF, 602
- UART_VALID_FCR, 602
- UART_VALID_IER, 602
- UART_VALID_IIR, 602
- UART_VALID_LCR, 602
- UART_VALID_LSR, 602
- UART_VALID_MCR, 602
- UART_VALID_MSR, 602
- UART_VAPI_BUF_LEN, 602
- uart_vapi_cmd, 615
- uart_vapi_id, 616
- uart_vapi_read, 616
- uart_write_byte, 616
- 16450.h
 - reg_uart_sec, 619
 - uart_reset, 620
 - uart_status, 620
- a0

- dma_channel, 81
- a1
 - dma_channel, 81
- abstract.c
 - adjust_rw_delay, 213
 - bit_mask, 213
 - cur_area, 228
 - cur_vadd, 228
 - data_ci, 228
 - dev_list, 228
 - disassemble_memory, 213
 - done_memory_table, 213
 - dump_memory, 213
 - eval_direct16, 214
 - eval_direct32, 214
 - eval_direct8, 215
 - eval_insn, 215
 - eval_mem16, 215
 - eval_mem32, 216
 - eval_mem8, 216
 - eval_mem_16_inv, 217
 - eval_mem_16_inv_direct, 217
 - eval_mem_32_inv, 218
 - eval_mem_32_inv_direct, 218
 - eval_mem_8_inv, 218
 - eval_mem_8_inv_direct, 218
 - evalsim_mem16, 219
 - evalsim_mem32, 219
 - evalsim_mem8, 219
 - generate_time_pretty, 220
 - insn_ci, 228
 - mc_area, 228
 - memory_table_status, 220
 - reg_mem_area, 220
 - register_memoryarea_mask, 221
 - set_direct16, 221
 - set_direct32, 222
 - set_direct8, 222
 - set_mem16, 223
 - set_mem32, 223
 - set_mem8, 224
 - set_mem_16_inv, 224
 - set_mem_16_inv_direct, 225
 - set_mem_32_inv, 225
 - set_mem_32_inv_direct, 225
 - set_mem_8_inv, 226
 - set_mem_8_inv_direct, 226
 - set_mem_valid, 226
 - set_program32, 226
 - set_program8, 226
 - setsim_mem16, 226
 - setsim_mem32, 227
 - setsim_mem8, 227
 - verify_memoryarea, 228
- abstract.h
 - adjust_rw_delay, 231
 - CT_NONE, 231
 - CT_PHYSICAL, 231
 - CT_VIRTUAL, 231
 - cur_area, 243
 - data_ci, 243
 - DEFAULT_MEMORY_LEN, 231
 - DEFAULT_MEMORY_START, 231
 - disassemble_memory, 231
 - done_memory_table, 232
 - dump_memory, 232
 - eval_direct16, 232
 - eval_direct32, 233
 - eval_direct8, 233
 - eval_insn, 234
 - eval_mem16, 234
 - eval_mem32, 235
 - eval_mem8, 235
 - evalsim_mem16, 236
 - evalsim_mem32, 236
 - evalsim_mem8, 237
 - generate_time_pretty, 237
 - hist_exec_tail, 243
 - HISTEXEC_LEN, 231
 - insn_ci, 243
 - INSNAME_LEN, 231
 - LABELNAME_LEN, 231
 - LE16, 231
 - LONGEST, 231
 - MAX_OPERANDS, 231
 - memory_table_status, 237
 - OP_MEM_ACCESS, 231
 - OPERANDNAME_LEN, 231
 - reg_mem_area, 237
 - set_direct16, 238
 - set_direct32, 238
 - set_direct8, 239
 - set_mem16, 239
 - set_mem32, 240
 - set_mem8, 240
 - set_mem_valid, 241
 - set_program32, 241
 - set_program8, 241
 - setsim_mem16, 241
 - setsim_mem32, 242
 - setsim_mem8, 242
 - STACK_SIZE, 231
 - ULONGEST, 231
 - verify_memoryarea, 243
- add_breakpoint
 - labels.c, 269
 - labels.h, 271
- add_crc

- eth_device, 113
- add_data_dep
 - cuc.h, 469
 - insn.c, 483
- add_dep
 - cuc.h, 470
 - insn.c, 483
- add_handler
 - vapi.c, 866
- add_label
 - labels.c, 269
 - labels.h, 271
- add_latches
 - cuc.h, 470
 - insn.c, 483
- add_memory_dep
 - cuc.h, 470
 - cuc/memory.c, 500
- add_to_op_params
 - dyn-rec.c, 335
 - dyn-rec.h, 353
- add_to_opq
 - dyn-rec.c, 335
 - dyn-rec.h, 353
- addstats
 - stats.c, 281
 - stats.h, 285
- addend
 - reloc, 161
- addfstats
 - stats.c, 281
 - stats.h, 285
- addprogram
 - parse.c, 273
- addr
 - bpb_entry, 30
 - breakpoint_entry, 33
 - btic_entry, 34
 - fb_state, 118
 - func_struct, 122
 - hist_exec, 125
 - label_entry, 145
 - memory_hash, 153
 - mp_entry, 154
 - mprofentry_struct, 155
 - stack_struct, 175
- ADDR_C
 - arch.h, 290
- addr_compare
 - dev_memarea, 79
- addr_mask
 - dev_memarea, 79
- addr_val
 - param_val, 159
- address
 - jtr_read_block_message, 136
 - jtr_read_message, 138
 - jtr_write_block_message, 140
 - jtr_write_message, 142
- addstats
 - stats.c, 281
 - stats.h, 285
- adjust_rw_delay
 - abstract.c, 213
 - abstract.h, 231
- adv.c
 - calc_max, 441
 - detect_max_values, 441
 - insert_conditional_facts, 441
 - mark_successors, 441
 - mask, 441
 - max_op, 441
- am0
 - dma_channel, 81
- am1
 - dma_channel, 81
- analyse_function
 - cuc.c, 451
- analyse_timings
 - cuc.h, 470
 - timings.c, 508
- analysis
 - execute.c, 367
 - execute.h, 262
- apply_edge_condition
 - insn.c, 483
- arch.h
 - ADDR_C, 290
 - LINK_REGNO, 290
 - oraddr_t, 291
 - orreg_t, 291
 - PRIdREG, 290
 - PRiXADDR, 290
 - PRiXREG, 291
 - REG_C, 291
 - uorreg_t, 291
- archf, 18
 - gen_func_reloc, 18
 - gen_reloc, 18
 - get_real_func_len, 18
- archfs
 - dyngen-i386.c, 360
 - dyngen.h, 365
- argv
 - xterm_channel, 181
- ascii2hex
 - rsp-server.c, 541
- asm

- op-i386.h, [403](#)
- op.c, [423](#)
- ASSIGN_FLAG
 - fields.h, [720](#)
- AT_BASE
 - elf.h, [257](#)
- AT_EGID
 - elf.h, [257](#)
- AT_ENTRY
 - elf.h, [257](#)
- AT_EUID
 - elf.h, [257](#)
- AT_EXECPD
 - elf.h, [257](#)
- AT_FLAGS
 - elf.h, [257](#)
- AT_GID
 - elf.h, [257](#)
- AT_IGNORE
 - elf.h, [257](#)
- AT_NOTELF
 - elf.h, [257](#)
- AT_NULL
 - elf.h, [257](#)
- AT_PAGESZ
 - elf.h, [257](#)
- AT_PHDR
 - elf.h, [257](#)
- AT_PHEMT
 - elf.h, [257](#)
- AT_PHNUM
 - elf.h, [257](#)
- AT_UID
 - elf.h, [257](#)
- ATA_ADDR_SPACE
 - atahost.h, [657](#)
- ATA_ASR
 - atadevice.h, [642](#)
- ata_baseaddr
 - atahost.c, [646](#)
- ATA_BELECO
 - atahost.h, [657](#)
- ATA_BELEC1
 - atahost.h, [657](#)
- ata_calc_lba
 - atadevice-cmdi.c, [626](#)
- ATA_CHR
 - atadevice.h, [642](#)
- ATA_CLR
 - atadevice.h, [642](#)
- ata_cmd_complete
 - atadevice-cmdi.c, [626](#)
- ATA_CR
 - atadevice.h, [642](#)
- ATA_CTRL
 - atahost.h, [657](#)
- ATA_DA
 - atadevice.h, [642](#)
- ATA_DAR_DS0
 - atadevice.h, [642](#)
- ATA_DAR_DS1
 - atadevice.h, [642](#)
- ATA_DAR_H
 - atadevice.h, [642](#)
- ATA_DAR_WTG
 - atadevice.h, [642](#)
- ATA_DCR
 - atadevice.h, [642](#)
- ATA_DCR_IEN
 - atadevice.h, [642](#)
- ATA_DCR_RST
 - atadevice.h, [642](#)
- ata_dev_id
 - atahost.c, [646](#)
- ata_device, [19](#)
 - command, [22](#)
 - conf, [22](#)
 - cylinder_high, [22](#)
 - cylinder_low, [22](#)
 - daspi, [22](#)
 - daspo, [22](#)
 - dataport_i, [22](#)
 - dbuf, [22](#)
 - dbuf_cnt, [22](#)
 - dbuf_ptr, [22](#)
 - dev, [22](#)
 - device_control, [22](#)
 - device_head, [22](#)
 - dma_mode, [22](#)
 - dmarq, [22](#)
 - end_t_func, [22](#)
 - error, [22](#)
 - features, [22](#)
 - file, [22](#)
 - firmware, [22](#)
 - heads, [22](#)
 - heads_per_cylinder, [22](#)
 - host, [22](#)
 - internals, [22](#)
 - intrq, [22](#)
 - iordy, [22](#)
 - lba, [22](#)
 - mwdma, [22](#)
 - nr_sect, [22](#)
 - packet, [22](#)
 - pdiagi, [22](#)
 - pdiago, [22](#)
 - pio, [22](#)

- pio_mode, 22
- regs, 22
- sector_count, 22
- sector_number, 22
- sectors, 22
- sectors_per_track, 22
- sigs, 22
- size, 22
- size_sect, 22
- state, 22
- status, 22
- stream, 22
- type, 22
- ata_device_do_command_register
 - atadevice.c, 636
- ata_device_do_control_register
 - atadevice.c, 636
- ata_device_execute_cmd
 - atadevice-cmdi.c, 626
 - atadevice-cmdi.h, 633
- ata_device_hw_reset
 - atadevice.c, 636
- ata_device_init
 - atadevice.c, 636
- ata_device_reset_cmd
 - atadevice-cmdi.c, 626
- ata_device_write
 - atadevice.c, 636
- ata_devices, 24
 - device, 24
- ata_devices_hw_reset
 - atadevice.c, 637
 - atadevice.h, 642
- ata_devices_init
 - atadevice.c, 637
 - atadevice.h, 643
- ata_devices_read
 - atadevice.c, 637
 - atadevice.h, 643
- ata_devices_write
 - atadevice.c, 637
 - atadevice.h, 643
- ATA_DEVID
 - atahost.h, 657
- ATA_DHR
 - atadevice.h, 642
- ATA_DHR_DEV
 - atadevice.h, 642
- ATA_DHR_H
 - atadevice.h, 642
- ATA_DHR_LBA
 - atadevice.h, 642
- ata_dma_delay
 - atahost.h, 657
- ATA_DMA_EN
 - atahost.h, 657
- ata_dma_mode0_td
 - atahost.c, 646
- ata_dma_mode0_tec
 - atahost.c, 646
- ata_dma_mode0_tm
 - atahost.c, 646
- ATA_DMA_RD
 - atahost.h, 657
- ATA_DMA_TIP
 - atahost.h, 657
- ATA_DMA_WR
 - atahost.h, 657
- ATA_DMARQ
 - atahost.h, 657
- ATA_DR
 - atadevice.h, 642
- ATA_DRBE
 - atahost.h, 657
- ATA_DTBF
 - atahost.h, 657
- ATA_DTRO
 - atahost.h, 657
- ATA_DTRI
 - atahost.h, 657
- ata_enabled
 - atahost.c, 646
- ata_enddevice
 - atahost.c, 646
- ATA_ERR
 - atadevice.h, 642
- ATA_ERR_ABT
 - atadevice.h, 642
- ATA_ERR_AMN
 - atadevice.h, 642
- ATA_ERR_BBK
 - atadevice.h, 642
- ATA_ERR_IDNF
 - atadevice.h, 642
- ATA_ERR_TON
 - atadevice.h, 642
- ATA_ERR_UNC
 - atadevice.h, 642
- ata_execute_device_diagnostics_cmd
 - atadevice-cmdi.c, 626
 - atadevice-cmdi.h, 634
- ata_file
 - atahost.c, 646
- ata_firmware
 - atahost.c, 646
- ATA_FR
 - atadevice.h, 642
- ATA_FTE0

- atahost.h, 657
- ATA_FTE1
 - atahost.h, 657
- ata_heads
 - atahost.c, 647
- ata_host, 25
 - baseaddr, 26
 - ctrl, 26
 - dev_id, 26
 - dev_sel, 26
 - devices, 26
 - dma_mode0_td, 26
 - dma_mode0_teoc, 26
 - dma_mode0_tm, 26
 - dtr0, 26
 - dtr1, 26
 - enabled, 26
 - irq, 26
 - mem, 26
 - pctr, 26
 - pftr0, 26
 - pftr1, 26
 - pio_mode0_t1, 26
 - pio_mode0_t2, 26
 - pio_mode0_t4, 26
 - pio_mode0_teoc, 26
 - regs, 26
 - rev, 26
 - rxb, 26
 - stat, 26
 - txb, 26
- ATA_IDE_EN
 - atahost.h, 657
- ATA_IDEIS
 - atahost.h, 657
- ata_identify_device_cmd
 - atadevice-cmdi.c, 627
- ata_initialize_device_parameters_cmd
 - atadevice-cmdi.c, 627
- ata_int
 - atahost.c, 647
 - atahost.h, 657
- ATA_IORDY
 - atahost.h, 657
- ATA_IORDY_FTE0
 - atahost.h, 657
- ATA_IORDY_FTE1
 - atahost.h, 657
- ata_irq
 - atahost.c, 647
- ata_mwdma
 - atahost.c, 647
- ata_packet
 - atahost.c, 647
- ATA_PCTR
 - atahost.h, 657
- ATA_PFTR0
 - atahost.h, 657
- ATA_PFTR1
 - atahost.h, 657
- ata_pio
 - atahost.c, 647
- ata_pio_delay
 - atahost.h, 657
- ata_pio_mode0_t1
 - atahost.c, 647
- ata_pio_mode0_t2
 - atahost.c, 648
- ata_pio_mode0_t4
 - atahost.c, 648
- ata_pio_mode0_teoc
 - atahost.c, 648
- ATA_PIO_TIP
 - atahost.h, 657
- ATA_PWPP
 - atahost.h, 657
- ATA_PWPPF
 - atahost.h, 657
- ata_read32
 - atahost.c, 648
- ata_read_native_max_addr
 - atadevice-cmdi.c, 627
- ata_read_sect
 - atadevice-cmdi.c, 627
- ata_read_sectors_cmd
 - atadevice-cmdi.c, 627
- ata_reset
 - atahost.c, 648
- ata_rev
 - atahost.c, 648
- ATA_REVNO
 - atahost.h, 657
- ATA_RST
 - atahost.h, 657
- ATA_RXB
 - atahost.h, 657
- ATA_SCR
 - atadevice.h, 642
- ata_sec_end
 - atahost.c, 648
- ata_sec_start
 - atahost.c, 649
- ata_sectors
 - atahost.c, 649
- ata_set_device_signature
 - atadevice-cmdi.c, 628
- ata_set_features
 - atadevice-cmdi.c, 628

- ata_set_sect
 - atadevice-cmdi.c, 628
- ata_size
 - atahost.c, 650
- ATA_SNR
 - atadevice.h, 642
- ATA_SR
 - atadevice.h, 642
- ATA_SR_BSY
 - atadevice.h, 642
- ATA_SR_COR
 - atadevice.h, 642
- ATA_SR_DF
 - atadevice.h, 642
- ATA_SR_DRDY
 - atadevice.h, 642
- ATA_SR_DRQ
 - atadevice.h, 642
- ATA_SR_DSC
 - atadevice.h, 642
- ATA_SR_ERR
 - atadevice.h, 642
- ATA_SR_IDX
 - atadevice.h, 642
- ata_start_device
 - atahost.c, 650
- ATA_STAT
 - atahost.h, 657
- ATA_STATE_HW_RST
 - atadevice.h, 642
- ATA_STATE_IDLE
 - atadevice.h, 642
- ATA_STATE_SW_RST
 - atadevice.h, 642
- ata_status
 - atahost.c, 650
- ATA_T1
 - atahost.h, 657
- ATA_T2
 - atahost.h, 657
- ATA_T4
 - atahost.h, 657
- ATA_TD
 - atahost.h, 657
- ATA_TEOC
 - atahost.h, 657
- ATA_TM
 - atahost.h, 657
- ATA_TXB
 - atahost.h, 657
- ata_type
 - atahost.c, 650
- ata_write32
 - atahost.c, 650
- ata_write_sect
 - atadevice-cmdi.c, 628
- ata_write_sectors
 - atadevice-cmdi.c, 628
- atacmd.h
 - CFA_DISABLE_8BIT_PIO_TRANSFER_ -
MODE, 624
 - CFA_DISABLE_POWER_MODE1, 624
 - CFA_ENABLE_8BIT_PIO_TRANSFER_ -
MODE, 624
 - CFA_ENABLE_POWER_MODE1, 624
 - CFA_ERASE_SECTORS, 624
 - CFA_REQUEST_EXTENDED_ERROR_ -
CODE, 624
 - CFA_TRANSLATE_SECTOR, 624
 - CFA_WRITE_MULTIPLE_WITHOUT_ -
ERASE, 624
 - CFA_WRITE_SECTORS_WITHOUT_ -
ERASE, 624
 - CHECK_POWER_MODE, 624
 - DEVICE_RESET, 624
 - DISABLE_ADVANCED_POWER_ -
MANAGEMENT, 624
 - DISABLE_MEDIA_STATUS_ -
NOTIFICATION, 624
 - DISABLE_POWERUP_IN_STANDBY_ -
FEATURE_SET, 624
 - DISABLE_READ_LOOKAHEAD, 624
 - DISABLE_RELEASE_INTERRUPT, 624
 - DISABLE_REVERTING_TO_POWERON_ -
DEFAULTS, 624
 - DISABLE_SERVICE_INTERRUPT, 624
 - DISABLE_WRITE_CACHE, 624
 - DOWNLOAD_MICROCODE, 624
 - ENABLE_ADVANCED_POWER_ -
MANAGEMENT, 624
 - ENABLE_MEDIA_STATUS_ -
NOTIFICATION, 624
 - ENABLE_POWERUP_IN_STANDBY_ -
FEATURE_SET, 624
 - ENABLE_READ_LOOKAHEAD_ -
FEATURE, 624
 - ENABLE_RELEASE_INTERRUPT, 624
 - ENABLE_REVERTING_TO_POWERON_ -
DEFAULTS, 624
 - ENABLE_SERVICE_INTERRUPT, 624
 - ENABLE_WRITE_CACHE, 624
 - EXECUTE_DEVICE_DIAGNOSTICS, 624
 - FLUSH_CACHE, 624
 - GET_MEDIA_STATUS, 624
 - IDENTIFY_DEVICE, 624
 - IDENTIFY_PACKET_DEVICE, 624
 - IDLE, 624
 - IDLE_IMMEDIATE, 624

- INITIALIZE_DEVICE_PARAMETERS, 624
- MEDIA_EJECT, 624
- MEDIA_LOCK, 624
- MEDIA_UNLOCK, 624
- NOP, 624
- PACKET, 624
- POWERUP_IN_STANDBY_FEATURE_-
SET_SPINUP, 624
- READ_BUFFER, 624
- READ_DMA, 624
- READ_DMA_QUEUED, 624
- READ_MULTIPLE, 624
- READ_NATIVE_MAX_ADDRESS, 624
- READ_SECTOR, 624
- READ_SECTORS, 624
- READ_VERIFY_SECTOR, 624
- READ_VERIFY_SECTORS, 624
- SECURITY_DISABLE_PASSWORD, 624
- SECURITY_ERASE_PREPARE, 624
- SECURITY_ERASE_UNIT, 624
- SECURITY_FREEZE_LOCK, 624
- SECURITY_SET_PASSWORD, 624
- SECURITY_UNLOCK, 624
- SEEK, 624
- SERVICE, 624
- SET_FEATURES, 624
- SET_MAX, 624
- SET_MAX_ADDRESS, 624
- SET_MAX_FREEZE_LOCK, 624
- SET_MAX_LOCK, 624
- SET_MAX_SET_PASSWORD, 624
- SET_MAX_UNLOCK, 624
- SET_MULTIPLE_MODE, 624
- SET_TRANSFER_MODE_SECTOR_-
COUNT_REG, 624
- SLEEP, 624
- SMART, 624
- SMART_ATTRIBUTE_AUTOSAVE, 624
- SMART_DISABLE_OPERATIONS, 624
- SMART_ENABLE_OPERATIONS, 624
- SMART_EXECUTE_OFFLINE_-
IMMEDIATE, 624
- SMART_READ_DATA, 624
- SMART_READ_LOG, 624
- SMART_RETURN_STATUS, 624
- SMART_SAVE_ATTRIBUTE_VALUES, 624
- SMART_WRITE_LOG, 624
- STANDBY, 624
- STANDBY_IMMEDIATE, 624
- WRITE_BUFFER, 624
- WRITE_DMA, 624
- WRITE_DMA_QUEUED, 624
- WRITE_MULTIPLE, 624
- WRITE_SECTOR, 624
- WRITE_SECTORS, 624
- atadevice-cmdi.c
 - ata_calc_lba, 626
 - ata_cmd_complete, 626
 - ata_device_execute_cmd, 626
 - ata_device_reset_cmd, 626
 - ata_execute_device_diagnostics_cmd, 626
 - ata_identify_device_cmd, 627
 - ata_initialize_device_parameters_cmd, 627
 - ata_read_native_max_addr, 627
 - ata_read_sect, 627
 - ata_read_sectors_cmd, 627
 - ata_set_device_signature, 628
 - ata_set_features, 628
 - ata_set_sect, 628
 - ata_write_sect, 628
 - ata_write_sectors, 628
- atadevice-cmdi.h
 - ata_device_execute_cmd, 633
 - ata_execute_device_diagnostics_cmd, 634
 - BYTES_PER_SECTOR, 633
 - MIN_MWDMA_CYCLE_TIME, 633
 - MIN_PIO_CYCLE_TIME_IORDY, 633
 - MIN_PIO_CYCLE_TIME_NO_IORDY, 633
 - QUEUE_DEPTH, 633
 - RECOMMENDED_MWDMA_CYCLE_-
TIME, 633
 - SET_FEATURES_REQUIRED_AFTER_-
POWER_UP, 633
 - SUPPORT_APM, 633
 - SUPPORT_CFA, 633
 - SUPPORT_DEVICE_RESET_CMD, 633
 - SUPPORT_DOWNLOAD_MICROCODE,
633
 - SUPPORT_HOST_PROTECTED_AREA,
633
 - SUPPORT_LOOKAHEAD, 633
 - SUPPORT_NOP_CMD, 633
 - SUPPORT_POWER_MANAGEMENT, 633
 - SUPPORT_POWER_UP_IN_STANDBY_-
MODE, 633
 - SUPPORT_READ_BUFFER_CMD, 633
 - SUPPORT_READ_WRITE_DMA_-
QUEUED, 633
 - SUPPORT_RELEASE_INTERRUPT, 633
 - SUPPORT_REMOVABLE_MEDIA, 633
 - SUPPORT_REMOVABLE_MEDIA_-
NOTIFICATION, 633
 - SUPPORT_SECURITY_MODE, 633
 - SUPPORT_SERVICE_INTERRUPT, 633
 - SUPPORT_SET_MAX, 633
 - SUPPORT_SMART, 633
 - SUPPORT_WRITE_BUFFER_CMD, 633
 - SUPPORT_WRITE_CACHE, 633

- atadevice.c
 - ata_device_do_command_register, 636
 - ata_device_do_control_register, 636
 - ata_device_hw_reset, 636
 - ata_device_init, 636
 - ata_device_write, 636
 - ata_devices_hw_reset, 637
 - ata_devices_init, 637
 - ata_devices_read, 637
 - ata_devices_write, 637
 - open_file, 637
 - open_local, 638
- atadevice.h
 - ATA_ASR, 642
 - ATA_CHR, 642
 - ATA_CLR, 642
 - ATA_CR, 642
 - ATA_DA, 642
 - ATA_DAR_DS0, 642
 - ATA_DAR_DS1, 642
 - ATA_DAR_H, 642
 - ATA_DAR_WTG, 642
 - ATA_DCR, 642
 - ATA_DCR_IEN, 642
 - ATA_DCR_RST, 642
 - ata_devices_hw_reset, 642
 - ata_devices_init, 643
 - ata_devices_read, 643
 - ata_devices_write, 643
 - ATA_DHR, 642
 - ATA_DHR_DEV, 642
 - ATA_DHR_H, 642
 - ATA_DHR_LBA, 642
 - ATA_DR, 642
 - ATA_ERR, 642
 - ATA_ERR_ABT, 642
 - ATA_ERR_AMN, 642
 - ATA_ERR_BBK, 642
 - ATA_ERR_IDNF, 642
 - ATA_ERR_TON, 642
 - ATA_ERR_UNC, 642
 - ATA_FR, 642
 - ATA_SCR, 642
 - ATA_SNR, 642
 - ATA_SR, 642
 - ATA_SR_BSY, 642
 - ATA_SR_COR, 642
 - ATA_SR_DF, 642
 - ATA_SR_DRDY, 642
 - ATA_SR_DRQ, 642
 - ATA_SR_DSC, 642
 - ATA_SR_ERR, 642
 - ATA_SR_IDX, 642
 - ATA_STATE_HW_RST, 642
 - ATA_STATE_IDLE, 642
 - ATA_STATE_SW_RST, 642
 - TYPE_FILE, 642
 - TYPE_LOCAL, 642
 - TYPE_NO_CONNECT, 642
- atahost.c
 - ata_baseaddr, 646
 - ata_dev_id, 646
 - ata_dma_mode0_td, 646
 - ata_dma_mode0_teoc, 646
 - ata_dma_mode0_tm, 646
 - ata_enabled, 646
 - ata_enddevice, 646
 - ata_file, 646
 - ata_firmware, 646
 - ata_heads, 647
 - ata_int, 647
 - ata_irq, 647
 - ata_mwdma, 647
 - ata_packet, 647
 - ata_pio, 647
 - ata_pio_mode0_t1, 647
 - ata_pio_mode0_t2, 648
 - ata_pio_mode0_t4, 648
 - ata_pio_mode0_teoc, 648
 - ata_read32, 648
 - ata_reset, 648
 - ata_rev, 648
 - ata_sec_end, 648
 - ata_sec_start, 649
 - ata_sectors, 649
 - ata_size, 650
 - ata_start_device, 650
 - ata_status, 650
 - ata_type, 650
 - ata_write32, 650
 - conf_dev, 653
 - DMA_MODE0_TD, 646
 - DMA_MODE0_TEOC, 646
 - DMA_MODE0_TM, 646
 - PIO_MODE0_T1, 646
 - PIO_MODE0_T2, 646
 - PIO_MODE0_T4, 646
 - PIO_MODE0_TEOC, 646
 - reg_ata_sec, 650
- atahost.h
 - ATA_ADDR_SPACE, 657
 - ATA_BELEC0, 657
 - ATA_BELEC1, 657
 - ATA_CTRL, 657
 - ATA_DEVID, 657
 - ata_dma_delay, 657
 - ATA_DMA_EN, 657
 - ATA_DMA_RD, 657

- ATA_DMA_TIP, [657](#)
- ATA_DMA_WR, [657](#)
- ATA_DMARQ, [657](#)
- ATA_DRBE, [657](#)
- ATA_DTBF, [657](#)
- ATA_DTR0, [657](#)
- ATA_DTR1, [657](#)
- ATA_FTE0, [657](#)
- ATA_FTE1, [657](#)
- ATA_IDE_EN, [657](#)
- ATA_IDEIS, [657](#)
- ata_int, [657](#)
- ATA_IORDY, [657](#)
- ATA_IORDY_FTE0, [657](#)
- ATA_IORDY_FTE1, [657](#)
- ATA_PCTR, [657](#)
- ATA_PFTR0, [657](#)
- ATA_PFTR1, [657](#)
- ata_pio_delay, [657](#)
- ATA_PIO_TIP, [657](#)
- ATA_PWPP, [657](#)
- ATA_PWPPF, [657](#)
- ATA_REVNO, [657](#)
- ATA_RST, [657](#)
- ATA_RXB, [657](#)
- ATA_STAT, [657](#)
- ATA_T1, [657](#)
- ATA_T2, [657](#)
- ATA_T4, [657](#)
- ATA_TD, [657](#)
- ATA_TEOC, [657](#)
- ATA_TM, [657](#)
- ATA_TXB, [657](#)
- is_ata_hostadr, [657](#)
- reg_ata_sec, [658](#)
- audio_cnt
 - sprs.c, [322](#)
- automata
 - or32.c, [435](#)
- aux
 - gpio_device, [124](#)
- auxiliary_inputs
 - gpio_device, [124](#)
- b
 - cuc_timings, [69](#)
- ba_mask
 - mc, [147](#)
- backward
 - branchstat, [32](#)
- base_id
 - vapi_handler, [178](#)
- base_include
 - sim-config.c, [802](#)
- base_vapi_id
 - eth_device, [113](#)
 - gpio_device, [124](#)
- baseaddr
 - ata_host, [26](#)
 - dev_16450, [74](#)
 - dev_generic, [77](#)
 - dma_controller, [84](#)
 - eth_device, [113](#)
 - fb_state, [118](#)
 - gpio_device, [124](#)
 - kbd_state, [144](#)
 - mc, [147](#)
 - mem_config, [150](#)
 - vga_state, [180](#)
- basename
 - xterm.c, [676](#)
- baud_table
 - tty.c, [673](#)
- bb
 - _cuc_func, [16](#)
- bb.c
 - build_bb, [443](#)
 - count_bb_seq, [443](#)
 - cpy_bb, [443](#)
 - cuc_check, [443](#)
 - detect_bb, [443](#)
 - dup_func, [444](#)
 - expand_bb, [444](#)
 - free_func, [444](#)
 - generate_bb_seq, [444](#)
 - join_bb, [444](#)
 - optimize_bb, [445](#)
 - preunroll_loop, [445](#)
 - print_bb_num, [445](#)
 - print_cuc_bb, [446](#)
 - recalc_last_used_reg, [446](#)
 - reg_dep, [446](#)
 - reg_dep_rec, [446](#)
 - relocate_bb, [446](#)
 - remove_dead_bb, [446](#)
 - roll_loop, [446](#)
 - simplify_bb, [446](#)
- BB_DEAD
 - cuc.h, [469](#)
- BB_INLOOP
 - cuc.h, [469](#)
- BB_OPTIONAL
 - cuc.h, [469](#)
- bb_size
 - timings.c, [508](#)
- BBID_END
 - cuc.h, [469](#)
- BBID_START

- cuc.h, 469
- bd
 - eth_device, 113
- bd_addr
 - eth_device, 113
- bd_index
 - eth_device, 113
- bd_ram
 - eth_device, 113
- bf
 - mstats_entry, 156
- bff, 28
 - close_obj, 28
 - get_func_len, 28
 - get_func_name, 28
 - get_func_reloc, 28
 - get_func_start, 28
 - open_obj, 28
- bfbs
 - dyngen-elf.c, 359
 - dyngen.h, 365
- bit_mask
 - abstract.c, 213
- bits
 - INFOHEADER, 131
 - spr_def, 173
- block_jtag
 - gdbcomm.c, 532
 - gdbcomm.h, 535
- block_mask
 - ic, 127
- block_offset_mask
 - ic, 127
- blocksize
 - config, 55
 - ic, 127
- blocksize_log2
 - ic, 127
- BMP_HEADER, 29
 - offset, 29
 - reserved1, 29
 - reserved2, 29
 - size, 29
 - type, 29
- bnf
 - mstats_entry, 156
- boolean
 - gdbcomm.h, 535
- BP_HARDWARE
 - rsp-server.c, 540
- BP_MEMORY
 - rsp-server.c, 540
- bpb
 - branch-predict.c, 187
 - config, 55
 - mstats_entry, 156
 - bpb/branch-predict.c, 183
 - bpb/branch-predict.h, 188
 - bpb_btic
 - branch-predict.c, 186
 - bpb_enabled
 - branch-predict.c, 186
 - bpb_entry, 30
 - addr, 30
 - lru, 30
 - taken, 30
 - way, 30
 - bpb_hitdelay
 - branch-predict.c, 186
 - bpb_info
 - branch-predict.c, 186
 - branch-predict.h, 189
 - BPB_LEN
 - branch-predict.c, 186
 - bpb_missdelay
 - branch-predict.c, 186
 - BPB_PSTATES
 - branch-predict.c, 186
 - bpb_sbp_bf_fwd
 - branch-predict.c, 186
 - bpb_sbp_bnf_fwd
 - branch-predict.c, 186
 - bpb_update
 - branch-predict.c, 186
 - branch-predict.h, 189
 - BPB_USTATES
 - branch-predict.c, 186
 - BPB_WAYS
 - branch-predict.c, 186
 - bpbstat, 31
 - correct, 31
 - hit, 31
 - incorrect, 31
 - miss, 31
 - branch-predict.c
 - bpb, 187
 - bpb_btic, 186
 - bpb_enabled, 186
 - bpb_hitdelay, 186
 - bpb_info, 186
 - BPB_LEN, 186
 - bpb_missdelay, 186
 - BPB_PSTATES, 186
 - bpb_sbp_bf_fwd, 186
 - bpb_sbp_bnf_fwd, 186
 - bpb_update, 186
 - BPB_USTATES, 186
 - BPB_WAYS, 186

- btic, 187
- BTIC_BLOCKSIZE, 186
- btic_info, 186
- BTIC_LEN, 186
- btic_update, 186
- BTIC_USTATES, 186
- BTIC_WAYS, 186
- reg_bpb_sec, 186
- branch-predict.h
 - bpb_info, 189
 - bpb_update, 189
 - btic_info, 189
 - btic_update, 189
 - reg_bpb_sec, 189
- branch_index
 - verilog.c, 511
- branchstat, 32
 - backward, 32
 - forward, 32
 - nottaken, 32
 - taken, 32
- breakpoint
 - execute.c, 375
- breakpoint_entry, 33
 - addr, 33
 - next, 33
- breakpoints
 - labels.c, 269
 - labels.h, 271
- bsize
 - COFF_AOUTHDR, 40
- btic
 - branch-predict.c, 187
 - config, 55
 - mstats_entry, 156
- BTIC_BLOCKSIZE
 - branch-predict.c, 186
- btic_entry, 34
 - addr, 34
 - insn, 34
 - lru, 34
 - way, 34
- btic_info
 - branch-predict.c, 186
 - branch-predict.h, 189
- BTIC_LEN
 - branch-predict.c, 186
- btic_update
 - branch-predict.c, 186
 - branch-predict.h, 189
- BTIC_USTATES
 - branch-predict.c, 186
- BTIC_WAYS
 - branch-predict.c, 186
- bticstat, 35
 - hit, 35
 - miss, 35
- buf
 - kbd_state, 144
- buf_count
 - kbd_state, 144
- buf_head
 - kbd_state, 144
- BUF_SIZE
 - mprofiler.c, 588
- buf_tail
 - kbd_state, 144
- build_automata
 - or32.c, 432
- build_bb
 - bb.c, 443
 - cuc.h, 470
- build_insn
 - load.c, 495
- byte_enabled
 - dev_generic, 77
- byteadd
 - mstats_entry, 156
- bytes_left
 - eth_device, 113
- BYTES_PER_SECTOR
 - atadevice-cmdi.h, 633
- bytes_read
 - eth_device, 113
- bytes_sent
 - eth_device, 113
- C_ALIAS
 - coff.h, 249
- C_ARG
 - coff.h, 249
- C_AUTO
 - coff.h, 249
- C_AUTOARG
 - coff.h, 249
- C_BCOMM
 - coff.h, 249
- C_BINCL
 - coff.h, 249
- C_BLOCK
 - coff.h, 249
- C_BSTAT
 - coff.h, 249
- C_DECL
 - coff.h, 249
- C_DEFINE
 - coff.h, 249
- C_ECOML

- [coff.h](#), 249
- C_ECOMM
 - [coff.h](#), 249
- C_EFCN
 - [coff.h](#), 249
- C_EINCL
 - [coff.h](#), 249
- C_ENTAG
 - [coff.h](#), 249
- C_ENTRY
 - [coff.h](#), 249
- C_EOS
 - [coff.h](#), 249
- C_ESTAT
 - [coff.h](#), 249
- C_EXT
 - [coff.h](#), 249
- C_EXTDEF
 - [coff.h](#), 249
- C_EXTLAB
 - [coff.h](#), 249
- C_FCNC
 - [coff.h](#), 249
- C_FIELD
 - [coff.h](#), 249
- C_FILE
 - [coff.h](#), 249
- c_file_head
 - [dyngen.c](#), 363
- C_FUN
 - [coff.h](#), 249
- C_GSYM
 - [coff.h](#), 249
- C_HIDDEN
 - [coff.h](#), 249
- C_HIDEXT
 - [coff.h](#), 249
- C_LABEL
 - [coff.h](#), 249
- C_LASTENT
 - [coff.h](#), 249
- C_LEAFEXT
 - [coff.h](#), 249
- C_LEAFPROC
 - [coff.h](#), 249
- C_LEAFSTAT
 - [coff.h](#), 249
- C_LINE
 - [coff.h](#), 249
- C_LSYM
 - [coff.h](#), 249
- C_MOE
 - [coff.h](#), 249
- C_MOS
 - [coff.h](#), 249
- C_MOU
 - [coff.h](#), 249
- C_NT_WEAK
 - [coff.h](#), 249
- C_NULL
 - [coff.h](#), 249
- C_OPTVAR
 - [coff.h](#), 249
- C_PRAGMA
 - [coff.h](#), 249
- C_PSYM
 - [coff.h](#), 249
- C_REG
 - [coff.h](#), 249
- C_REGPARAM
 - [coff.h](#), 249
- c_rel_file_head
 - [dyngen.c](#), 363
- c_rel_file_tail
 - [dyngen.c](#), 363
- C_RPSYM
 - [coff.h](#), 249
- C_RSYM
 - [coff.h](#), 249
- C_SCALL
 - [coff.h](#), 249
- C_SECTION
 - [coff.h](#), 249
- C_SEGMENT
 - [coff.h](#), 249
- C_SHADOW
 - [coff.h](#), 249
- C_STAT
 - [coff.h](#), 249
- C_STATLAB
 - [coff.h](#), 249
- C_STRTAG
 - [coff.h](#), 249
- C_STSYM
 - [coff.h](#), 249
- c_sw_file_head
 - [dyngen.c](#), 363
- c_sw_file_tail
 - [dyngen.c](#), 364
- C_SYSTEM
 - [coff.h](#), 249
- C_TCSYM
 - [coff.h](#), 249
- C_THUMBEXT
 - [coff.h](#), 249
- C_THUMBEXTFUNC
 - [coff.h](#), 249
- C_THUMBLABEL

- coff.h, 249
- C_THUMBSTAT
 - coff.h, 249
- C_THUMBSTATFUNC
 - coff.h, 249
- C_TPDEF
 - coff.h, 249
- C_UEXT
 - coff.h, 249
- C_ULABEL
 - coff.h, 249
- C_UNTAG
 - coff.h, 249
- C_USTATIC
 - coff.h, 249
- C_VERSION
 - coff.h, 249
- C_WEAKEXT
 - coff.h, 249
- cache/dcache-model.c, 190
- cache/dcache-model.h, 195
- cache/icache-model.c, 198
- cache/icache-model.h, 203
- cachestats_entry, 36
 - readhit, 36
 - readmiss, 36
 - writehit, 36
 - writemiss, 36
- calc_cycles
 - cuc.c, 452
- calc_max
 - adv.c, 441
- calc_size
 - cuc.c, 452
- calculate_watchpoints
 - debug-unit.c, 516
- caller_saved
 - cuc.c, 462
 - cuc.h, 481
- calling_convention
 - config, 55
- calls
 - func_struct, 122
- cam_addr
 - fb_state, 118
- CAM_SIZEX
 - fb.c, 714
- CAM_SIZEY
 - fb.c, 714
- camera_pos
 - fb_state, 118
- camerax
 - fb_state, 118
- cameray
 - fb_state, 118
- ccmd
 - kbd_state, 144
- ccmdbyte
 - kbd_state, 144
- ce
 - mem_config, 150
- CFA_DISABLE_8BIT_PIO_TRANSFER_MODE
 - atacmd.h, 624
- CFA_DISABLE_POWER_MODE1
 - atacmd.h, 624
- CFA_ENABLE_8BIT_PIO_TRANSFER_MODE
 - atacmd.h, 624
- CFA_ENABLE_POWER_MODE1
 - atacmd.h, 624
- CFA_ERASE_SECTORS
 - atacmd.h, 624
- CFA_REQUEST_EXTENDED_ERROR_CODE
 - atacmd.h, 624
- CFA_TRANSLATE_SECTOR
 - atacmd.h, 624
- CFA_WRITE_MULTIPLE_WITHOUT_ERASE
 - atacmd.h, 624
- CFA_WRITE_SECTORS_WITHOUT_ERASE
 - atacmd.h, 624
- ch
 - dma_controller, 84
- chain
 - jtr_chain_message, 133
- change_buf_addr
 - fb.c, 715
- change_insn_type
 - insn.c, 484
 - insn.h, 492
- channel, 37
 - data, 37
 - dev_16450, 74
 - ops, 37
- channel.c
 - channel_close, 661
 - channel_init, 661
 - channel_open, 661
 - channel_read, 661
 - channel_write, 661
 - find_channel_factory, 661
 - head, 661
 - preloaded, 661
- channel.h
 - channel_close, 662
 - channel_init, 662
 - channel_open, 662
 - channel_read, 662
 - channel_write, 662
- channel_close

- channel.c, 661
- channel.h, 662
- channel_factory, 38
 - name, 38
 - next, 38
 - ops, 38
- channel_init
 - channel.c, 661
 - channel.h, 662
- channel_mask
 - dma_channel, 81
- CHANNEL_ND_I
 - dma.c, 689
- channel_number
 - dma_channel, 81
- channel_open
 - channel.c, 661
 - channel.h, 662
- channel_ops, 39
 - close, 39
 - free, 39
 - init, 39
 - isok, 39
 - open, 39
 - read, 39
 - status, 39
 - write, 39
- channel_read
 - channel.c, 661
 - channel.h, 662
- channel_str
 - dev_16450, 74
- channel_write
 - channel.c, 661
 - channel.h, 662
- channels/generic.c
 - generic_close, 727
 - generic_free, 727
 - generic_open, 727
- channels/generic.h
 - generic_close, 729
 - generic_free, 729
 - generic_open, 729
- char_clks
 - 16450.c, 604
 - dev_16450, 74
- check_debug_unit
 - debug-unit.c, 516
 - debug-unit.h, 525
- check_depend
 - execute.c, 368
- check_dma_ack_o
 - dma.c, 689
 - dma.h, 695
- check_insn_exec
 - sim-cmd.c, 785
- check_int
 - toplevel-support.c, 852
 - toplevel-support.h, 857
- CHECK_INT_TIME
 - sim-config.h, 810
- check_memory_conflict
 - cuc/memory.c, 500
- CHECK_POWER_MODE
 - atacmd.h, 624
- chunk_size
 - dma_channel, 81
- class_ptr
 - config, 55
- clean_deps
 - cuc.h, 471
 - cuc/memory.c, 500
- clear_dma_nd_i
 - dma.c, 689
 - dma.h, 695
- clear_dma_req_i
 - dma.c, 689
 - dma.h, 695
- CLEAR_FLAG
 - fields.h, 720
- clear_interrupt
 - pic.c, 766
 - pic.h, 769
- client_fd
 - rsp-server.c, 559
- client_waiting
 - rsp-server.c, 559
- clkcycle_ps
 - config, 55
- close
 - channel_ops, 39
- close_obj
 - bff, 28
- cmatch
 - _csm_list, 14
 - cuc_shared_item, 67
- cmd_handle
 - sim_command, 169
- cmov_needed
 - insn.c, 484
- cmovs
 - _csm_list, 14
- cnt
 - _csm_list, 14
 - cuc_bb, 63
 - memory_hash, 153
- cnt_dynamic
 - dstats_entry, 88

- fstats_entry, 121
- sstats_entry, 174
- CNV16
 - fb.c, 714
- CNV32
 - fb.c, 714
- code
 - ps2kbd.c, 756
- coff.h
 - C_ALIAS, 249
 - C_ARG, 249
 - C_AUTO, 249
 - C_AUTOARG, 249
 - C_BCOMM, 249
 - C_BINCL, 249
 - C_BLOCK, 249
 - C_BSTAT, 249
 - C_DECL, 249
 - C_DEFINE, 249
 - C_ECOML, 249
 - C_ECOMM, 249
 - C_EFCN, 249
 - C_EINCL, 249
 - C_ENTAG, 249
 - C_ENTRY, 249
 - C_EOS, 249
 - C_ESTAT, 249
 - C_EXT, 249
 - C_EXTDEF, 249
 - C_EXTLAB, 249
 - C_FCN, 249
 - C_FIELD, 249
 - C_FILE, 249
 - C_FUN, 249
 - C_GSYM, 249
 - C_HIDDEN, 249
 - C_HIDEEXT, 249
 - C_LABEL, 249
 - C_LASTENT, 249
 - C_LEAFEXT, 249
 - C_LEAFPROC, 249
 - C_LEAFSTAT, 249
 - C_LINE, 249
 - C_LSYM, 249
 - C_MOE, 249
 - C_MOS, 249
 - C_MOU, 249
 - C_NT_WEAK, 249
 - C_NULL, 249
 - C_OPTVAR, 249
 - C_PRAGMA, 249
 - C_PSYM, 249
 - C_REG, 249
 - C_REGPARM, 249
 - C_RPSYM, 249
 - C_RSYM, 249
 - C_SCALL, 249
 - C_SECTION, 249
 - C_SEGMENT, 249
 - C_SHADOW, 249
 - C_STAT, 249
 - C_STATLAB, 249
 - C_STRTAG, 249
 - C_STSYM, 249
 - C_SYSTEM, 249
 - C_TCSYM, 249
 - C_THUMBEXT, 249
 - C_THUMBEXTFUNC, 249
 - C_THUMBLABEL, 249
 - C_THUMBSTAT, 249
 - C_THUMBSTATFUNC, 249
 - C_TPDEF, 249
 - C_UEXT, 249
 - C_ULABEL, 249
 - C_UNTAG, 249
 - C_USTATIC, 249
 - C_VERSION, 249
 - C_WEAKEXT, 249
 - COFF_AOUTSZ, 249
 - COFF_AUXENT, 249
 - COFF_AUXESZ, 249
 - COFF_BSS, 249
 - COFF_COMMENT, 249
 - COFF_DATA, 249
 - COFF_DEF_BSS_SECTION_ALIGNMENT, 249
 - COFF_DEF_DATA_SECTION_ALIGNMENT, 249
 - COFF_DEF_SECTION_ALIGNMENT, 249
 - COFF_DEF_TEXT_SECTION_ALIGNMENT, 249
 - COFF_DMAGIC, 249
 - COFF_E_DIMNUM, 249
 - COFF_E_FILNMLEN, 249
 - COFF_E_SYMNMLEN, 249
 - COFF_ETEXT, 249
 - COFF_F_AR16WR, 249
 - COFF_F_AR32W, 249
 - COFF_F_AR32WR, 249
 - COFF_F_EXEC, 249
 - COFF_F_LNNO, 249
 - COFF_F_LSYMS, 249
 - COFF_F_MINMAL, 249
 - COFF_F_NODEF, 249
 - COFF_F_PATCH, 249
 - COFF_F_RELFLG, 249
 - COFF_F_SWABD, 249
 - COFF_F_UPDATE, 249

- COFF_FILHDR, 249
- COFF_FILHSZ, 249
- COFF_I386BADMAG, 249
- COFF_I386MAGIC, 249
- COFF_JMAGIC, 249
- COFF_LIB, 249
- COFF_LINENO, 249
- COFF_LINESZ, 249
- COFF_LONG, 249
- COFF_LONG_H, 249
- COFF_LONG_L, 249
- COFF_N_BTMASK, 249
- COFF_N_BTSHFT, 249
- COFF_N_TMASK, 249
- COFF_N_TSHIFT, 249
- COFF_OMAGIC, 249
- COFF_RELOC, 249
- COFF_RELSZ, 249
- COFF_SCNHDR, 249
- COFF_SCNHSZ, 249
- COFF_SECT_BSS, 249
- COFF_SECT_DATA, 249
- COFF_SECT_REQD, 249
- COFF_SECT_TEXT, 249
- COFF_SHMAGIC, 249
- COFF_SHORT, 249
- COFF_SHORT_H, 249
- COFF_SHORT_L, 249
- COFF_SLIBHD, 249
- COFF_SLIBSZ, 249
- COFF_STMAGIC, 249
- COFF_STYP_BSS, 249
- COFF_STYP_COPY, 249
- COFF_STYP_DATA, 249
- COFF_STYP_DSECT, 249
- COFF_STYP_GROUP, 249
- COFF_STYP_INFO, 249
- COFF_STYP_LIB, 249
- COFF_STYP_NOLOAD, 249
- COFF_STYP_OVER, 249
- COFF_STYP_PAD, 249
- COFF_STYP_REG, 249
- COFF_STYP_TEXT, 249
- COFF_SYMENT, 249
- COFF_SYMESZ, 249
- COFF_TEXT, 249
- COFF_ZMAGIC, 249
- E_DIMNUM, 249
- E_FILNMLEN, 249
- E_SYMNMLEN, 249
- KEEP_ENDIAN_LONG, 249
- KEEP_ENDIAN_SHORT, 250
- SWAP_ENDIAN_LONG, 250
- SWAP_ENDIAN_SHORT, 250
- COFF_AOUTHDR, 40
 - bsize, 40
 - data_start, 40
 - dsiz, 40
 - entry, 40
 - magic, 40
 - text_start, 40
 - tsiz, 40
 - vstamp, 40
- COFF_AOUTSZ
 - coff.h, 249
- COFF_AUXENT
 - coff.h, 249
- COFF_auxent, 41
 - x_ary, 43
 - x_dimen, 43
 - x_endndx, 43
 - x_fcn, 43
 - x_fcny, 43
 - x_file, 43
 - x_fname, 43
 - x_fsize, 43
 - x_inno, 43
 - x_innoptr, 43
 - x_lnsz, 43
 - x_misc, 43
 - x_n, 43
 - x_nlinno, 43
 - x_nreloc, 43
 - x_offset, 43
 - x_scn, 43
 - x_scnlen, 43
 - x_size, 43
 - x_sym, 43
 - x_tagndx, 43
 - x_tv, 43
 - x_tvfill, 43
 - x_tvlen, 43
 - x_tvndx, 43
 - x_tvrn, 43
 - x_zeroes, 43
- COFF_AUXESZ
 - coff.h, 249
- COFF_BSS
 - coff.h, 249
- COFF_COMMENT
 - coff.h, 249
- COFF_DATA
 - coff.h, 249
- COFF_DEF_BSS_SECTION_ALIGNMENT
 - coff.h, 249
- COFF_DEF_DATA_SECTION_ALIGNMENT
 - coff.h, 249
- COFF_DEF_SECTION_ALIGNMENT

- coff.h, 249
- COFF_DEF_TEXT_SECTION_ALIGNMENT
 - coff.h, 249
- COFF_DMAGIC
 - coff.h, 249
- COFF_E_DIMNUM
 - coff.h, 249
- COFF_E_FILNMLEN
 - coff.h, 249
- COFF_E_SYMNMLEN
 - coff.h, 249
- COFF_ETEXT
 - coff.h, 249
- COFF_F_AR16WR
 - coff.h, 249
- COFF_F_AR32W
 - coff.h, 249
- COFF_F_AR32WR
 - coff.h, 249
- COFF_F_EXEC
 - coff.h, 249
- COFF_F_LNNO
 - coff.h, 249
- COFF_F_LSYMS
 - coff.h, 249
- COFF_F_MINMAL
 - coff.h, 249
- COFF_F_NODF
 - coff.h, 249
- COFF_F_PATCH
 - coff.h, 249
- COFF_F_RELFLG
 - coff.h, 249
- COFF_F_SWABD
 - coff.h, 249
- COFF_F_UPDATE
 - coff.h, 249
- COFF_filehdr, 45
 - f_flags, 45
 - f_magic, 45
 - f_nscns, 45
 - f_nsyms, 45
 - f_opthdr, 45
 - f_symptr, 45
 - f_timdat, 45
- COFF_FILHDR
 - coff.h, 249
- COFF_FILHSZ
 - coff.h, 249
- COFF_I386BADMAG
 - coff.h, 249
- COFF_I386MAGIC
 - coff.h, 249
- COFF_JMAGIC
 - coff.h, 249
- COFF_LIB
 - coff.h, 249
- COFF_LINENO
 - coff.h, 249
- COFF_lineno, 46
 - l_addr, 46
 - l_inno, 46
 - l_paddr, 46
 - l_symndx, 46
- COFF_LINESZ
 - coff.h, 249
- COFF_LONG
 - coff.h, 249
- COFF_LONG_H
 - coff.h, 249
- COFF_LONG_L
 - coff.h, 249
- COFF_N_BTMASK
 - coff.h, 249
- COFF_N_BTSHFT
 - coff.h, 249
- COFF_N_TMASK
 - coff.h, 249
- COFF_N_TSHIFT
 - coff.h, 249
- COFF_OMAGIC
 - coff.h, 249
- COFF_RELOC
 - coff.h, 249
- COFF_reloc, 47
 - r_symndx, 47
 - r_type, 47
 - r_vaddr, 47
- COFF_RELSZ
 - coff.h, 249
- COFF_SCNHDR
 - coff.h, 249
- COFF_scnhdr, 48
 - s_flags, 48
 - s_innoptr, 48
 - s_name, 48
 - s_nlnno, 48
 - s_nreloc, 48
 - s_paddr, 48
 - s_relptr, 48
 - s_scnptr, 48
 - s_size, 48
 - s_vaddr, 48
- COFF_SCNHSZ
 - coff.h, 249
- COFF_SECT_BSS
 - coff.h, 249
- COFF_SECT_DATA
 - coff.h, 249

- coff.h, 249
- COFF_SECT_REQD
 - coff.h, 249
- COFF_SECT_TEXT
 - coff.h, 249
- COFF_SHMAGIC
 - coff.h, 249
- COFF_SHORT
 - coff.h, 249
- COFF_SHORT_H
 - coff.h, 249
- COFF_SHORT_L
 - coff.h, 249
- COFF_slib, 49
 - sl_entsz, 49
 - sl_pathndx, 49
- COFF_SLIBHD
 - coff.h, 249
- COFF_SLIBSZ
 - coff.h, 249
- COFF_STMAGIC
 - coff.h, 249
- COFF_STYP_BSS
 - coff.h, 249
- COFF_STYP_COPY
 - coff.h, 249
- COFF_STYP_DATA
 - coff.h, 249
- COFF_STYP_DSECT
 - coff.h, 249
- COFF_STYP_GROUP
 - coff.h, 249
- COFF_STYP_INFO
 - coff.h, 249
- COFF_STYP_LIB
 - coff.h, 249
- COFF_STYP_NOLOAD
 - coff.h, 249
- COFF_STYP_OVER
 - coff.h, 249
- COFF_STYP_PAD
 - coff.h, 249
- COFF_STYP_REG
 - coff.h, 249
- COFF_STYP_TEXT
 - coff.h, 249
- COFF_SYMENT
 - coff.h, 249
- COFF_syment, 50
 - e, 50
 - e_name, 50
 - e_numaux, 50
 - e_offset, 50
 - e_sclass, 50
 - e_scnum, 50
 - e_type, 50
 - e_value, 50
 - e_zeroes, 50
- COFF_SYMESZ
 - coff.h, 249
- COFF_TEXT
 - coff.h, 249
- COFF_ZMAGIC
 - coff.h, 249
- collconf
 - eth_device, 113
- command
 - ata_device, 22
 - jtr_chain_message, 133
 - jtr_read_block_message, 136
 - jtr_read_message, 138
 - jtr_write_block_message, 140
 - jtr_write_message, 142
- common-i386.h
 - get_pc, 325
 - high32, 325
 - low32, 325
 - set_pc, 325
 - upd_sim_cycles, 325
 - useless_x86, 325
 - val3232, 325
 - val64, 325
- COMP
 - op.c, 423
- COMP_CAST
 - op.c, 423
- COMP_NAME
 - op.c, 423
- compression
 - INFOHEADER, 131
- comutative
 - cuc_known_insn, 66
- conf
 - ata_device, 22
- conf_dev
 - atahost.c, 653
- config, 51
 - blocksize, 55
 - bpb, 55
 - btic, 55
 - calling_convention, 55
 - class_ptr, 55
 - clkcycle_ps, 55
 - cpu, 55
 - cuc, 55
 - dc, 55
 - debug, 55
 - dependstats, 55

- enable_bursts, 55
- enabled, 55
- exe_log, 55
- exe_log_end, 55
- exe_log_fn, 55
- exe_log_marker, 55
- exe_log_start, 55
- exe_log_type, 55
- ext, 55
- gdb_enabled, 55
- hazards, 55
- hide_device_id, 55
- history, 55
- hitdelay, 55
- load_hitdelay, 55
- load_missdelay, 55
- log_enabled, 55
- memory_order, 55
- missdelay, 55
- mprof_fn, 55
- mprofile, 55
- no_multicycle, 55
- nsets, 55
- nways, 55
- pic, 55
- pm, 55
- prof_fn, 55
- profile, 55
- read_up, 55
- rsp_enabled, 55
- rsp_port, 55
- sbp_bf_fwd, 55
- sbp_bnf_fwd, 55
- sbuf_len, 55
- server_port, 55
- sim, 55
- sim-config.c, 808
- sim-config.h, 814
- store_hitdelay, 55
- store_missdelay, 55
- superscalar, 55
- timings_fn, 55
- ustates, 55
- vapi, 55
- vapi_fn, 55
- vapi_id, 55
- verbose, 55
- write_up, 55
- config::pic, 57
 - edge_trigger, 57
 - enabled, 57
- config_param, 58
 - func, 58
 - name, 58
 - next, 58
 - type, 58
- config_section, 59
 - dat, 59
 - name, 59
 - next, 59
 - params, 59
 - sec_end, 59
 - sec_start, 59
- connected
 - tcp_channel, 176
- controller
 - dma_channel, 81
- controlmoder
 - eth_device, 113
- conv
 - load.c, 498
- correct
 - bpbstat, 31
- count_bb_seq
 - bb.c, 443
- count_cmovs
 - insn.c, 484
- cover_insn
 - or32.c, 432
- cpu
 - config, 55
 - runtime, 165
- cpu-config.c, 206
 - cpu_cfg, 207
 - cpu_cfg, 207
 - cpu_dependstats, 207
 - cpu_hazards, 207
 - cpu_rev, 207
 - cpu_sbuf_len, 207
 - cpu_sr, 207
 - cpu_superscalar, 208
 - cpu_upr, 208
 - cpu_ver, 208
 - reg_cpu_sec, 208
 - WARNING, 207
- cpu-config.h, 210
 - reg_cpu_sec, 210
- cpu/common/abstract.c, 211
- cpu/common/abstract.h, 229
- cpu/common/coff.h, 244
- cpu/common/elf.h, 251
- cpu/common/execute.h, 261
- cpu/common/labels.c, 268
- cpu/common/labels.h, 270
- cpu/common/parse.c, 272
- cpu/common/parse.h, 278
- cpu/common/stats.c, 280
- cpu/common/stats.h, 284

- cpu/common/trace.c, 286
- cpu/common/trace.h, 288
- cpu/or1k/arch.h, 290
- cpu/or1k/except.c, 292
- cpu/or1k/except.h, 294
- cpu/or1k/spr-defs.h, 298
- cpu/or1k/spr-dump.c, 308
- cpu/or1k/spr-dump.h, 320
- cpu/or1k/sprs.c, 321
- cpu/or1k/sprs.h, 323
- cpu/or32/common-i386.h, 325
- cpu/or32/def-op-t.h, 326
- cpu/or32/dyn-rec.c, 328
- cpu/or32/dyn-rec.h, 352
- cpu/or32/dyn32-defs.h, 357
- cpu/or32/dyngen-elf.c, 358
- cpu/or32/dyngen-i386.c, 360
- cpu/or32/dyngen.c, 362
- cpu/or32/dyngen.h, 365
- cpu/or32/execute.c, 366
- cpu/or32/generate.c, 377
- cpu/or32/i386-regs.h, 380
- cpu/or32/insnset.c, 381
- cpu/or32/op-1t-op.h, 393
- cpu/or32/op-1t.h, 394
- cpu/or32/op-2t-op.h, 395
- cpu/or32/op-2t.h, 396
- cpu/or32/op-3t-op.h, 397
- cpu/or32/op-3t.h, 398
- cpu/or32/op-arith-op.h, 399
- cpu/or32/op-comp-op.h, 400
- cpu/or32/op-extend-op.h, 401
- cpu/or32/op-ff1-op.h, 402
- cpu/or32/op-i386.h, 403
- cpu/or32/op-lwhb-op.h, 404
- cpu/or32/op-mac-op.h, 405
- cpu/or32/op-mftspr-op.h, 406
- cpu/or32/op-support.c, 407
- cpu/or32/op-support.h, 409
- cpu/or32/op-swhb-op.h, 411
- cpu/or32/op-t-reg-mov-op.h, 412
- cpu/or32/op.c, 416
- cpu/or32/or32.c, 430
- cpu/or32/rec-i386.h, 437
- cpu/or32/sched-i386.h, 438
- cpu/or32/simpl32-defs.h, 439
- cpu_cfg
 - cpu-config.c, 207
- cpu_cfg
 - cpu-config.c, 207
- cpu_clock
 - execute.c, 368
 - execute.h, 262
- cpu_dependstats
 - cpu-config.c, 207
- cpu_hazards
 - cpu-config.c, 207
- cpu_reset
 - execute.c, 369
 - execute.h, 263
- cpu_rev
 - cpu-config.c, 207
- cpu_sbuf_len
 - cpu-config.c, 207
- cpu_sr
 - cpu-config.c, 207
- cpu_state, 60
 - delay_insn, 61
 - execute.c, 375
 - execute.h, 267
 - icomplet, 61
 - insn_ea, 60
 - iqueue, 61
 - pc, 61
 - pc_delay, 61
 - pic_lines, 61
 - reg, 60
 - sprs, 60
- CPU_STATE_REG
 - i386-regs.h, 380
- cpu_superscalar
 - cpu-config.c, 208
- cpu_upr
 - cpu-config.c, 208
- cpu_ver
 - cpu-config.c, 208
- cpy_bb
 - bb.c, 443
- crc32
 - crc32.c, 678
 - crc32.h, 679
- crc32.c
 - crc32, 678
 - crc32_close, 678
 - crc32_feed_bytes, 678
 - crc32_init, 678
 - crc32_table, 678
- crc32.h
 - crc32, 679
 - crc32_close, 679
 - crc32_feed_bytes, 679
 - crc32_init, 679
- crc32_close
 - crc32.c, 678
 - crc32.h, 679
- crc32_feed_bytes
 - crc32.c, 678
 - crc32.h, 679

- crc32_init
 - crc32.c, 678
 - crc32.h, 679
 - crc32_table
 - crc32.c, 678
 - crc_dly
 - eth_device, 113
 - crc_value
 - eth_device, 113
 - cs
 - mc_area, 148
 - csc
 - mc, 147
 - cse
 - cuc.h, 471
 - insn.c, 484
 - csm
 - cuc.h, 471
 - insn.c, 484
 - csm_gen
 - cuc.h, 471
 - insn.c, 484
 - csr
 - dma_channel, 81
 - dma_controller, 84
 - mc, 147
 - CT_NONE
 - abstract.h, 231
 - CT_PHYSICAL
 - abstract.h, 231
 - CT_VIRTUAL
 - abstract.h, 231
 - ctrl
 - ata_host, 26
 - fb_state, 118
 - gpio_device, 124
 - vga_state, 180
 - ctrl_c
 - toplevel-support.c, 853
 - toplevel-support.h, 858
 - cuc
 - config, 55
 - runtime, 165
 - cuc.c
 - analyse_function, 451
 - calc_cycles, 452
 - calc_size, 452
 - caller_saved, 462
 - cuc_calling_conv, 452
 - cuc_debug, 463
 - cuc_enable_bursts, 452
 - cuc_memory_order, 452
 - cuc_no_multicycle, 452
 - cuc_optimize, 453
 - cuc_timings_fn, 453
 - extract_function, 454
 - flog, 463
 - format_func_options, 454
 - func, 463
 - func_v, 463
 - gen_option, 454
 - generate_function, 456
 - main_cuc, 457
 - option_char, 463
 - options_cmd, 459
 - preunroll_bb, 459
 - print_option, 462
 - reg_cuc_sec, 462
 - set_func_deps, 462
 - tim_comp, 462
 - cuc.h
 - add_data_dep, 469
 - add_dep, 470
 - add_latches, 470
 - add_memory_dep, 470
 - analyse_timings, 470
 - BB_DEAD, 469
 - BB_INLOOP, 469
 - BB_OPTIONAL, 469
 - BBID_END, 469
 - BBID_START, 469
 - build_bb, 470
 - caller_saved, 481
 - clean_deps, 471
 - cse, 471
 - csm, 471
 - csm_gen, 471
 - cuc_check, 471
 - cuc_debug, 481
 - cuc_func, 469
 - cuc_load, 471
 - CUC_MAX_STACK, 469
 - cuc_shared_list, 469
 - CUC_WIDTH_ITERATIONS, 469
 - cucdebug, 469
 - dep_list, 469
 - detect_bb, 472
 - detect_max_values, 472
 - dispose_list, 472
 - dup_func, 473
 - expand_bb, 473
 - FLAG_REG, 469
 - flog, 481
 - free_func, 473
 - generate_bb_seq, 473
 - insert_conditional_facts, 473
 - insert_insns, 474
 - INSN, 469

- insn, 481
- IT_BBEND, 469
- IT_BBSTART, 469
- IT_BRANCH, 469
- IT_COND, 469
- IT_CUT, 469
- IT_FLAG1, 469
- IT_FLAG2, 469
- IT_INDELAY, 469
- IT_LATCHED, 469
- IT_MEMADD, 469
- IT_MEMORY, 469
- IT_OUTPUT, 469
- IT_SIGNED, 469
- IT_UNUSED, 469
- IT_VOLATILE, 469
- log, 469
- LRBB_REG, 469
- main_cuc, 474
- mark_cut, 477
- MAX, 469
- MAX_BB, 469
- MAX_INSNS, 469
- MAX_PREROLL, 469
- MAX_REGS, 469
- MAX_UNROLL, 469
- MIN, 469
- MO_EXACT, 469
- MO_NONE, 469
- MO_STRONG, 469
- MO_WEAK, 469
- MT_BURST, 469
- MT_BURSTE, 469
- MT_CALL, 469
- MT_LOAD, 469
- MT_SIGNED, 469
- MT_STORE, 469
- MT_WIDTH, 469
- negate_conditional, 477
- num_insn, 481
- OPT_BB, 469
- OPT_CONST, 469
- OPT_DEST, 469
- OPT_JUMP, 469
- OPT_LRBB, 469
- OPT_NONE, 469
- OPT_REF, 469
- OPT_REGISTER, 469
- optimize_bb, 477
- optimize_cmovs, 477
- optimize_tree, 477
- preunroll_loop, 478
- print_bb_num, 478
- print_cuc_bb, 478
- print_cuc_insns, 478
- print_insns, 479
- recalc_cnts, 479
- REF, 469
- REF_BB, 469
- REF_I, 469
- reg_cuc_sec, 479
- reg_dep, 479
- reloc, 481
- remove_dead, 480
- remove_dead_bb, 480
- remove_nops, 480
- remove_trivial_regs, 480
- schedule_memory, 480
- set_io, 481
- cuc/adv.c, 440
- cuc/bb.c, 442
- cuc/cuc.c, 448
- cuc/cuc.h, 464
- cuc/insn.c, 482
- cuc/insn.h, 488
- cuc/load.c, 494
- cuc/memory.c, 499
 - add_memory_dep, 500
 - check_memory_conflict, 500
 - clean_deps, 500
 - join_transfers, 500
 - mem_ordering_cmp, 500
 - same_transfers, 500
 - schedule_memory, 500
- cuc/timings.c, 507
- cuc/verilog.c, 510
- cuc/verilog.h, 513
- cuc_bb, 62
 - cnt, 63
 - first, 63
 - insn, 63
 - last, 63
 - last_used_reg, 63
 - mdep, 63
 - next, 63
 - ninsn, 63
 - nmemory, 63
 - ntim, 63
 - prev, 63
 - selected_tim, 63
 - tim, 63
 - tmp, 63
 - type, 63
 - unrolled, 63
- cuc_calling_conv
 - cuc.c, 452
- cuc_check
 - bb.c, 443

- cuc.h, 471
- cuc_conv, 64
 - from, 64
 - to, 64
- cuc_debug
 - cuc.c, 463
 - cuc.h, 481
- cuc_enable_bursts
 - cuc.c, 452
- cuc_func
 - cuc.h, 469
- cuc_insn, 65
 - dep, 65
 - disasm, 65
 - index, 65
 - insn, 65
 - max, 65
 - op, 65
 - opt, 65
 - tmp, 65
 - type, 65
- cuc_insn_name
 - insn.c, 484
 - insn.h, 492
- cuc_known_insn, 66
 - comutative, 66
 - name, 66
 - rtl, 66
- cuc_load
 - cuc.h, 471
 - load.c, 495
- CUC_MAX_STACK
 - cuc.h, 469
- cuc_memory_order
 - cuc.c, 452
- cuc_no_multicycle
 - cuc.c, 452
- cuc_optimize
 - cuc.c, 453
- cuc_shared_item, 67
 - cmatch, 67
 - ref, 67
- cuc_shared_list
 - cuc.h, 469
- cuc_timing_table, 68
 - delay, 68
 - delayi, 68
 - size, 68
 - sizei, 68
- cuc_timings, 69
 - b, 69
 - new_time, 69
 - nshared, 69
 - preroll, 69
 - shared, 69
 - size, 69
 - unroll, 69
- cuc_timings_fn
 - cuc.c, 453
- CUC_WIDTH_ITERATIONS
 - cuc.h, 469
- cucdebug
 - cuc.h, 469
- cum_cycles
 - func_struct, 122
- cumulative
 - profiler.c, 779
- cur_area
 - abstract.c, 228
 - abstract.h, 243
- cur_section
 - sim-config.c, 808
 - sim-config.h, 814
- cur_vadd
 - abstract.c, 228
- CURINSN
 - execute.h, 262
- curpass
 - or32.c, 435
- curr
 - gpio_device, 124
- current_descriptor
 - dma_channel, 81
- current_scan_chain
 - debug-unit.c, 523
- cut_tree
 - timings.c, 508
- cycle_duration
 - runtime, 165
- cycles
 - runtime, 165
 - stack_struct, 175
- cycles_start
 - tick.c, 842
- cylinder_high
 - ata_device, 22
- cylinder_low
 - ata_device, 22
- d_ptr
 - dynamic, 90
 - Elf64_Dyn, 99
- d_tag
 - dynamic, 90
 - Elf64_Dyn, 99
- d_un
 - dynamic, 90
 - Elf64_Dyn, 99

- d_val
 - dynamic, 90
 - Elf64_Dyn, 99
- DADDR_PAGE
 - dmmu.h, 577
- daspi
 - ata_device, 22
- daspo
 - ata_device, 22
- dat
 - config_section, 59
 - sim_reset_hook, 170
 - sim_stat, 171
- data
 - channel, 37
 - jtr_read_block_response, 137
 - jtr_write_block_message, 140
 - rsp_buf, 162
- data_ci
 - abstract.c, 228
 - abstract.h, 243
- data_h
 - jtr_read_response, 139
 - jtr_write_message, 142
- data_l
 - jtr_read_response, 139
 - jtr_write_message, 142
- data_start
 - COFF_AOUTHDR, 40
- dataport_i
 - ata_device, 22
- dbuf
 - ata_device, 22
- dbuf_cnt
 - ata_device, 22
- dbuf_ptr
 - ata_device, 22
- dc
 - config, 55
 - dcache-model.c, 194
- dc_blocksize
 - dcache-model.c, 191
- dc_enabled
 - dcache-model.c, 191
- dc_info
 - dcache-model.c, 191
 - dcache-model.h, 196
- dc_inv
 - dcache-model.c, 191
 - dcache-model.h, 196
- dc_load_hitdelay
 - dcache-model.c, 191
- dc_load_missdelay
 - dcache-model.c, 191
- dc_nsets
 - dcache-model.c, 191
- dc_nways
 - dcache-model.c, 192
- dc_set, 70
 - line, 70
 - lru, 70
 - tagaddr, 70
 - way, 70
- dc_simulate_read
 - dcache-model.c, 192
 - dcache-model.h, 196
- dc_simulate_write
 - dcache-model.c, 192
 - dcache-model.h, 196
- dc_stats
 - stats.c, 282
 - stats.h, 285
- dc_store_hitdelay
 - dcache-model.c, 193
- dc_store_missdelay
 - dcache-model.c, 193
- dc_ustates
 - dcache-model.c, 193
- dcache-model.c
 - dc, 194
 - dc_blocksize, 191
 - dc_enabled, 191
 - dc_info, 191
 - dc_inv, 191
 - dc_load_hitdelay, 191
 - dc_load_missdelay, 191
 - dc_nsets, 191
 - dc_nways, 192
 - dc_simulate_read, 192
 - dc_simulate_write, 192
 - dc_store_hitdelay, 193
 - dc_store_missdelay, 193
 - dc_ustates, 193
 - reg_dc_sec, 193
- dcache-model.h
 - dc_info, 196
 - dc_inv, 196
 - dc_simulate_read, 196
 - dc_simulate_write, 196
 - MAX_DC_BLOCK_SIZE, 196
 - MAX_DC_SETS, 196
 - MAX_DC_WAYS, 196
 - MIN_DC_BLOCK_SIZE, 196
 - reg_dc_sec, 197
- dead
 - _csm_list, 14
- debug
 - config, 55

- debug.c, [817](#)
- debug.h, [820](#)
- debug-unit.c
 - calculate_watchpoints, [516](#)
 - check_debug_unit, [516](#)
 - current_scan_chain, [523](#)
 - debug_enabled, [516](#)
 - debug_gdb_enabled, [517](#)
 - debug_get_mem, [517](#)
 - debug_get_register, [517](#)
 - debug_ignore_exception, [518](#)
 - debug_rsp_enabled, [518](#)
 - debug_rsp_port, [519](#)
 - debug_server_port, [519](#)
 - debug_set_chain, [519](#)
 - debug_set_mem, [519](#)
 - debug_set_register, [520](#)
 - debug_vapi_id, [520](#)
 - DEVELOPINT_MAX, [516](#)
 - DEVELOPINT_RISCOP, [516](#)
 - development, [523](#)
 - development_interface_address_space, [516](#)
 - du_reset, [521](#)
 - get_devint_reg, [521](#)
 - in_reset, [523](#)
 - reg_debug_sec, [521](#)
 - RISCOP_RESET, [515](#)
 - RISCOP_STALL, [515](#)
 - set_devint_reg, [522](#)
 - set_stall_state, [523](#)
- debug-unit.h
 - check_debug_unit, [525](#)
 - debug_get_register, [525](#)
 - debug_ignore_exception, [526](#)
 - debug_scan_chain_ids, [524](#)
 - debug_set_chain, [526](#)
 - debug_set_register, [526](#)
 - debug_unit_action, [524](#)
 - DebugInstructionFetch, [525](#)
 - DebugLoadAddress, [525](#)
 - DebugLoadData, [525](#)
 - DebugStoreAddress, [525](#)
 - DebugStoreData, [525](#)
 - du_reset, [527](#)
 - JTAG_CHAIN_DEBUG_UNIT, [524](#)
 - JTAG_CHAIN_DEVELOPMENT, [524](#)
 - JTAG_CHAIN_GLOBAL, [524](#)
 - JTAG_CHAIN_TRACE, [524](#)
 - JTAG_CHAIN_WISHBONE, [524](#)
 - reg_debug_sec, [527](#)
 - set_stall_state, [528](#)
- debug.c
 - __ORSIM_NO_DEC_DBCH, [817](#)
 - __orsim_dbchs, [817](#)
- debug, [817](#)
- debug_classes, [817](#)
- DECLARE_DEBUG_CHANNEL, [817](#)
- orsim_dbcl_set, [817](#)
- orsim_dbcl_set_name, [817](#)
- orsim_dbg_log, [817](#)
- parse_dbchs, [817](#)
- debug.h
 - __ORSIM_DBCL_ERR, [820](#)
 - __ORSIM_DBCL_FIXME, [820](#)
 - __ORSIM_DBCL_TRACE, [820](#)
 - __ORSIM_DBCL_WARN, [820](#)
 - __ORSIM_DBG_USE_FUNC, [819](#)
 - __ORSIM_DEBUG_CLASS, [820](#)
 - __ORSIM_DEBUG_LOG, [819](#)
 - __ORSIM_DPRINTF, [819](#)
 - __ORSIM_GET_DEBUGGING, [819](#)
 - __ORSIM_GET_DEBUGGING_ERR, [819](#)
 - __ORSIM_GET_DEBUGGING_FIXME, [819](#)
 - __ORSIM_GET_DEBUGGING_TRACE, [819](#)
 - __ORSIM_GET_DEBUGGING_WARN, [819](#)
 - debug, [820](#)
 - DECLARE_DEBUG_CHANNEL, [819](#)
 - DEFAULT_DEBUG_CHANNEL, [819](#)
 - ERR, [819](#)
 - ERR_, [820](#)
 - ERR_ON, [820](#)
 - FIXME, [820](#)
 - FIXME_, [820](#)
 - FIXME_ON, [820](#)
 - orsim_dbcl_set_name, [820](#)
 - orsim_dbg_log, [821](#)
 - parse_dbchs, [821](#)
 - TRACE, [820](#)
 - TRACE_, [820](#)
 - TRACE_ON, [820](#)
 - WARN, [820](#)
 - WARN_, [820](#)
 - WARN_ON, [820](#)
- debug/debug-unit.c, [514](#)
- debug/debug-unit.h, [524](#)
- debug/gdb.h, [529](#)
- debug/gdbcomm.c, [531](#)
- debug/gdbcomm.h, [535](#)
- debug/rsp-server.c, [537](#)
- debug/rsp-server.h, [560](#)
- debug_classes
 - debug.c, [817](#)
- debug_enabled
 - debug-unit.c, [516](#)
- debug_gdb_enabled
 - debug-unit.c, [517](#)
- debug_get_mem
 - debug-unit.c, [517](#)

- debug_get_register
 - debug-unit.c, 517
 - debug-unit.h, 525
- debug_ignore_exception
 - debug-unit.c, 518
 - debug-unit.h, 526
- debug_rsp_enabled
 - debug-unit.c, 518
- debug_rsp_port
 - debug-unit.c, 519
- debug_scan_chain_ids
 - debug-unit.h, 524
- debug_server_port
 - debug-unit.c, 519
- debug_set_chain
 - debug-unit.c, 519
 - debug-unit.h, 526
- debug_set_mem
 - debug-unit.c, 519
- debug_set_register
 - debug-unit.c, 520
 - debug-unit.h, 526
- debug_unit_action
 - debug-unit.h, 524
- debug_vapi_id
 - debug-unit.c, 520
- DebugInstructionFetch
 - debug-unit.h, 525
- DebugLoadAddress
 - debug-unit.h, 525
- DebugLoadData
 - debug-unit.h, 525
- DebugStoreAddress
 - debug-unit.h, 525
- DebugStoreData
 - debug-unit.h, 525
- DECLARE_DEBUG_CHANNEL
 - debug.c, 817
 - debug.h, 819
 - sched.c, 833
 - sprs.c, 322
- decode_execute
 - execute.c, 369
- decode_execute_wrapper
 - execute.c, 369
- def-op-t.h
 - DEF_1T_OP, 326
 - DEF_2T_OP, 326
 - DEF_2T_OP_NEQ, 326
 - DEF_3T_OP, 326
 - DEF_3T_OP_NEQ, 327
 - DEF_GPR_OP, 327
 - GPR_T, 327
- DEF_1T_OP
 - def-op-t.h, 326
- def-op-t.h, 326
- dyn-rec.c, 335
- DEF_2T_OP
 - def-op-t.h, 326
 - dyn-rec.c, 335
- DEF_2T_OP_NEQ
 - def-op-t.h, 326
 - dyn-rec.c, 335
- DEF_3T_OP
 - def-op-t.h, 326
 - dyn-rec.c, 335
- DEF_3T_OP_NEQ
 - def-op-t.h, 327
 - dyn-rec.c, 335
- DEF_GPR_OP
 - def-op-t.h, 327
 - dyn-rec.c, 335
- DEFAULT_BAUD
 - tty.c, 673
- DEFAULT_DEBUG_CHANNEL
 - debug.h, 819
- DEFAULT_MEMORY_LEN
 - abstract.h, 231
- DEFAULT_MEMORY_START
 - abstract.h, 231
- DEFAULT_TTY_DEVICE
 - tty.c, 673
- delay
 - cuc_timing_table, 68
- delay_insn
 - cpu_state, 61
- delayi
 - cuc_timing_table, 68
- delayr
 - dyn_page, 89
 - mem_config, 150
 - mem_ops, 152
- delayw
 - mem_config, 150
 - mem_ops, 152
- dep
 - cuc_insn, 65
- dep_list
 - cuc.h, 469
- depend
 - dstats_entry, 88
 - fstats_entry, 121
- depend_operands
 - execute.h, 263
- dependstats
 - config, 55
- desc
 - dma_channel, 81
- destination

- dma_channel, 81
- destination_mask
 - dma_channel, 81
- destruct_automata
 - or32.c, 432
- detect_bb
 - bb.c, 443
 - cuc.h, 472
- detect_locals
 - load.c, 496
- detect_max_values
 - adv.c, 441
 - cuc.h, 472
- dev
 - ata_device, 22
- dev_generic
 - GENERIC_BYTE, 76
 - GENERIC_HW, 76
 - GENERIC_READ, 76
 - GENERIC_WORD, 76
 - GENERIC_WRITE, 76
- dev_16450, 71
 - baseaddr, 74
 - channel, 74
 - channel_str, 74
 - char_clks, 74
 - dlh, 74
 - dll, 74
 - enabled, 74
 - fcr, 74
 - fifo_len, 74
 - ier, 74
 - iir, 74
 - ints, 74
 - iregs, 74
 - irq, 74
 - istat, 74
 - jitter, 74
 - lcr, 74
 - loopback, 74
 - lsr, 74
 - mcr, 74
 - msr, 74
 - receiveng, 74
 - recv_break, 74
 - regs, 74
 - rxbuf, 74
 - rxbuf_full, 74
 - rxbuf_head, 74
 - rxbuf_tail, 74
 - rxser, 74
 - scr, 74
 - skew, 74
 - txbuf, 74
 - txbuf_full, 74
 - txbuf_head, 74
 - txbuf_tail, 74
 - txser, 74
 - uart16550, 74
 - vapi, 74
 - vapi_buf, 74
 - vapi_buf_head_ptr, 74
 - vapi_buf_tail_ptr, 74
 - vapi_id, 74
- dev_generic, 76
 - baseaddr, 77
 - byte_enabled, 77
 - enabled, 77
 - hw_enabled, 77
 - name, 77
 - size, 77
 - trans_direction, 77
 - trans_size, 77
 - value, 77
 - word_enabled, 77
- dev_id
 - ata_host, 26
- dev_list
 - abstract.c, 228
- dev_memarea, 78
 - addr_compare, 79
 - addr_mask, 79
 - direct_ops, 79
 - log, 79
 - next, 79
 - ops, 79
 - size, 79
 - size_mask, 79
 - valid, 79
- dev_sel
 - ata_host, 26
- DEVELOPINT_MAX
 - debug-unit.c, 516
- DEVELOPINT_RISCOMP
 - debug-unit.c, 516
- development
 - debug-unit.c, 523
- development_interface_address_space
 - debug-unit.c, 516
- device
 - ata_devices, 24
- device_control
 - ata_device, 22
- device_head
 - ata_device, 22
- DEVICE_RESET
 - atacmd.h, 624
- devices

- ata_host, 26
- direct_ops
 - dev_memarea, 79
- dirty
 - dyn_page, 89
- dirtyfy_page
 - dyn-rec.c, 335
- DISABLE_ADVANCED_POWER_-
MANAGEMENT
 - atacmd.h, 624
- DISABLE_MEDIA_STATUS_NOTIFICATION
 - atacmd.h, 624
- DISABLE_POWERUP_IN_STANDBY_-
FEATURE_SET
 - atacmd.h, 624
- DISABLE_READ_LOOKAHEAD
 - atacmd.h, 624
- DISABLE_RELEASE_INTERRUPT
 - atacmd.h, 624
- DISABLE_REVERTING_TO_POWERON_-
DEFAULTS
 - atacmd.h, 624
- DISABLE_SERVICE_INTERRUPT
 - atacmd.h, 624
- DISABLE_WRITE_CACHE
 - atacmd.h, 624
- disasm
 - cuc_insn, 65
- disassemble_index
 - or32.c, 433
- disassemble_insn
 - or32.c, 433
- disassemble_memory
 - abstract.c, 213
 - abstract.h, 231
- disassembled
 - or32.c, 435
- disassembled_str
 - or32.c, 435
- dispose_list
 - cuc.h, 472
 - insn.c, 485
- DISWIDTH
 - dumpverilog.c, 823
- dlh
 - dev_16450, 74
- dll
 - dev_16450, 74
- dma
 - eth_device, 113
- dma-defs.h
 - DMA_ADDR_SPACE, 682
 - DMA_CH_A0, 682
 - DMA_CH_A0_ADDR_OFFSET, 682
 - DMA_CH_A0_ADDR_WIDTH, 682
 - DMA_CH_A1, 682
 - DMA_CH_A1_ADDR_OFFSET, 682
 - DMA_CH_A1_ADDR_WIDTH, 682
 - DMA_CH_AM0, 682
 - DMA_CH_AM0_MASK_OFFSET, 682
 - DMA_CH_AM0_MASK_WIDTH, 682
 - DMA_CH_AM1, 682
 - DMA_CH_AM1_MASK_OFFSET, 682
 - DMA_CH_AM1_MASK_WIDTH, 682
 - DMA_CH_BASE, 682
 - DMA_CH_CSR, 682
 - DMA_CH_CSR_ARS_OFFSET, 684
 - DMA_CH_CSR_BUSY_OFFSET, 684
 - DMA_CH_CSR_CH_EN_OFFSET, 684
 - DMA_CH_CSR_DONE_OFFSET, 684
 - DMA_CH_CSR_DST_SEL_OFFSET, 684
 - DMA_CH_CSR_ERR_OFFSET, 684
 - DMA_CH_CSR_INC_DST_OFFSET, 684
 - DMA_CH_CSR_INC_SRC_OFFSET, 684
 - DMA_CH_CSR_INE_CHK_DONE_-
OFFSET, 684
 - DMA_CH_CSR_INE_DONE_OFFSET, 684
 - DMA_CH_CSR_INE_ERR_OFFSET, 684
 - DMA_CH_CSR_INT_CHUNK_DONE_-
OFFSET, 684
 - DMA_CH_CSR_INT_DONE_OFFSET, 684
 - DMA_CH_CSR_INT_ERR_OFFSET, 684
 - DMA_CH_CSR_MODE_OFFSET, 684
 - DMA_CH_CSR_PRIORITY_OFFSET, 684
 - DMA_CH_CSR_PRIORITY_WIDTH, 684
 - DMA_CH_CSR_RESERVED_OFFSET, 684
 - DMA_CH_CSR_RESERVED_WIDTH, 684
 - DMA_CH_CSR_REST_EN_OFFSET, 684
 - DMA_CH_CSR_SRC_SEL_OFFSET, 684
 - DMA_CH_CSR_STOP_OFFSET, 684
 - DMA_CH_CSR_SZ_WB_OFFSET, 684
 - DMA_CH_CSR_USE_ED_OFFSET, 684
 - DMA_CH_CSR_WRITE_MASK, 684
 - DMA_CH_DESC, 684
 - DMA_CH_DESC_ADDR_OFFSET, 684
 - DMA_CH_DESC_ADDR_WIDTH, 684
 - DMA_CH_SIZE, 684
 - DMA_CH_SWPTR, 684
 - DMA_CH_SWPTR_EN_OFFSET, 686
 - DMA_CH_SWPTR_PTR_OFFSET, 686
 - DMA_CH_SWPTR_PTR_WIDTH, 686
 - DMA_CH_SZ, 686
 - DMA_CH_SZ_CHK_SZ_OFFSET, 686
 - DMA_CH_SZ_CHK_SZ_WIDTH, 686
 - DMA_CH_SZ_TOT_SZ_OFFSET, 686
 - DMA_CH_SZ_TOT_SZ_WIDTH, 686
 - DMA_CSR, 686
 - DMA_CSR_PAUSE_OFFSET, 686

- DMA_DESC_ADR0, 686
- DMA_DESC_ADR1, 686
- DMA_DESC_CSR, 686
- DMA_DESC_CSR_DST_SEL_OFFSET, 686
- DMA_DESC_CSR_EOL_OFFSET, 686
- DMA_DESC_CSR_INC_DST_OFFSET, 686
- DMA_DESC_CSR_INC_SRC_OFFSET, 686
- DMA_DESC_CSR_SRC_SEL_OFFSET, 686
- DMA_DESC_CSR_TOT_SZ_OFFSET, 686
- DMA_DESC_CSR_TOT_SZ_WIDTH, 686
- DMA_DESC_NEXT, 686
- DMA_INT_MSK_A, 686
- DMA_INT_MSK_B, 686
- DMA_INT_SRC_A, 686
- DMA_INT_SRC_B, 686
- DMA_NUM_CHANNELS, 686
- dma.c
 - CHANNEL_ND_I, 689
 - check_dma_ack_o, 689
 - clear_dma_nd_i, 689
 - clear_dma_req_i, 689
 - dma_baseaddr, 689
 - dma_channel_clock, 689
 - dma_channel_terminate_transfer, 689
 - dma_enabled, 690
 - dma_init_transfer, 690
 - dma_irq, 690
 - dma_load_descriptor, 690
 - dma_read32, 690
 - dma_read_ch_csr, 691
 - dma_reset, 691
 - dma_sec_end, 691
 - dma_sec_start, 691
 - dma_status, 691
 - dma_vapi_id, 692
 - dma_write32, 692
 - dma_write_ch_csr, 692
 - dmass, 693
 - find_dma_controller_ch, 692
 - masked_increase, 693
 - reg_dma_sec, 693
 - set_dma_nd_i, 693
 - set_dma_req_i, 693
- dma.h
 - check_dma_ack_o, 695
 - clear_dma_nd_i, 695
 - clear_dma_req_i, 695
 - find_dma_controller_ch, 695
 - reg_dma_sec, 695
 - set_dma_nd_i, 695
 - set_dma_req_i, 695
- dma_ack_o
 - dma_channel, 81
- DMA_ADDR_SPACE
 - dma-defs.h, 682
 - dma_baseaddr
 - dma.c, 689
 - DMA_CH_A0
 - dma-defs.h, 682
 - DMA_CH_A0_ADDR_OFFSET
 - dma-defs.h, 682
 - DMA_CH_A0_ADDR_WIDTH
 - dma-defs.h, 682
 - DMA_CH_A1
 - dma-defs.h, 682
 - DMA_CH_A1_ADDR_OFFSET
 - dma-defs.h, 682
 - DMA_CH_A1_ADDR_WIDTH
 - dma-defs.h, 682
 - DMA_CH_AM0
 - dma-defs.h, 682
 - DMA_CH_AM0_MASK_OFFSET
 - dma-defs.h, 682
 - DMA_CH_AM0_MASK_WIDTH
 - dma-defs.h, 682
 - DMA_CH_AM1
 - dma-defs.h, 682
 - DMA_CH_AM1_MASK_OFFSET
 - dma-defs.h, 682
 - DMA_CH_AM1_MASK_WIDTH
 - dma-defs.h, 682
 - DMA_CH_BASE
 - dma-defs.h, 682
 - DMA_CH_CSR
 - dma-defs.h, 682
 - DMA_CH_CSR_ARS_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_BUSY_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_CH_EN_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_DONE_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_DST_SEL_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_ERR_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_INC_DST_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_INC_SRC_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_INE_CHK_DONE_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_INE_DONE_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_INE_ERR_OFFSET
 - dma-defs.h, 684
 - DMA_CH_CSR_INT_CHUNK_DONE_OFFSET

- dma-defs.h, 684
- DMA_CH_CSR_INT_DONE_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_INT_ERR_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_MODE_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_PRIORITY_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_PRIORITY_WIDTH
 - dma-defs.h, 684
- DMA_CH_CSR_RESERVED_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_RESERVED_WIDTH
 - dma-defs.h, 684
- DMA_CH_CSR_REST_EN_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_SRC_SEL_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_STOP_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_SZ_WB_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_USE_ED_OFFSET
 - dma-defs.h, 684
- DMA_CH_CSR_WRITE_MASK
 - dma-defs.h, 684
- DMA_CH_DESC
 - dma-defs.h, 684
- DMA_CH_DESC_ADDR_OFFSET
 - dma-defs.h, 684
- DMA_CH_DESC_ADDR_WIDTH
 - dma-defs.h, 684
- DMA_CH_SIZE
 - dma-defs.h, 684
- DMA_CH_SWPTR
 - dma-defs.h, 684
- DMA_CH_SWPTR_EN_OFFSET
 - dma-defs.h, 686
- DMA_CH_SWPTR_PTR_OFFSET
 - dma-defs.h, 686
- DMA_CH_SWPTR_PTR_WIDTH
 - dma-defs.h, 686
- DMA_CH_SZ
 - dma-defs.h, 686
- DMA_CH_SZ_CHK_SZ_OFFSET
 - dma-defs.h, 686
- DMA_CH_SZ_CHK_SZ_WIDTH
 - dma-defs.h, 686
- DMA_CH_SZ_TOT_SZ_OFFSET
 - dma-defs.h, 686
- DMA_CH_SZ_TOT_SZ_WIDTH
 - dma-defs.h, 686
- dma_channel, 80
- a0, 81
- a1, 81
- am0, 81
- am1, 81
- channel_mask, 81
- channel_number, 81
- chunk_size, 81
- controller, 81
- csr, 81
- current_descriptor, 81
- desc, 81
- destination, 81
- destination_mask, 81
- dma_ack_o, 81
- dma_nd_i, 81
- dma_req_i, 81
- load_next_descriptor_when_done, 81
- referenced, 81
- regs, 81
- source, 81
- source_mask, 81
- swptr, 81
- sz, 81
- total_size, 81
- words_transferred, 81
- dma_channel_clock
 - dma.c, 689
- dma_channel_terminate_transfer
 - dma.c, 689
- dma_controller, 83
 - baseaddr, 84
 - ch, 84
 - csr, 84
 - enabled, 84
 - int_msk_a, 84
 - int_msk_b, 84
 - int_src_a, 84
 - int_src_b, 84
 - irq, 84
 - next, 84
 - regs, 84
 - vapi_id, 84
- DMA_CSR
 - dma-defs.h, 686
- DMA_CSR_PAUSE_OFFSET
 - dma-defs.h, 686
- DMA_DESC_ADR0
 - dma-defs.h, 686
- DMA_DESC_ADR1
 - dma-defs.h, 686
- DMA_DESC_CSR
 - dma-defs.h, 686
- DMA_DESC_CSR_DST_SEL_OFFSET
 - dma-defs.h, 686

- DMA_DESC_CSR_EOL_OFFSET
 - dma-defs.h, 686
- DMA_DESC_CSR_INC_DST_OFFSET
 - dma-defs.h, 686
- DMA_DESC_CSR_INC_SRC_OFFSET
 - dma-defs.h, 686
- DMA_DESC_CSR_SRC_SEL_OFFSET
 - dma-defs.h, 686
- DMA_DESC_CSR_TOT_SZ_OFFSET
 - dma-defs.h, 686
- DMA_DESC_CSR_TOT_SZ_WIDTH
 - dma-defs.h, 686
- DMA_DESC_NEXT
 - dma-defs.h, 686
- dma_enabled
 - dma.c, 690
- dma_init_transfer
 - dma.c, 690
- DMA_INT_MSK_A
 - dma-defs.h, 686
- DMA_INT_MSK_B
 - dma-defs.h, 686
- DMA_INT_SRC_A
 - dma-defs.h, 686
- DMA_INT_SRC_B
 - dma-defs.h, 686
- dma_irq
 - dma.c, 690
- dma_load_descriptor
 - dma.c, 690
- dma_mode
 - ata_device, 22
- DMA_MODE0_TD
 - atahost.c, 646
- dma_mode0_td
 - ata_host, 26
- DMA_MODE0_TEOC
 - atahost.c, 646
- dma_mode0_teoc
 - ata_host, 26
- DMA_MODE0_TM
 - atahost.c, 646
- dma_mode0_tm
 - ata_host, 26
- dma_nd_i
 - dma_channel, 81
- DMA_NUM_CHANNELS
 - dma-defs.h, 686
- dma_read32
 - dma.c, 690
- dma_read_ch_csr
 - dma.c, 691
- dma_req_i
 - dma_channel, 81
- dma_reset
 - dma.c, 691
- dma_sec_end
 - dma.c, 691
- dma_sec_start
 - dma.c, 691
- dma_status
 - dma.c, 691
- dma_vapi_id
 - dma.c, 692
- dma_write32
 - dma.c, 692
- dma_write_ch_csr
 - dma.c, 692
- dmarq
 - ata_device, 22
- dmass
 - dma.c, 693
- dmmu, 85
 - enabled, 86
 - entrysize, 86
 - hitdelay, 86
 - lru_reload, 86
 - missdelay, 86
 - nsets, 86
 - nways, 86
 - page_mask, 86
 - page_offset_mask, 86
 - pagesize, 86
 - pagesize_log2, 86
 - set_mask, 86
 - ustates, 86
 - vpn_mask, 86
- dmmu.c
 - dmmu_enabled, 571
 - dmmu_end_sec, 571
 - dmmu_entrysize, 571
 - dmmu_find_tlbmr, 571
 - dmmu_hitdelay, 572
 - dmmu_missdelay, 572
 - dmmu_nsets, 572
 - dmmu_nways, 572
 - dmmu_pagesize, 572
 - dmmu_start_sec, 573
 - dmmu_state, 575
 - dmmu_translate, 573
 - dmmu_ustates, 573
 - dtlb_status, 574
 - peek_into_dtlb, 574
 - reg_dmmu_sec, 574
- dmmu.h
 - DADDR_PAGE, 577
 - dmmu_simulate_tlb, 577
 - dmmu_state, 578

- dmmu_translate, 577
- peek_into_dtlb, 577
- reg_dmmu_sec, 577
- dmmu_enabled
 - dmmu.c, 571
- dmmu_end_sec
 - dmmu.c, 571
- dmmu_entrysize
 - dmmu.c, 571
- dmmu_find_tlbmr
 - dmmu.c, 571
- dmmu_hitdelay
 - dmmu.c, 572
- dmmu_missdelay
 - dmmu.c, 572
- dmmu_nsets
 - dmmu.c, 572
- dmmu_nways
 - dmmu.c, 572
- dmmu_pagesize
 - dmmu.c, 572
- dmmu_simulate_tlb
 - dmmu.h, 577
- dmmu_start_sec
 - dmmu.c, 573
- dmmu_state
 - dmmu.c, 575
 - dmmu.h, 578
- dmmu_stats
 - stats.c, 282
 - stats.h, 285
- dmmu_translate
 - dmmu.c, 573
 - dmmu.h, 577
- dmmu_ustates
 - dmmu.c, 573
- dmmustats_entry, 87
 - loads_pagefaults, 87
 - loads_tlbhit, 87
 - loads_tlbmiss, 87
 - stores_pagefaults, 87
 - stores_tlbhit, 87
 - stores_tlbmiss, 87
- do_jump
 - op-support.c, 408
 - op-support.h, 409
- do_sched_wrap
 - op.c, 423
- do_sched_wrap_delay
 - op.c, 424
- do_scheduler
 - sched.c, 833
 - sched.h, 835
- do_stats
 - dyn-rec.c, 350
 - execute.c, 375
 - execute.h, 267
 - sim-config.h, 814
- done_memory_table
 - abstract.c, 213
 - abstract.h, 232
- DOWNLOAD_MICROCODE
 - atacmd.h, 624
- dsize
 - COFF_AOUTHDR, 40
- dstats
 - stats.c, 282
- dstats_entry, 88
 - cnt_dynamic, 88
 - depend, 88
 - insn1, 88
 - insn2, 88
- DSTATS_LEN
 - stats.c, 281
- DT_DEBUG
 - elf.h, 257
- DT_FINI
 - elf.h, 257
- DT_HASH
 - elf.h, 257
- DT_HIPROC
 - elf.h, 257
- DT_INIT
 - elf.h, 257
- DT_JMPREL
 - elf.h, 257
- DT_LOPROC
 - elf.h, 257
- DT_NEEDED
 - elf.h, 257
- DT_NULL
 - elf.h, 257
- DT_PLTGOT
 - elf.h, 257
- DT_PLTREL
 - elf.h, 257
- DT_PLTRELSZ
 - elf.h, 257
- DT_REL
 - elf.h, 257
- DT_RELA
 - elf.h, 257
- DT_RELAENT
 - elf.h, 257
- DT_RELASZ
 - elf.h, 257
- DT_RELENT
 - elf.h, 257

- DT_RELSZ
 - elf.h, 257
- DT_RPATH
 - elf.h, 257
- DT_SONAME
 - elf.h, 257
- DT_STRSZ
 - elf.h, 257
- DT_STARTAB
 - elf.h, 257
- DT_SYMBOLIC
 - elf.h, 257
- DT_SYMENT
 - elf.h, 257
- DT_SYMTAB
 - elf.h, 257
- DT_TEXTREL
 - elf.h, 257
- dtlb_status
 - dmmu.c, 574
- dtr0
 - ata_host, 26
- dtr1
 - ata_host, 26
- du_reset
 - debug-unit.c, 521
 - debug-unit.h, 527
- dump_exe_log
 - execute.c, 370
 - execute.h, 263
- dump_memory
 - abstract.c, 213
 - abstract.h, 232
- dump_spr
 - spr-dump.c, 309
 - spr-dump.h, 320
 - sprs.h, 323
- dumphex
 - dumpverilog.c, 824
 - dumpverilog.h, 826
- dumpreg
 - execute.c, 370
 - execute.h, 264
- dumpverilog
 - dumpverilog.c, 824
 - dumpverilog.h, 826
- dumpverilog.c
 - DISWIDTH, 823
 - dumphex, 824
 - dumpverilog, 824
 - DW, 823
 - DWQ, 823
 - LABELEND_CHAR, 823
 - OR1K_MEM_VERILOG_FOOTER, 823
 - OR1K_MEM_VERILOG_HEADER, 823
- dumpverilog.h
 - dumphex, 826
 - dumpverilog, 826
- dup_func
 - bb.c, 444
 - cuc.h, 473
- DW
 - dumpverilog.c, 823
- DWQ
 - dumpverilog.c, 823
- dyn-rec.c
 - __op_param1, 350
 - __op_param2, 350
 - __op_param3, 350
 - add_to_op_params, 335
 - add_to_opq, 335
 - DEF_1T_OP, 335
 - DEF_2T_OP, 335
 - DEF_2T_OP_NEQ, 335
 - DEF_3T_OP, 335
 - DEF_3T_OP_NEQ, 335
 - DEF_GPR_OP, 335
 - dirtyfy_page, 335
 - do_stats, 350
 - dyn_checkwrite, 336
 - dyn_main, 336
 - dyn_ret_stack_prot, 336
 - dyn_sigsegv_debug, 337
 - enough_host_page, 337
 - eval_insn_ops, 337
 - find_jump_loc, 337
 - find_t, 337
 - gen_j_imm, 338
 - gen_j_reg, 338
 - gen_l_add, 338
 - gen_l_addc, 338
 - gen_l_and, 338
 - gen_l_bf, 338
 - gen_l_bnf, 338
 - gen_l_cmov, 339
 - gen_l_cust1, 340
 - gen_l_cust2, 340
 - gen_l_cust3, 340
 - gen_l_cust4, 340
 - gen_l_cust5, 340
 - gen_l_cust6, 340
 - gen_l_cust7, 340
 - gen_l_cust8, 340
 - gen_l_div, 340
 - gen_l_divu, 340
 - gen_l_extbs, 340
 - gen_l_extbz, 340
 - gen_l_exths, 340

- gen_l_exthz, 340
- gen_l_extws, 340
- gen_l_extwz, 340
- gen_l_ffl, 340
- gen_l_invalid, 340
- gen_l_j, 340
- gen_l_jal, 340
- gen_l_jalr, 341
- gen_l_jr, 341
- gen_l_lbs, 341
- gen_l_lbz, 342
- gen_l_lhs, 342
- gen_l_lhz, 342
- gen_l_lws, 342
- gen_l_lwz, 342
- gen_l_mac, 342
- gen_l_macrc, 343
- gen_l_mfspr, 343
- gen_l_movhi, 343
- gen_l_msb, 343
- gen_l_mtspr, 343
- gen_l_mul, 343
- gen_l_mulu, 343
- gen_l_nop, 343
- gen_l_or, 343
- gen_l_rfe, 343
- gen_l_sb, 343
- gen_l_sfeq, 343
- gen_l_sfges, 344
- gen_l_sfgeu, 344
- gen_l_sfgts, 344
- gen_l_sfgtu, 344
- gen_l_sfles, 344
- gen_l_sfleu, 344
- gen_l_sflts, 344
- gen_l_sfltu, 344
- gen_l_sfne, 344
- gen_l_sh, 344
- gen_l_sll, 344
- gen_l_sra, 344
- gen_l_srl, 344
- gen_l_sub, 344
- gen_l_sw, 344
- gen_l_sys, 344
- gen_l_trap, 345
- gen_l_xor, 345
- gen_lf_add_s, 345
- gen_lf_div_s, 345
- gen_lf_ftoi_s, 345
- gen_lf_itof_s, 345
- gen_lf_madd_s, 345
- gen_lf_mul_s, 345
- gen_lf_rem_s, 346
- gen_lf_sfeq_s, 346
- gen_lf_sfge_s, 346
- gen_lf_sfgt_s, 346
- gen_lf_sfle_s, 346
- gen_lf_sflt_s, 346
- gen_lf_sfne_s, 347
- gen_lf_sub_s, 347
- gen_op_mark_loc, 347
- generic_gen_op, 335
- imm_gen_op, 335
- immu_retranslate, 347
- init_dyn_recomp, 348
- new_dp, 348
- OPS_ENLARGE_BY, 335
- RECED_PAGE_ENLARGE_BY, 335
- recheck_immu, 348
- recompile_delay_insn, 348
- recompile_insn, 349
- recompile_page, 349
- run_sched_out_of_line, 349
- set_pc_delay_gpr, 350
- ship_gprs_out_t, 350
- ship_t_out, 350
- sigsegv_addr, 351
- sigsegv_state, 351
- T_NONE, 335
- TFLAG_DST, 335
- TFLAG_SAVED, 335
- TFLAG_SOURCED, 335
- TFLAG_SRC, 335
- dyn-rec.h
 - add_to_op_params, 353
 - add_to_opq, 353
 - dyn_checkwrite, 353
 - dyn_main, 353
 - enough_host_page, 354
 - enter_dyn_code, 354
 - gen_code_ent, 353
 - glue, 353
 - IMMU_GOT_DISABLED, 353
 - IMMU_GOT_ENABLED, 353
 - init_dyn_recomp, 354
 - new_dp, 354
 - rec_stack_base, 356
 - recheck_immu, 355
 - recompile_page, 355
 - run_sched_out_of_line, 355
 - xglue, 353
- dyn32-defs.h
 - l_none, 357
 - PARAMS, 357
- dyn_checkwrite
 - dyn-rec.c, 336
 - dyn-rec.h, 353
- dyn_main

- dyn-rec.c, 336
- dyn-rec.h, 353
- dyn_page, 89
- delayr, 89
- dirty, 89
- host_len, 89
- host_page, 89
- insn_indexes, 89
- insns, 89
- locs, 89
- or_page, 89
- ts_bound, 89
- dyn_ret_stack_prot
 - dyn-rec.c, 336
- dyn_sigsegv_debug
 - dyn-rec.c, 337
- dynamic, 90
 - d_ptr, 90
 - d_tag, 90
 - d_un, 90
 - d_val, 90
- dyngen-elf.c
 - bffs, 359
 - elf_close_obj, 358
 - elf_find_func, 358
 - elf_get_func_len, 358
 - elf_get_func_name, 359
 - elf_get_func_reloc, 359
 - elf_get_func_start, 359
 - elf_get_sym_name, 359
 - elf_open_obj, 359
- dyngen-i386.c
 - archfs, 360
 - i386_gen_func_reloc, 360
 - i386_gen_reloc, 360
 - i386_get_real_func_len, 360
 - RET_OPCODE, 360
- dyngen.c
 - c_file_head, 363
 - c_rel_file_head, 363
 - c_rel_file_tail, 363
 - c_sw_file_head, 363
 - c_sw_file_tail, 364
 - gen_code_proto, 364
 - gen_func_proto, 363
 - main, 363
 - MAX_PARAMS, 363
 - OP_FUNC_PARAM_PREFIX, 363
 - OP_FUNC_PREFIX, 363
- dyngen.h
 - archfs, 365
 - bffs, 365
- e
 - COFF_syment, 50
 - E_DIMNUM
 - coff.h, 249
 - e_ehsize
 - elf32_hdr, 92
 - elf64_hdr, 101
 - e_entry
 - elf32_hdr, 92
 - elf64_hdr, 101
 - E_FILNMLEN
 - coff.h, 249
 - e_flags
 - elf32_hdr, 92
 - elf64_hdr, 101
 - e_hdr
 - elf_obj, 109
 - e_ident
 - elf32_hdr, 92
 - elf64_hdr, 101
 - e_machine
 - elf32_hdr, 92
 - elf64_hdr, 101
 - e_name
 - COFF_syment, 50
 - e_numaux
 - COFF_syment, 50
 - e_offset
 - COFF_syment, 50
 - e_phentsize
 - elf32_hdr, 92
 - elf64_hdr, 101
 - e_phnum
 - elf32_hdr, 92
 - elf64_hdr, 101
 - e_phoff
 - elf32_hdr, 92
 - elf64_hdr, 101
 - e_rel_num
 - elf_obj, 109
 - e_rel_sec
 - elf_obj, 109
 - e_rel_sym
 - elf_obj, 109
 - e_rela_num
 - elf_obj, 109
 - e_rela_sec
 - elf_obj, 109
 - e_rela_sym
 - elf_obj, 109
 - e_relas
 - elf_obj, 109
 - e_rels
 - elf_obj, 109
 - e_sclass

- COFF_syment, 50
- e_scnnum
 - COFF_syment, 50
- e_sections
 - elf_obj, 109
- e_shdrs
 - elf_obj, 109
- e_shentsize
 - elf32_hdr, 92
 - elf64_hdr, 101
- e_shnum
 - elf32_hdr, 92
 - elf64_hdr, 101
- e_shoff
 - elf32_hdr, 92
 - elf64_hdr, 101
- e_shstrndx
 - elf32_hdr, 92
 - elf64_hdr, 101
- e_sym_num
 - elf_obj, 109
- e_sym_str_tab
 - elf_obj, 109
- E_SYMNMLEN
 - coff.h, 249
- e_syms
 - elf_obj, 109
- e_type
 - COFF_syment, 50
 - elf32_hdr, 92
 - elf64_hdr, 101
- e_value
 - COFF_syment, 50
- e_version
 - elf32_hdr, 92
 - elf64_hdr, 101
- e_zeroes
 - COFF_syment, 50
- edge_trigger
 - config::pic, 57
- EF
 - or32.c, 432
- EFI
 - or32.c, 432
- EFN
 - or32.c, 432
- EI_CLASS
 - elf.h, 257
- EI_DATA
 - elf.h, 257
- EI_MAG0
 - elf.h, 257
- EI_MAG1
 - elf.h, 257
- EI_MAG2
 - elf.h, 257
- EI_MAG3
 - elf.h, 257
- EI_NIDENT
 - elf.h, 257
- EI_PAD
 - elf.h, 257
- EI_VERSION
 - elf.h, 257
- elf.h
 - _DYNAMIC, 260
 - AT_BASE, 257
 - AT_EGID, 257
 - AT_ENTRY, 257
 - AT_EUID, 257
 - AT_EXECPD, 257
 - AT_FLAGS, 257
 - AT_GID, 257
 - AT_IGNORE, 257
 - AT_NOTELF, 257
 - AT_NULL, 257
 - AT_PAGESZ, 257
 - AT_PHDR, 257
 - AT_PHEXT, 257
 - AT_PHNUM, 257
 - AT_UID, 257
 - DT_DEBUG, 257
 - DT_FINI, 257
 - DT_HASH, 257
 - DT_HIPOC, 257
 - DT_INIT, 257
 - DT_JMPREL, 257
 - DT_LOPROC, 257
 - DT_NEEDED, 257
 - DT_NULL, 257
 - DT_PLTGOT, 257
 - DT_PLTREL, 257
 - DT_PLTRELSZ, 257
 - DT_REL, 257
 - DT_RELA, 257
 - DT_RELAENT, 257
 - DT_RELASZ, 257
 - DT_RELENT, 257
 - DT_RELSZ, 257
 - DT_RPATH, 257
 - DT_SONAME, 257
 - DT_STRSZ, 257
 - DT_STRTAB, 257
 - DT_SYMBOLIC, 257
 - DT_SYMENT, 257
 - DT_SYMTAB, 257
 - DT_TEXTREL, 257
 - EI_CLASS, 257

EL_DATA, [257](#)
EL_MAG0, [257](#)
EL_MAG1, [257](#)
EL_MAG2, [257](#)
EL_MAG3, [257](#)
EL_NIDENT, [257](#)
EL_PAD, [257](#)
EL_VERSION, [257](#)
Elf32_Addr, [260](#)
Elf32_Dyn, [260](#)
Elf32_Ehdr, [260](#)
Elf32_Half, [260](#)
Elf32_Nhdr, [260](#)
Elf32_Off, [260](#)
Elf32_Phdr, [260](#)
ELF32_R_SYM, [257](#)
ELF32_R_TYPE, [257](#)
Elf32_Rel, [260](#)
Elf32_Rela, [260](#)
Elf32_Shdr, [260](#)
ELF32_ST_BIND, [257](#)
ELF32_ST_TYPE, [257](#)
Elf32_Sword, [260](#)
Elf32_Sym, [260](#)
Elf32_Word, [260](#)
Elf64_Ehdr, [260](#)
Elf64_Nhdr, [260](#)
Elf64_Phdr, [260](#)
Elf64_Rel, [260](#)
Elf64_Rela, [260](#)
Elf64_Shdr, [260](#)
Elf64_Sym, [260](#)
ELF_LONG_H, [257](#)
elf_note, [258](#)
elf_phdr, [258](#)
ELF_SHORT_H, [258](#)
ELFCLASS32, [258](#)
ELFCLASS64, [260](#)
ELFCLASSNONE, [260](#)
ELFCLASSNUM, [260](#)
ELFDATA2LSB, [260](#)
ELFDATA2MSB, [260](#)
ELFDATANONE, [260](#)
elfhdr, [260](#)
ELFMAG, [260](#)
ELFMAG0, [260](#)
ELFMAG1, [260](#)
ELFMAG2, [260](#)
ELFMAG3, [260](#)
EM_386, [260](#)
EM_486, [260](#)
EM_68K, [260](#)
EM_860, [260](#)
EM_88K, [260](#)
EM_ALPHA, [260](#)
EM_M32, [260](#)
EM_MIPS, [260](#)
EM_MIPS_RS4_BE, [260](#)
EM_NONE, [260](#)
EM_PARISC, [260](#)
EM_PPC, [260](#)
EM_SPARC, [260](#)
EM_SPARC32PLUS, [260](#)
EM_SPARC64, [260](#)
ET_CORE, [260](#)
ET_DYN, [260](#)
ET_EXEC, [260](#)
ET_HIPROC, [260](#)
ET_LOPROC, [260](#)
ET_NONE, [260](#)
ET_REL, [260](#)
EV_CURRENT, [260](#)
EV_NONE, [260](#)
EV_NUM, [260](#)
NT_PRFPREG, [260](#)
NT_PRPSINFO, [260](#)
NT_PRSTATUS, [260](#)
NT_TASKSTRUCT, [260](#)
PF_R, [260](#)
PF_W, [260](#)
PF_X, [260](#)
PT_DYNAMIC, [260](#)
PT_HIPROC, [260](#)
PT_INTERP, [260](#)
PT_LOAD, [260](#)
PT_LOPROC, [260](#)
PT_NOTE, [260](#)
PT_NULL, [260](#)
PT_PHDR, [260](#)
PT_SHLIB, [260](#)
R_386_32, [260](#)
R_386_COPY, [260](#)
R_386_GLOB_DAT, [260](#)
R_386_GOT32, [260](#)
R_386_GOTOFF, [260](#)
R_386_GOTPC, [260](#)
R_386_JMP_SLOT, [260](#)
R_386_NONE, [260](#)
R_386_NUM, [260](#)
R_386_PC32, [260](#)
R_386_PLT32, [260](#)
R_386_RELATIVE, [260](#)
R_68K_16, [260](#)
R_68K_32, [260](#)
R_68K_8, [260](#)
R_68K_COPY, [260](#)
R_68K_GLOB_DAT, [260](#)
R_68K_GOT16, [260](#)

- R_68K_GOT16O, 260
- R_68K_GOT32, 260
- R_68K_GOT32O, 260
- R_68K_GOT8, 260
- R_68K_GOT8O, 260
- R_68K_JMP_SLOT, 260
- R_68K_NONE, 260
- R_68K_PC16, 260
- R_68K_PC32, 260
- R_68K_PC8, 260
- R_68K_PLT16, 260
- R_68K_PLT16O, 260
- R_68K_PLT32, 260
- R_68K_PLT32O, 260
- R_68K_PLT8, 260
- R_68K_PLT8O, 260
- R_68K_RELATIVE, 260
- SELMAG, 260
- SHF_ALLOC, 260
- SHF_EXECINSTR, 260
- SHF_MASKPROC, 260
- SHF_WRITE, 260
- SHN_ABS, 260
- SHN_COMMON, 260
- SHN_HIPROC, 260
- SHN_HIRESERVE, 260
- SHN_LOPROC, 260
- SHN_LORESERVE, 260
- SHN_UNDEF, 260
- SHT_DYNAMIC, 260
- SHT_DYNSYM, 260
- SHT_HASH, 260
- SHT_HIPROC, 260
- SHT_HIUSER, 260
- SHT_LOPROC, 260
- SHT_LOUSER, 260
- SHT_NOBITS, 260
- SHT_NOTE, 260
- SHT_NULL, 260
- SHT_NUM, 260
- SHT_PROGBITS, 260
- SHT_REL, 260
- SHT_RELA, 260
- SHT_SHLIB, 260
- SHT_STRTAB, 260
- SHT_SYMTAB, 260
- STB_GLOBAL, 260
- STB_LOCAL, 260
- STB_WEAK, 260
- STT_FILE, 260
- STT_FUNC, 260
- STT_NOTYPE, 260
- STT_OBJECT, 260
- STT_SECTION, 260
- Elf32_Addr
 - elf.h, 260
- Elf32_Dyn
 - elf.h, 260
- Elf32_Ehdr
 - elf.h, 260
- Elf32_Half
 - elf.h, 260
- elf32_hdr, 91
 - e_ehsize, 92
 - e_entry, 92
 - e_flags, 92
 - e_ident, 92
 - e_machine, 92
 - e_phentsize, 92
 - e_phnum, 92
 - e_phoff, 92
 - e_shentsize, 92
 - e_shnum, 92
 - e_shoff, 92
 - e_shstrndx, 92
 - e_type, 92
 - e_version, 92
- Elf32_Nhdr
 - elf.h, 260
- elf32_note, 93
 - n_descsz, 93
 - n_namesz, 93
 - n_type, 93
- Elf32_Off
 - elf.h, 260
- Elf32_Phdr
 - elf.h, 260
- elf32_phdr, 94
 - p_align, 94
 - p_filesz, 94
 - p_flags, 94
 - p_memsz, 94
 - p_offset, 94
 - p_paddr, 94
 - p_type, 94
 - p_vaddr, 94
- ELF32_R_SYM
 - elf.h, 257
- ELF32_R_TYPE
 - elf.h, 257
- Elf32_Rel
 - elf.h, 260
- elf32_rel, 95
 - r_info, 95
 - r_offset, 95
- Elf32_Rela
 - elf.h, 260
- elf32_rela, 96

- r_addend, 96
- r_info, 96
- r_offset, 96
- Elf32_Shdr
 - elf.h, 260
- elf32_shdr, 97
 - sh_addr, 97
 - sh_addralign, 97
 - sh_entsize, 97
 - sh_flags, 97
 - sh_info, 97
 - sh_link, 97
 - sh_name, 97
 - sh_offset, 97
 - sh_size, 97
 - sh_type, 97
- ELF32_ST_BIND
 - elf.h, 257
- ELF32_ST_TYPE
 - elf.h, 257
- Elf32_Sword
 - elf.h, 260
- Elf32_Sym
 - elf.h, 260
- elf32_sym, 98
 - st_info, 98
 - st_name, 98
 - st_other, 98
 - st_shndx, 98
 - st_size, 98
 - st_value, 98
- Elf32_Word
 - elf.h, 260
- Elf64_Dyn, 99
 - d_ptr, 99
 - d_tag, 99
 - d_un, 99
 - d_val, 99
- Elf64_Ehdr
 - elf.h, 260
- elf64_hdr, 100
 - e_ehsize, 101
 - e_entry, 101
 - e_flags, 101
 - e_ident, 101
 - e_machine, 101
 - e_phentsize, 101
 - e_phnum, 101
 - e_phoff, 101
 - e_shentsize, 101
 - e_shnum, 101
 - e_shoff, 101
 - e_shstrndx, 101
 - e_type, 101
 - e_version, 101
- Elf64_Nhdr
 - elf.h, 260
- elf64_note, 102
 - n_descsz, 102
 - n_namesz, 102
 - n_type, 102
- Elf64_Phdr
 - elf.h, 260
- elf64_phdr, 103
 - p_align, 103
 - p_filesz, 103
 - p_flags, 103
 - p_memsz, 103
 - p_offset, 103
 - p_paddr, 103
 - p_type, 103
 - p_vaddr, 103
- Elf64_Rel
 - elf.h, 260
- elf64_rel, 104
 - r_info, 104
 - r_offset, 104
- Elf64_Rela
 - elf.h, 260
- elf64_rela, 105
 - r_addend, 105
 - r_info, 105
 - r_offset, 105
- Elf64_Shdr
 - elf.h, 260
- elf64_shdr, 106
 - sh_addr, 106
 - sh_addralign, 106
 - sh_entsize, 106
 - sh_flags, 106
 - sh_info, 106
 - sh_link, 106
 - sh_name, 106
 - sh_offset, 106
 - sh_size, 106
 - sh_type, 106
- Elf64_Sym
 - elf.h, 260
- elf64_sym, 107
 - st_info, 107
 - st_name, 107
 - st_other, 107
 - st_shndx, 107
 - st_size, 107
 - st_value, 107
- elf_close_obj
 - dyngen-elf.c, 358
- elf_find_func

- dyngen-elf.c, 358
- elf_get_func_len
 - dyngen-elf.c, 358
- elf_get_func_name
 - dyngen-elf.c, 359
- elf_get_func_reloc
 - dyngen-elf.c, 359
- elf_get_func_start
 - dyngen-elf.c, 359
- elf_get_sym_name
 - dyngen-elf.c, 359
- ELF_LONG_H
 - elf.h, 257
- elf_note
 - elf.h, 258
- elf_obj, 108
 - e_hdr, 109
 - e_rel_num, 109
 - e_rel_sec, 109
 - e_rel_sym, 109
 - e_rela_num, 109
 - e_rela_sec, 109
 - e_rela_sym, 109
 - e_relas, 109
 - e_rels, 109
 - e_sections, 109
 - e_shdrs, 109
 - e_sym_num, 109
 - e_sym_str_tab, 109
 - e_syms, 109
- elf_open_obj
 - dyngen-elf.c, 359
- elf_phdr
 - elf.h, 258
- ELF_SHORT_H
 - elf.h, 258
- ELFCLASS32
 - elf.h, 258
- ELFCLASS64
 - elf.h, 260
- ELFCLASSNONE
 - elf.h, 260
- ELFCLASSNUM
 - elf.h, 260
- ELFDATA2LSB
 - elf.h, 260
- ELFDATA2MSB
 - elf.h, 260
- ELFDATANONE
 - elf.h, 260
- elfhdr
 - elf.h, 260
- ELFMAG
 - elf.h, 260
- ELFMAG0
 - elf.h, 260
- ELFMAG1
 - elf.h, 260
- ELFMAG2
 - elf.h, 260
- ELFMAG3
 - elf.h, 260
- EM_386
 - elf.h, 260
- EM_486
 - elf.h, 260
- EM_68K
 - elf.h, 260
- EM_860
 - elf.h, 260
- EM_88K
 - elf.h, 260
- EM_ALPHA
 - elf.h, 260
- EM_M32
 - elf.h, 260
- EM_MIPS
 - elf.h, 260
- EM_MIPS_RS4_BE
 - elf.h, 260
- EM_NONE
 - elf.h, 260
- EM_PARISC
 - elf.h, 260
- EM_PPC
 - elf.h, 260
- EM_SPARC
 - elf.h, 260
- EM_SPARC32PLUS
 - elf.h, 260
- EM_SPARC64
 - elf.h, 260
- ENABLE_ADVANCED_POWER_-
MANAGEMENT
 - atacmd.h, 624
- enable_bursts
 - config, 55
- ENABLE_MEDIA_STATUS_NOTIFICATION
 - atacmd.h, 624
- ENABLE_POWERUP_IN_STANDBY_-
FEATURE_SET
 - atacmd.h, 624
- ENABLE_READ_LOOKAHEAD_FEATURE
 - atacmd.h, 624
- ENABLE_RELEASE_INTERRUPT
 - atacmd.h, 624
- ENABLE_REVERTING_TO_POWERON_-
DEFAULTS

- atacmd.h, 624
- ENABLE_SERVICE_INTERRUPT
 - atacmd.h, 624
- ENABLE_WRITE_CACHE
 - atacmd.h, 624
- enabled
 - ata_host, 26
 - config, 55
 - config::pic, 57
 - dev_16450, 74
 - dev_generic, 77
 - dma_controller, 84
 - dmmu, 86
 - eth_device, 113
 - fb_state, 118
 - gpio_device, 124
 - ic, 127
 - immu, 129
 - kbd_state, 144
 - mc, 147
 - runtime, 165
 - vga_state, 180
- end_addr
 - _cuc_func, 16
 - mprofiler.c, 589
- end_cycles
 - runtime, 165
- end_t_func
 - ata_device, 22
- enough_host_page
 - dyn-rec.c, 337
 - dyn-rec.h, 354
- enter_dyn_code
 - dyn-rec.h, 354
 - op.c, 424
- entry
 - COFF_AOUTHDR, 40
- entrysize
 - dmmu, 86
 - immu, 129
- ERR
 - debug.h, 819
- ERR_CRC
 - gdb.h, 529
- ERR_MEM
 - gdb.h, 529
- ERR_NONE
 - gdb.h, 529
- ERR_
 - debug.h, 820
- ERR_ON
 - debug.h, 820
- error
 - ata_device, 22
 - eth_device, 113
- ET_CORE
 - elf.h, 260
- ET_DYN
 - elf.h, 260
- ET_EXEC
 - elf.h, 260
- ET_HIPROC
 - elf.h, 260
- ET_LOPROC
 - elf.h, 260
- ET_NONE
 - elf.h, 260
- ET_REL
 - elf.h, 260
- eth.c
 - ETH_NUM_VAPI_IDS, 703
 - ETH_VAPI_CTRL, 702
 - ETH_VAPI_DATA, 702
 - ETH_ADDR_SPACE, 702
 - ETH_ALEN, 702
 - eth_baseaddr, 703
 - ETH_BD_BASE, 702
 - ETH_BD_COUNT, 702
 - ETH_BD_SPACE, 702
 - ETH_CMODER_PASSALL_OFFSET, 702
 - ETH_CMODER_RXFLOW_OFFSET, 702
 - ETH_CMODER_TXFLOW_OFFSET, 702
 - ETH_COLLCONF, 702
 - ETH_COLLCONF_COLLVALID_OFFSET, 702
 - ETH_COLLCONF_COLLVALID_WIDTH, 702
 - ETH_COLLCONF_MAXRET_OFFSET, 702
 - ETH_COLLCONF_MAXRET_WIDTH, 702
 - eth_controller_rx_clock, 703
 - eth_controller_tx_clock, 703
 - ETH_CTRLMODER, 702
 - eth_dma, 704
 - ETH_DMA_RX_TX, 702
 - eth_enabled, 704
 - ETH_HASH0, 702
 - ETH_HASH1, 702
 - ETH_INT_MASK, 702
 - ETH_INT_MASK_BUSY_M_OFFSET, 702
 - ETH_INT_MASK_RXB_M_OFFSET, 702
 - ETH_INT_MASK_RXC_M_OFFSET, 702
 - ETH_INT_MASK_RXE_M_OFFSET, 702
 - ETH_INT_MASK_TXB_M_OFFSET, 702
 - ETH_INT_MASK_TXC_M_OFFSET, 702
 - ETH_INT_MASK_TXE_M_OFFSET, 702
 - ETH_INT_SOURCE, 702
 - ETH_INT_SOURCE_BUSY_OFFSET, 702
 - ETH_INT_SOURCE_RXB_OFFSET, 702

- ETH_INT_SOURCE_RXC_OFFSET, 702
- ETH_INT_SOURCE_RXE_OFFSET, 702
- ETH_INT_SOURCE_TXB_OFFSET, 702
- ETH_INT_SOURCE_TXC_OFFSET, 702
- ETH_INT_SOURCE_TXE_OFFSET, 702
- ETH_IPGR1, 702
- ETH_IPGR2, 702
- ETH_IPGT, 702
- eth_irq, 704
- ETH_MAC_ADDR0, 702
- ETH_MAC_ADDR1, 702
- ETH_MAXPL, 702
- ETH_MIIADDR_FIAD_OFFSET, 702
- ETH_MIIADDR_FIAD_WIDTH, 702
- ETH_MIIADDR_RGAD_OFFSET, 702
- ETH_MIIADDR_RGAD_WIDTH, 702
- ETH_MIIADDRESS, 702
- ETH_MIICOMM_RSTAT_OFFSET, 702
- ETH_MIICOMM_SCANS_OFFSET, 702
- ETH_MIICOMM_WCDATA_OFFSET, 702
- ETH_MIICOMMAND, 702
- ETH_MIIMODER, 702
- ETH_MIIMODER_CLKDIV_OFFSET, 702
- ETH_MIIMODER_CLKDIV_WIDTH, 702
- ETH_MIIMODER_MRST_OFFSET, 702
- ETH_MIIMODER_NOPRE_OFFSET, 702
- ETH_MIIRX_DATA, 702
- ETH_MIISTAT_BUSY_OFFSET, 702
- ETH_MIISTAT_FAIL_OFFSET, 702
- ETH_MIISTAT_NVALID_OFFSET, 702
- ETH_MIISTATUS, 702
- ETH_MIITX_DATA, 702
- ETH_MODER, 702
- ETH_MODER_BRO_OFFSET, 702
- ETH_MODER_CRCEN_OFFSET, 702
- ETH_MODER_DLYCRCEN_OFFSET, 702
- ETH_MODER_DMAEN_OFFSET, 702
- ETH_MODER_EXDFREN_OFFSET, 702
- ETH_MODER_FULLD_OFFSET, 702
- ETH_MODER_HUGEN_OFFSET, 702
- ETH_MODER_IAM_OFFSET, 702
- ETH_MODER_IFG_OFFSET, 702
- ETH_MODER_LOOPBCK_OFFSET, 702
- ETH_MODER_NOBCKOF_OFFSET, 702
- ETH_MODER_NOPRE_OFFSET, 702
- ETH_MODER_PAD_OFFSET, 702
- ETH_MODER_PRO_OFFSET, 702
- ETH_MODER_RECSMALL_OFFSET, 702
- ETH_MODER_RST_OFFSET, 702
- ETH_MODER_RXEN_OFFSET, 702
- ETH_MODER_TXEN_OFFSET, 702
- ETH_PACKETLEN, 702
- ETH_PACKETLEN_MAXFL_OFFSET, 702
- ETH_PACKETLEN_MAXFL_WIDTH, 702
- ETH_PACKETLEN_MINFL_OFFSET, 702
- ETH_PACKETLEN_MINFL_WIDTH, 702
- eth_read32, 704
- eth_read_rx_file, 704
- eth_reset, 704
- ETH_RTX_FILE, 702
- ETH_RTX_SOCKET, 702
- eth_rtx_type, 704
- ETH_RTX_VAPI, 702
- ETH_RX_BD_COLLISION_OFFSET, 702
- ETH_RX_BD_CRC_OFFSET, 702
- ETH_RX_BD_DRIBBLE_OFFSET, 702
- ETH_RX_BD_INVALID_OFFSET, 702
- ETH_RX_BD_IRQ_OFFSET, 702
- ETH_RX_BD_LENGTH_OFFSET, 702
- ETH_RX_BD_LENGTH_WIDTH, 702
- ETH_RX_BD_MISS_OFFSET, 702
- ETH_RX_BD_READY_OFFSET, 702
- ETH_RX_BD_TOOBIG_OFFSET, 702
- ETH_RX_BD_TOOSHORT_OFFSET, 702
- ETH_RX_BD_UVERRUN_OFFSET, 702
- ETH_RX_BD_WRAP_OFFSET, 702
- eth_rx_channel, 705
- eth_rx_next_packet, 705
- eth_rxfile, 705
- ETH_RXSTATE_IDLE, 702
- ETH_RXSTATE_RECV, 702
- ETH_RXSTATE_WAIT4BD, 702
- ETH_RXSTATE_WRITEFIFO, 702
- eth_sec_end, 705
- eth_sec_start, 708
- eth_skip_rx_file, 708
- eth_sockif, 708
- eth_status, 708
- ETH_TX_BD_COLLISION_OFFSET, 702
- ETH_TX_BD_CRC_OFFSET, 702
- ETH_TX_BD_DEFER_OFFSET, 702
- ETH_TX_BD_IRQ_OFFSET, 702
- ETH_TX_BD_LAST_OFFSET, 702
- ETH_TX_BD_LENGTH_OFFSET, 702
- ETH_TX_BD_LENGTH_WIDTH, 702
- ETH_TX_BD_NO_CARRIER_OFFSET, 702
- ETH_TX_BD_NUM, 702
- ETH_TX_BD_PAD_OFFSET, 702
- ETH_TX_BD_PAUSE_OFFSET, 702
- ETH_TX_BD_READY_OFFSET, 702
- ETH_TX_BD_RETRANSMIT_OFFSET, 702
- ETH_TX_BD_RETRY_OFFSET, 702
- ETH_TX_BD_RETRY_WIDTH, 702
- ETH_TX_BD_UNDERRUN_OFFSET, 702
- ETH_TX_BD_WRAP_OFFSET, 702
- eth_tx_channel, 708
- eth_txfile, 708
- ETH_TXSTATE_IDLE, 702

- ETH_TXSTATE_READFIFO, 702
- ETH_TXSTATE_TRANSMIT, 702
- ETH_TXSTATE_WAIT4BD, 702
- eth_vapi_id, 708
- eth_vapi_read, 709
- eth_write32, 709
- eth_write_tx_bd_num, 709
- ETHER_ADDR_LEN, 702
- ETHER_CRC_LEN, 702
- ETHER_HDR_LEN, 702
- ETHER_IS_VALID_LEN, 702
- ETHER_MAX_LEN, 702
- ETHER_MIN_LEN, 702
- ETHER_TYPE_LEN, 702
- ETHERMIN, 702
- ETHERMTU, 702
- ETHERTYPE_ARP, 702
- ETHERTYPE_IP, 702
- ETHERTYPE_NTRAILER, 702
- ETHERTYPE_PUP, 702
- ETHERTYPE_REVARP, 702
- ETHERTYPE_TRAIL, 702
- reg_ethernet_sec, 709
- eth.h
 - reg_ethernet_sec, 711
- ETH_NUM_VAPI_IDS
 - eth.c, 703
- ETH_VAPI_CTRL
 - eth.c, 702
- ETH_VAPI_DATA
 - eth.c, 702
- ETH_ADDR_SPACE
 - eth.c, 702
- ETH_ALEN
 - eth.c, 702
- eth_baseaddr
 - eth.c, 703
- ETH_BD_BASE
 - eth.c, 702
- ETH_BD_COUNT
 - eth.c, 702
- ETH_BD_SPACE
 - eth.c, 702
- ETH_CMODER_PASSALL_OFFSET
 - eth.c, 702
- ETH_CMODER_RXFLOW_OFFSET
 - eth.c, 702
- ETH_CMODER_TXFLOW_OFFSET
 - eth.c, 702
- ETH_COLLCONF
 - eth.c, 702
- ETH_COLLCONF_COLLVALID_OFFSET
 - eth.c, 702
- ETH_COLLCONF_COLLVALID_WIDTH
 - eth.c, 702
- ETH_COLLCONF_MAXRET_OFFSET
 - eth.c, 702
- ETH_COLLCONF_MAXRET_WIDTH
 - eth.c, 702
- eth_controller_rx_clock
 - eth.c, 703
- eth_controller_tx_clock
 - eth.c, 703
- ETH_CTRLMODER
 - eth.c, 702
- eth_device, 110
 - add_crc, 113
 - base_vapi_id, 113
 - baseaddr, 113
 - bd, 113
 - bd_addr, 113
 - bd_index, 113
 - bd_ram, 113
 - bytes_left, 113
 - bytes_read, 113
 - bytes_sent, 113
 - collconf, 113
 - controlmoder, 113
 - crc_dly, 113
 - crc_value, 113
 - dma, 113
 - enabled, 113
 - error, 113
 - fd, 113
 - hash0, 113
 - hash1, 113
 - ifr, 113
 - int_mask, 113
 - int_source, 113
 - ipgr1, 113
 - ipgr2, 113
 - ipgt, 113
 - lo_buff, 113
 - loopback_offset, 113
 - mac_address, 113
 - mac_int, 113
 - maximum_length, 113
 - miiaddress, 113
 - miicommand, 113
 - miimoder, 113
 - miirx_data, 113
 - miistatus, 113
 - miitx_data, 113
 - minimum_length, 113
 - moder, 113
 - offset, 113
 - packet_length, 113
 - packetlen, 113

- regs, 113
- rfds, 113
- rtx_sock, 113
- rtx_type, 113
- rx, 113
- rx_buff, 113
- rx_channel, 113
- rxfd, 113
- rxfile, 113
- sockif, 113
- state, 113
- tx, 113
- tx_bd_num, 113
- tx_buff, 113
- tx_channel, 113
- txfd, 113
- txfile, 113
- waiting_for_dma, 113
- wfds, 113
- working, 113
- eth_dma
 - eth.c, 704
- ETH_DMA_RX_TX
 - eth.c, 702
- eth_enabled
 - eth.c, 704
- ETH_HASH0
 - eth.c, 702
- ETH_HASH1
 - eth.c, 702
- ETH_INT_MASK
 - eth.c, 702
- ETH_INT_MASK_BUSY_M_OFFSET
 - eth.c, 702
- ETH_INT_MASK_RXB_M_OFFSET
 - eth.c, 702
- ETH_INT_MASK_RXC_M_OFFSET
 - eth.c, 702
- ETH_INT_MASK_RXE_M_OFFSET
 - eth.c, 702
- ETH_INT_MASK_TXB_M_OFFSET
 - eth.c, 702
- ETH_INT_MASK_TXC_M_OFFSET
 - eth.c, 702
- ETH_INT_MASK_TXE_M_OFFSET
 - eth.c, 702
- ETH_INT_SOURCE
 - eth.c, 702
- ETH_INT_SOURCE_BUSY_OFFSET
 - eth.c, 702
- ETH_INT_SOURCE_RXB_OFFSET
 - eth.c, 702
- ETH_INT_SOURCE_RXC_OFFSET
 - eth.c, 702
- ETH_INT_SOURCE_RXE_OFFSET
 - eth.c, 702
- ETH_INT_SOURCE_TXB_OFFSET
 - eth.c, 702
- ETH_INT_SOURCE_TXC_OFFSET
 - eth.c, 702
- ETH_INT_SOURCE_TXE_OFFSET
 - eth.c, 702
- ETH_IPGR1
 - eth.c, 702
- ETH_IPGR2
 - eth.c, 702
- ETH_IPGT
 - eth.c, 702
- eth_irq
 - eth.c, 704
- ETH_MAC_ADDR0
 - eth.c, 702
- ETH_MAC_ADDR1
 - eth.c, 702
- ETH_MAXPL
 - eth.c, 702
- ETH_MIIADDR_FIAD_OFFSET
 - eth.c, 702
- ETH_MIIADDR_FIAD_WIDTH
 - eth.c, 702
- ETH_MIIADDR_RGAD_OFFSET
 - eth.c, 702
- ETH_MIIADDR_RGAD_WIDTH
 - eth.c, 702
- ETH_MIIADDRESS
 - eth.c, 702
- ETH_MIICOMM_RSTAT_OFFSET
 - eth.c, 702
- ETH_MIICOMM_SCANS_OFFSET
 - eth.c, 702
- ETH_MIICOMM_WCDATA_OFFSET
 - eth.c, 702
- ETH_MIICOMMAND
 - eth.c, 702
- ETH_MIIMODER
 - eth.c, 702
- ETH_MIIMODER_CLKDIV_OFFSET
 - eth.c, 702
- ETH_MIIMODER_CLKDIV_WIDTH
 - eth.c, 702
- ETH_MIIMODER_MRST_OFFSET
 - eth.c, 702
- ETH_MIIMODER_NOPRE_OFFSET
 - eth.c, 702
- ETH_MIIRX_DATA
 - eth.c, 702
- ETH_MIISTAT_BUSY_OFFSET
 - eth.c, 702

- ETH_MIISTAT_FAIL_OFFSET
 - [eth.c, 702](#)
- ETH_MIISTAT_NVALID_OFFSET
 - [eth.c, 702](#)
- ETH_MIISTATUS
 - [eth.c, 702](#)
- ETH_MIITX_DATA
 - [eth.c, 702](#)
- ETH_MODER
 - [eth.c, 702](#)
- ETH_MODER_BRO_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_CRCEN_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_DLYCRCEN_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_DMAEN_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_EXDFREN_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_FULLD_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_HUGEN_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_IAM_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_IFG_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_LOOPBCK_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_NOBCKOF_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_NOPRE_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_PAD_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_PRO_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_RECSMALL_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_RST_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_RXEN_OFFSET
 - [eth.c, 702](#)
- ETH_MODER_TXEN_OFFSET
 - [eth.c, 702](#)
- ETH_PACKETLEN
 - [eth.c, 702](#)
- ETH_PACKETLEN_MAXFL_OFFSET
 - [eth.c, 702](#)
- ETH_PACKETLEN_MAXFL_WIDTH
 - [eth.c, 702](#)
- ETH_PACKETLEN_MINFL_OFFSET
 - [eth.c, 702](#)
- ETH_PACKETLEN_MINFL_WIDTH
 - [eth.c, 702](#)
- eth_read32
 - [eth.c, 704](#)
- eth_read_rx_file
 - [eth.c, 704](#)
- eth_reset
 - [eth.c, 704](#)
- ETH_RTX_FILE
 - [eth.c, 702](#)
- ETH_RTX SOCK
 - [eth.c, 702](#)
- eth_rtx_type
 - [eth.c, 704](#)
- ETH_RTX_VAPI
 - [eth.c, 702](#)
- ETH_RX_BD_COLLISION_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_CRC_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_DRIBBLE_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_INVALID_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_IRQ_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_LENGTH_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_LENGTH_WIDTH
 - [eth.c, 702](#)
- ETH_RX_BD_MISS_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_READY_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_TOOBIG_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_TOOSHORT_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_UVERRUN_OFFSET
 - [eth.c, 702](#)
- ETH_RX_BD_WRAP_OFFSET
 - [eth.c, 702](#)
- eth_rx_channel
 - [eth.c, 705](#)
- eth_rx_next_packet
 - [eth.c, 705](#)
- eth_rxfile
 - [eth.c, 705](#)
- ETH_RXSTATE_IDLE
 - [eth.c, 702](#)
- ETH_RXSTATE_RECV
 - [eth.c, 702](#)
- ETH_RXSTATE_WAIT4BD
 - [eth.c, 702](#)

- ETH_RXSTATE_WRITEFIFO
 - eth.c, 702
- eth_sec_end
 - eth.c, 705
- eth_sec_start
 - eth.c, 708
- eth_skip_rx_file
 - eth.c, 708
- eth_sockif
 - eth.c, 708
- eth_status
 - eth.c, 708
- ETH_TX_BD_COLLISION_OFFSET
 - eth.c, 702
- ETH_TX_BD_CRC_OFFSET
 - eth.c, 702
- ETH_TX_BD_DEFER_OFFSET
 - eth.c, 702
- ETH_TX_BD_IRQ_OFFSET
 - eth.c, 702
- ETH_TX_BD_LAST_OFFSET
 - eth.c, 702
- ETH_TX_BD_LENGTH_OFFSET
 - eth.c, 702
- ETH_TX_BD_LENGTH_WIDTH
 - eth.c, 702
- ETH_TX_BD_NO_CARRIER_OFFSET
 - eth.c, 702
- ETH_TX_BD_NUM
 - eth.c, 702
- ETH_TX_BD_PAD_OFFSET
 - eth.c, 702
- ETH_TX_BD_PAUSE_OFFSET
 - eth.c, 702
- ETH_TX_BD_READY_OFFSET
 - eth.c, 702
- ETH_TX_BD_RETRANSMIT_OFFSET
 - eth.c, 702
- ETH_TX_BD_RETRY_OFFSET
 - eth.c, 702
- ETH_TX_BD_RETRY_WIDTH
 - eth.c, 702
- ETH_TX_BD_UNDERRUN_OFFSET
 - eth.c, 702
- ETH_TX_BD_WRAP_OFFSET
 - eth.c, 702
- eth_tx_channel
 - eth.c, 708
- eth_txfile
 - eth.c, 708
- ETH_TXSTATE_IDLE
 - eth.c, 702
- ETH_TXSTATE_READFIFO
 - eth.c, 702
- ETH_TXSTATE_TRANSMIT
 - eth.c, 702
- ETH_TXSTATE_WAIT4BD
 - eth.c, 702
- eth_vapi_id
 - eth.c, 708
- eth_vapi_read
 - eth.c, 709
- eth_write32
 - eth.c, 709
- eth_write_tx_bd_num
 - eth.c, 709
- ether_addr, 115
 - ether_addr_octet, 115
- ETHER_ADDR_LEN
 - eth.c, 702
- ether_addr_octet
 - ether_addr, 115
- ETHER_CRC_LEN
 - eth.c, 702
- ether_dhost
 - ether_header, 116
- ETHER_HDR_LEN
 - eth.c, 702
- ether_header, 116
 - ether_dhost, 116
 - ether_shost, 116
 - ether_type, 116
- ETHER_IS_VALID_LEN
 - eth.c, 702
- ETHER_MAX_LEN
 - eth.c, 702
- ETHER_MIN_LEN
 - eth.c, 702
- ether_shost
 - ether_header, 116
- ether_type
 - ether_header, 116
- ETHER_TYPE_LEN
 - eth.c, 702
- ETHERMIN
 - eth.c, 702
- ETHERMTU
 - eth.c, 702
- ETHERTYPE_ARP
 - eth.c, 702
- ETHERTYPE_IP
 - eth.c, 702
- ETHERTYPE_NTRAILER
 - eth.c, 702
- ETHERTYPE_PUP
 - eth.c, 702
- ETHERTYPE_REVARP
 - eth.c, 702

- ETHERTYPE_TRAIL
 - eth.c, 702
- EV_CURRENT
 - elf.h, 260
- EV_NONE
 - elf.h, 260
- EV_NUM
 - elf.h, 260
- eval_direct16
 - abstract.c, 214
 - abstract.h, 232
- eval_direct32
 - abstract.c, 214
 - abstract.h, 233
- eval_direct8
 - abstract.c, 215
 - abstract.h, 233
- eval_insn
 - abstract.c, 215
 - abstract.h, 234
- eval_insn_ops
 - dyn-rec.c, 337
- eval_label
 - labels.c, 269
 - labels.h, 271
- eval_mem16
 - abstract.c, 215
 - abstract.h, 234
- eval_mem32
 - abstract.c, 216
 - abstract.h, 235
- eval_mem8
 - abstract.c, 216
 - abstract.h, 235
- eval_mem_16_inv
 - abstract.c, 217
- eval_mem_16_inv_direct
 - abstract.c, 217
- eval_mem_32_inv
 - abstract.c, 218
- eval_mem_32_inv_direct
 - abstract.c, 218
- eval_mem_8_inv
 - abstract.c, 218
- eval_mem_8_inv_direct
 - abstract.c, 218
- eval_operand_val
 - execute.c, 371
 - execute.h, 264
- evalsim_mem16
 - abstract.c, 219
 - abstract.h, 236
- evalsim_mem32
 - abstract.c, 219
 - abstract.h, 236
- evalsim_mem8
 - abstract.c, 219
 - abstract.h, 237
- evalsim_reg
 - execute.c, 371
 - execute.h, 265
- except.c
 - except_handle, 293
 - except_pending, 293
 - op_join_mem_cycles, 293
- except.h
 - EXCEPT_ALIGN, 296
 - EXCEPT_BUSERR, 296
 - EXCEPT_DPF, 296
 - EXCEPT_DTLBMISS, 296
 - EXCEPT_FPE, 296
 - except_handle, 296
 - EXCEPT_ILLEGAL, 296
 - EXCEPT_INT, 296
 - EXCEPT_IPF, 296
 - EXCEPT_ITLBMISS, 296
 - EXCEPT_NONE, 296
 - except_pending, 297
 - EXCEPT_RANGE, 296
 - EXCEPT_RESET, 296
 - EXCEPT_SYSCALL, 296
 - EXCEPT_TICK, 296
 - EXCEPT_TRAP, 296
- EXCEPT_ALIGN
 - except.h, 296
- EXCEPT_BUSERR
 - except.h, 296
- EXCEPT_DPF
 - except.h, 296
- EXCEPT_DTLBMISS
 - except.h, 296
- EXCEPT_FPE
 - except.h, 296
- except_handle
 - except.c, 293
 - except.h, 296
- EXCEPT_ILLEGAL
 - except.h, 296
- EXCEPT_INT
 - except.h, 296
- EXCEPT_IPF
 - except.h, 296
- EXCEPT_ITLBMISS
 - except.h, 296
- EXCEPT_NONE
 - except.h, 296
- except_pending
 - except.c, 293

- except.h, 297
- EXCEPT_RANGE
 - except.h, 296
- EXCEPT_RESET
 - except.h, 296
- EXCEPT_SYSCALL
 - except.h, 296
- EXCEPT_TICK
 - except.h, 296
- EXCEPT_TRAP
 - except.h, 296
- exe_log
 - config, 55
- exe_log_end
 - config, 55
- exe_log_fn
 - config, 55
- EXE_LOG_HARDWARE
 - sim-config.h, 810
- exe_log_marker
 - config, 55
- EXE_LOG_SIMPLE
 - sim-config.h, 810
- EXE_LOG_SOFTWARE
 - sim-config.h, 810
- exe_log_start
 - config, 55
- exe_log_type
 - config, 55
- exec_main
 - execute.c, 372
 - execute.h, 265
- execute.c
 - analysis, 367
 - breakpoint, 375
 - check_depend, 368
 - cpu_clock, 368
 - cpu_reset, 369
 - cpu_state, 375
 - decode_execute, 369
 - decode_execute_wrapper, 369
 - do_stats, 375
 - dump_exe_log, 370
 - dumpreg, 370
 - eval_operand_val, 371
 - evalsim_reg, 371
 - exec_main, 372
 - fetch, 374
 - hist_exec_tail, 375
 - issued_per_cycle, 376
 - l_invalid, 374
 - multissue, 376
 - next_delay_insn, 376
 - pcnext, 376
 - sbuf_buf, 376
 - sbuf_count, 376
 - sbuf_head, 376
 - sbuf_load, 374
 - sbuf_prev_cycles, 376
 - sbuf_store, 375
 - sbuf_tail, 376
 - sbuf_total_cyc, 376
 - sbuf_wait_cyc, 376
 - setsim_reg, 375
 - update_pc, 375
- execute.h
 - analysis, 262
 - cpu_clock, 262
 - cpu_reset, 263
 - cpu_state, 267
 - CURINSN, 262
 - depend_operands, 263
 - do_stats, 267
 - dump_exe_log, 263
 - dumpreg, 264
 - eval_operand_val, 264
 - evalsim_reg, 265
 - exec_main, 265
 - hist_exec_tail, 267
 - pcnext, 267
 - sbuf_total_cyc, 267
 - sbuf_wait_cyc, 267
 - SET_OV_FLAG_FN, 262
 - setsim_reg, 267
- EXECUTE_DEVICE_DIAGNOSTICS
 - atacmd.h, 624
- expand_bb
 - bb.c, 444
 - cuc.h, 473
- expand_branch
 - load.c, 496
- expand_calls
 - load.c, 496
- expand_memory
 - load.c, 497
- expand_signed
 - load.c, 497
- ext
 - config, 55
- EXT_CAST
 - op.c, 423
- ext_int
 - runtime, 165
- EXT_NAME
 - op.c, 423
- ext_read
 - generic.c, 723
- EXT_TYPE

- op.c, 423
- ext_write
 - generic.c, 723
- extend_imm
 - or32.c, 433
- external_clock
 - gpio_device, 124
- extract_function
 - cuc.c, 454
- f_flags
 - COFF_filehdr, 45
- f_magic
 - COFF_filehdr, 45
- f_nscns
 - COFF_filehdr, 45
- f_nsyms
 - COFF_filehdr, 45
- f_opthdr
 - COFF_filehdr, 45
- f_symptr
 - COFF_filehdr, 45
- f_timdat
 - COFF_filehdr, 45
- FALSE
 - gdbcomm.h, 535
- fb.c
 - CAM_SIZEX, 714
 - CAM_SIZEY, 714
 - change_buf_addr, 715
 - CNV16, 714
 - CNV32, 714
 - fb_baseaddr, 715
 - FB_BUFADDR, 714
 - FB_CAMBUFADDR, 714
 - FB_CAMPOSADDR, 714
 - FB_CTRL, 714
 - fb_dump_image24, 715
 - fb_dump_image8, 715
 - fb_enabled, 715
 - fb_filename, 715
 - fb_job, 715
 - FB_PAL, 714
 - fb_read32, 716
 - fb_refresh_rate, 716
 - fb_reset, 716
 - fb_sec_end, 716
 - fb_sec_start, 717
 - FB_SIZEX, 714
 - FB_SIZEY, 714
 - FB_WRAP, 714
 - fb_write32, 717
 - REFRESH_DIVIDER, 714
 - reg_fb_sec, 717
- fb.h
 - reg_fb_sec, 719
- fb_baseaddr
 - fb.c, 715
- FB_BUFADDR
 - fb.c, 714
- FB_CAMBUFADDR
 - fb.c, 714
- FB_CAMPOSADDR
 - fb.c, 714
- FB_CTRL
 - fb.c, 714
- fb_dump_image24
 - fb.c, 715
- fb_dump_image8
 - fb.c, 715
- fb_enabled
 - fb.c, 715
- fb_filename
 - fb.c, 715
- fb_job
 - fb.c, 715
- FB_PAL
 - fb.c, 714
- fb_read32
 - fb.c, 716
- fb_refresh_rate
 - fb.c, 716
- fb_reset
 - fb.c, 716
- fb_sec_end
 - fb.c, 716
- fb_sec_start
 - fb.c, 717
- FB_SIZEX
 - fb.c, 714
- FB_SIZEY
 - fb.c, 714
- fb_state, 117
 - addr, 118
 - baseaddr, 118
 - cam_addr, 118
 - camera_pos, 118
 - camerax, 118
 - cameray, 118
 - ctrl, 118
 - enabled, 118
 - filename, 118
 - in_refresh, 118
 - pal, 118
 - pic, 118
 - refresh, 118
 - refresh_count, 118
 - refresh_rate, 118

- FB_WRAP
 - fb.c, 714
- fb_write32
 - fb.c, 717
- for
 - dev_16450, 74
- fd
 - eth_device, 113
 - vapi_handler, 178
- fd.c
 - fd_channel_ops, 664
 - fd_init, 664
 - fd_isok, 664
 - fd_read, 664
 - fd_status, 664
 - fd_status_fd, 664
 - fd_write, 664
- fd.h
 - fd_channel_ops, 665
 - fd_read, 665
 - fd_write, 665
- fd_channel, 119
 - fdin, 119
 - fdout, 119
- fd_channel_ops
 - fd.c, 664
 - fd.h, 665
- fd_init
 - fd.c, 664
- fd_isok
 - fd.c, 664
- fd_read
 - fd.c, 664
 - fd.h, 665
- fd_status
 - fd.c, 664
- fd_status_fd
 - fd.c, 664
- fd_write
 - fd.c, 664
 - fd.h, 665
- fdeps
 - _cuc_func, 16
- fdin
 - fd_channel, 119
- fdout
 - fd_channel, 119
- fds
 - file_channel, 120
 - tcp_channel, 176
 - tty_channel, 177
 - vapi.c, 870
 - xterm_channel, 181
- features
 - ata_device, 22
- fetch
 - execute.c, 374
- fetch_pagefaults
 - immustats_entry, 130
- fetch_tlbhit
 - immustats_entry, 130
- fetch_tlbmiss
 - immustats_entry, 130
- fexe_log
 - runtime, 165
- FIELD_MASK
 - fields.h, 721
- FIELD_SHIFT
 - fields.h, 721
- fields.h
 - ASSIGN_FLAG, 720
 - CLEAR_FLAG, 720
 - FIELD_MASK, 721
 - FIELD_SHIFT, 721
 - FLAG_MASK, 721
 - FLAG_SHIFT, 721
 - GET_FIELD, 721
 - SET_FIELD, 721
 - SET_FLAG, 721
 - TEST_FLAG, 721
- fifo_len
 - dev_16450, 74
- file
 - ata_device, 22
- file.c
 - file_channel_ops, 667
 - file_close, 667
 - file_free, 667
 - file_init, 667
 - file_open, 667
- file.h
 - file_channel_ops, 668
- file_channel, 120
 - fds, 120
 - namein, 120
 - nameout, 120
- file_channel_ops
 - file.c, 667
 - file.h, 668
- file_close
 - file.c, 667
- file_free
 - file.c, 667
- file_init
 - file.c, 667
- file_open
 - file.c, 667
- filename

- fb_state, 118
- runtime, 165
- vga_state, 180
- find_channel_factory
 - channel.c, 661
- find_dma_controller_ch
 - dma.c, 692
 - dma.h, 695
- find_handler
 - vapi.c, 866
- find_jump_loc
 - dyn-rec.c, 337
- find_label
 - labels.c, 269
 - labels.h, 271
- find_lsc_index
 - verilog.c, 511
- find_t
 - dyn-rec.c, 337
- firmware
 - ata_device, 22
- first
 - cuc_bb, 63
- FIXME
 - debug.h, 820
- FIXME_
 - debug.h, 820
- FIXME_ON
 - debug.h, 820
- FLAG_MASK
 - fields.h, 721
- FLAG_REG
 - cuc.h, 469
- FLAG_SHIFT
 - fields.h, 721
- flog
 - cuc.c, 463
 - cuc.h, 481
- FLUSH_CACHE
 - atacmd.h, 624
- fmprof
 - runtime, 165
- FMTLEN
 - simprintf.c, 837
- fmtstr
 - simprintf.c, 837
- fo
 - sprs.c, 322
- FORCE_RET
 - op-i386.h, 403
- format_func_options
 - cuc.c, 454
- forward
 - branchstat, 32
- fout
 - runtime, 165
- fprof
 - mprofiler.c, 589
 - profiler.c, 779
 - runtime, 165
- free
 - channel_ops, 39
- free_func
 - bb.c, 444
 - cuc.h, 473
- free_job_queue
 - scheduler_struct, 168
- freemem
 - parse.c, 277
- from
 - _csm_list, 14
 - cuc_conv, 64
- from_spr
 - spr_def, 173
- fstats
 - stats.c, 282
- fstats_entry, 121
 - cnt_dynamic, 121
 - depend, 121
 - insn1, 121
 - insn2, 121
- FSTATS_LEN
 - stats.c, 281
- func
 - config_param, 58
 - cuc.c, 463
 - sched_entry, 167
- func_index
 - verilog.c, 511
- func_offset
 - reloc, 161
- func_struct, 122
 - addr, 122
 - calls, 122
 - cum_cycles, 122
 - name, 122
- func_unit_str
 - stats.c, 282
- func_v
 - cuc.c, 463
- gdb.h
 - ERR_CRC, 529
 - ERR_MEM, 529
 - ERR_NONE, 529
 - JTAG_PROXY_ACCESS_EXCEPTION, 530
 - JTAG_PROXY_COMMAND_NOT_-
IMPLEMENTED, 530

- JTAG_PROXY_INVALID_ADDRESS, 530
- JTAG_PROXY_INVALID_CHAIN, 530
- JTAG_PROXY_INVALID_COMMAND, 530
- JTAG_PROXY_INVALID_LENGTH, 530
- JTAG_PROXY_NO_CONNECTION, 530
- JTAG_PROXY_OUT_OF_MEMORY, 530
- JTAG_PROXY_PROTOCOL_ERROR, 530
- JTAG_PROXY_SERVER_TERMINATED, 530
- OR1K_JTAG_COMMAND_CHAIN, 530
- OR1K_JTAG_COMMAND_READ, 530
- OR1K_JTAG_COMMAND_READ_BLOCK, 530
- OR1K_JTAG_COMMAND_WRITE, 530
- OR1K_JTAG_COMMAND_WRITE_BLOCK, 530
- or1k_jtag_errors, 529
- or1k_jtag_proxy_protocol_commands, 530
- GDB_BUF_MAX
 - rsp-server.c, 539
- gdb_enabled
 - config, 55
- gdb_fd
 - gdbcomm.c, 534
- gdb_read
 - gdbcomm.c, 532
- gdb_request
 - gdbcomm.c, 532
- gdb_write
 - gdbcomm.c, 532
- gdbcomm.c
 - block_jtag, 532
 - gdb_fd, 534
 - gdb_read, 532
 - gdb_request, 532
 - gdb_write, 532
 - gdbcomm_init, 533
 - get_server_socket, 533
 - handle_server_socket, 533
 - jtag_request, 533
 - protocol_clean, 534
 - server_fd, 534
 - server_ip, 534
 - server_port, 534
 - tcp_level, 534
- gdbcomm.h
 - block_jtag, 535
 - boolean, 535
 - FALSE, 535
 - gdbcomm_init, 535
 - handle_server_socket, 535
 - TRUE, 535
- gdbcomm_init
 - gdbcomm.c, 533
 - gdbcomm.h, 535
- GEN
 - verilog.c, 511
- gen_code_ent
 - dyn-rec.h, 353
- gen_code_proto
 - dyngen.c, 364
- gen_eval_operands
 - generate.c, 377
- gen_func_proto
 - dyngen.c, 363
- gen_func_reloc
 - archf, 18
- gen_j_imm
 - dyn-rec.c, 338
- gen_j_reg
 - dyn-rec.c, 338
- gen_l_add
 - dyn-rec.c, 338
- gen_l_addc
 - dyn-rec.c, 338
- gen_l_and
 - dyn-rec.c, 338
- gen_l_bf
 - dyn-rec.c, 338
- gen_l_bnf
 - dyn-rec.c, 338
- gen_l_cmov
 - dyn-rec.c, 339
- gen_l_cust1
 - dyn-rec.c, 340
- gen_l_cust2
 - dyn-rec.c, 340
- gen_l_cust3
 - dyn-rec.c, 340
- gen_l_cust4
 - dyn-rec.c, 340
- gen_l_cust5
 - dyn-rec.c, 340
- gen_l_cust6
 - dyn-rec.c, 340
- gen_l_cust7
 - dyn-rec.c, 340
- gen_l_cust8
 - dyn-rec.c, 340
- gen_l_div
 - dyn-rec.c, 340
- gen_l_divu
 - dyn-rec.c, 340
- gen_l_extbs
 - dyn-rec.c, 340
- gen_l_extbz
 - dyn-rec.c, 340
- gen_l_exths
 - dyn-rec.c, 340

- dyn-rec.c, 340
- gen_l_exthz
 - dyn-rec.c, 340
- gen_l_extws
 - dyn-rec.c, 340
- gen_l_extwz
 - dyn-rec.c, 340
- gen_l_ffl
 - dyn-rec.c, 340
- gen_l_invalid
 - dyn-rec.c, 340
- gen_l_j
 - dyn-rec.c, 340
- gen_l_jal
 - dyn-rec.c, 340
- gen_l_jalr
 - dyn-rec.c, 341
- gen_l_jr
 - dyn-rec.c, 341
- gen_l_lbs
 - dyn-rec.c, 341
- gen_l_lbz
 - dyn-rec.c, 342
- gen_l_lhs
 - dyn-rec.c, 342
- gen_l_lhz
 - dyn-rec.c, 342
- gen_l_lws
 - dyn-rec.c, 342
- gen_l_lwz
 - dyn-rec.c, 342
- gen_l_mac
 - dyn-rec.c, 342
- gen_l_macrc
 - dyn-rec.c, 343
- gen_l_mfspr
 - dyn-rec.c, 343
- gen_l_movhi
 - dyn-rec.c, 343
- gen_l_msb
 - dyn-rec.c, 343
- gen_l_mtspr
 - dyn-rec.c, 343
- gen_l_mul
 - dyn-rec.c, 343
- gen_l_mulu
 - dyn-rec.c, 343
- gen_l_nop
 - dyn-rec.c, 343
- gen_l_or
 - dyn-rec.c, 343
- gen_l_rfe
 - dyn-rec.c, 343
- gen_l_sb
 - dyn-rec.c, 343
- gen_l_sfeq
 - dyn-rec.c, 343
- gen_l_sfges
 - dyn-rec.c, 344
- gen_l_sfges
 - dyn-rec.c, 344
- gen_l_sfgts
 - dyn-rec.c, 344
- gen_l_sfgtu
 - dyn-rec.c, 344
- gen_l_sfles
 - dyn-rec.c, 344
- gen_l_sfleu
 - dyn-rec.c, 344
- gen_l_sflts
 - dyn-rec.c, 344
- gen_l_sfltu
 - dyn-rec.c, 344
- gen_l_sfne
 - dyn-rec.c, 344
- gen_l_sh
 - dyn-rec.c, 344
- gen_l_sll
 - dyn-rec.c, 344
- gen_l_sra
 - dyn-rec.c, 344
- gen_l_srl
 - dyn-rec.c, 344
- gen_l_sub
 - dyn-rec.c, 344
- gen_l_sw
 - dyn-rec.c, 344
- gen_l_sys
 - dyn-rec.c, 344
- gen_l_trap
 - dyn-rec.c, 345
- gen_l_xor
 - dyn-rec.c, 345
- gen_lf_add_s
 - dyn-rec.c, 345
- gen_lf_div_s
 - dyn-rec.c, 345
- gen_lf_ftoi_s
 - dyn-rec.c, 345
- gen_lf_itof_s
 - dyn-rec.c, 345
- gen_lf_madd_s
 - dyn-rec.c, 345
- gen_lf_mul_s
 - dyn-rec.c, 345
- gen_lf_rem_s
 - dyn-rec.c, 346
- gen_lf_sfeq_s

- dyn-rec.c, 346
- gen_lf_sfge_s
 - dyn-rec.c, 346
- gen_lf_sfgt_s
 - dyn-rec.c, 346
- gen_lf_sfle_s
 - dyn-rec.c, 346
- gen_lf_sflt_s
 - dyn-rec.c, 346
- gen_lf_sfne_s
 - dyn-rec.c, 347
- gen_lf_sub_s
 - dyn-rec.c, 347
- gen_op_mark_loc
 - dyn-rec.c, 347
- gen_option
 - cuc.c, 454
- gen_reloc
 - archf, 18
- generate.c
 - gen_eval_operands, 377
 - generate_body, 378
 - generate_footer, 378
 - generate_header, 378
 - in_file, 379
 - main, 378
 - out_file, 379
 - out_lines, 379
 - output_call, 378
 - output_function, 379
 - shift_fprintf, 379
 - write_to_reg, 379
- generate_bb_seq
 - bb.c, 444
 - cuc.h, 473
- generate_body
 - generate.c, 378
- generate_footer
 - generate.c, 378
- generate_function
 - cuc.c, 456
- generate_header
 - generate.c, 378
- generate_main
 - verilog.c, 511
 - verilog.h, 513
- generate_time_pretty
 - abstract.c, 220
 - abstract.h, 237
- generic.c
 - ext_read, 723
 - ext_write, 723
 - generic_baseaddr, 723
 - generic_byte_enabled, 723
 - generic_enabled, 723
 - generic_hw_enabled, 723
 - generic_name, 723
 - generic_read_byte, 723
 - generic_read_hw, 723
 - generic_read_word, 723
 - generic_reset, 724
 - generic_sec_end, 724
 - generic_sec_start, 724
 - generic_size, 725
 - generic_status, 725
 - generic_word_enabled, 725
 - generic_write_byte, 725
 - generic_write_hw, 725
 - generic_write_word, 725
 - reg_generic_sec, 725
- generic.h
 - reg_generic_sec, 728
- GENERIC_BYTE
 - dev_generic, 76
- GENERIC_HW
 - dev_generic, 76
- GENERIC_READ
 - dev_generic, 76
- GENERIC_WORD
 - dev_generic, 76
- GENERIC_WRITE
 - dev_generic, 76
- generic_baseaddr
 - generic.c, 723
- generic_byte_enabled
 - generic.c, 723
- generic_close
 - channels/generic.c, 727
 - channels/generic.h, 729
- generic_enabled
 - generic.c, 723
- generic_free
 - channels/generic.c, 727
 - channels/generic.h, 729
- generic_gen_op
 - dyn-rec.c, 335
- generic_hw_enabled
 - generic.c, 723
- generic_name
 - generic.c, 723
- generic_open
 - channels/generic.c, 727
 - channels/generic.h, 729
- generic_read_byte
 - generic.c, 723
- generic_read_hw
 - generic.c, 723
- generic_read_word

- generic.c, 723
- generic_reset
 - generic.c, 724
- generic_sec_end
 - generic.c, 724
- generic_sec_start
 - generic.c, 724
- generic_size
 - generic.c, 725
- generic_status
 - generic.c, 725
- generic_word_enabled
 - generic.c, 725
- generic_write_byte
 - generic.c, 725
- generic_write_hw
 - generic.c, 725
- generic_write_word
 - generic.c, 725
- get_devint_reg
 - debug-unit.c, 521
- GET_FIELD
 - fields.h, 721
- get_func_len
 - bff, 28
- get_func_name
 - bff, 28
- get_func_reloc
 - bff, 28
- get_func_start
 - bff, 28
- get_label
 - labels.c, 269
 - labels.h, 271
- GET_MEDIA_STATUS
 - atacmd.h, 624
- get_packet
 - rsp-server.c, 541
- get_pc
 - common-i386.h, 325
- get_real_func_len
 - archf, 18
- get_rsp_char
 - rsp-server.c, 541
- get_server_socket
 - gdbcomm.c, 533
 - vapi.c, 866
- get_sp
 - rec-i386.h, 437
- glue
 - dyn-rec.h, 353
 - op-1t-op.h, 393
 - op-2t-op.h, 395
 - op-3t-op.h, 397
- op-extend-op.h, 401
- op-ff1-op.h, 402
- op-mac-op.h, 405
- op-swhb-op.h, 411
- op-t-reg-mov-op.h, 415
- gpio.c
 - GPIO_NUM_VAPI_IDS, 732
 - GPIO_VAPI_AUX, 732
 - GPIO_VAPI_CLOCK, 732
 - GPIO_VAPI_DATA, 732
 - GPIO_VAPI_RGPIO_AUX, 732
 - GPIO_VAPI_RGPIO_CTRL, 732
 - GPIO_VAPI_RGPIO_INTE, 732
 - GPIO_VAPI_RGPIO_OE, 732
 - GPIO_VAPI_RGPIO_PTRIG, 732
 - GPIO_ADDR_SPACE, 732
 - gpio_base_vapi_id, 733
 - gpio_baseaddr, 733
 - gpio_clock, 733
 - gpio_device_clock, 733
 - gpio_do_int, 733
 - gpio_enabled, 734
 - gpio_external_clock, 734
 - gpio_irq, 734
 - gpio_read32, 735
 - gpio_reset, 735
 - gpio_sec_end, 735
 - gpio_sec_start, 735
 - gpio_status, 736
 - gpio_vapi_read, 736
 - gpio_write32, 736
 - reg_gpio_sec, 736
 - RGPIO_AUX, 732
 - RGPIO_CTRL, 732
 - RGPIO_CTRL_ECLK, 732
 - RGPIO_CTRL_INTE, 732
 - RGPIO_CTRL_INTS, 732
 - RGPIO_CTRL_NEC, 732
 - RGPIO_IN, 732
 - RGPIO_INTE, 732
 - RGPIO_INTS, 732
 - RGPIO_OE, 732
 - RGPIO_OUT, 732
 - RGPIO_PTRIG, 732
- gpio.h
 - reg_gpio_sec, 737
- GPIO_NUM_VAPI_IDS
 - gpio.c, 732
- GPIO_VAPI_AUX
 - gpio.c, 732
- GPIO_VAPI_CLOCK
 - gpio.c, 732
- GPIO_VAPI_DATA
 - gpio.c, 732

- GPIO_VAPI_RGPIO_AUX
 - gpio.c, 732
- GPIO_VAPI_RGPIO_CTRL
 - gpio.c, 732
- GPIO_VAPI_RGPIO_INTE
 - gpio.c, 732
- GPIO_VAPI_RGPIO_OE
 - gpio.c, 732
- GPIO_VAPI_RGPIO_PTRIG
 - gpio.c, 732
- GPIO_ADDR_SPACE
 - gpio.c, 732
- gpio_base_vapi_id
 - gpio.c, 733
- gpio_baseaddr
 - gpio.c, 733
- gpio_clock
 - gpio.c, 733
- gpio_device, 123
 - aux, 124
 - auxiliary_inputs, 124
 - base_vapi_id, 124
 - baseaddr, 124
 - ctrl, 124
 - curr, 124
 - enabled, 124
 - external_clock, 124
 - gpio_number, 124
 - in, 124
 - inte, 124
 - ints, 124
 - irq, 124
 - next, 124
 - oe, 124
 - out, 124
 - ptrig, 124
- gpio_device_clock
 - gpio.c, 733
- gpio_do_int
 - gpio.c, 733
- gpio_enabled
 - gpio.c, 734
- gpio_external_clock
 - gpio.c, 734
- gpio_irq
 - gpio.c, 734
- gpio_number
 - gpio_device, 124
- gpio_read32
 - gpio.c, 735
- gpio_reset
 - gpio.c, 735
- gpio_sec_end
 - gpio.c, 735
- gpio_sec_start
 - gpio.c, 735
- gpio_status
 - gpio.c, 736
- gpio_vapi_read
 - gpio.c, 736
- gpio_write32
 - gpio.c, 736
- GPR_T
 - def-op-t.h, 327
- group_bits
 - mprofiler.c, 589
- handle_rsp
 - rsp-server.c, 542
 - rsp-server.h, 560
- handle_server_socket
 - gdbcomm.c, 533
 - gdbcomm.h, 535
- handle_sim_command
 - sim-cmd.c, 785
 - sim-cmd.h, 799
- handler_fits_id
 - vapi.c, 867
- has_breakpoint
 - labels.c, 269
 - labels.h, 271
- hash
 - mprofiler.c, 589
- hash0
 - eth_device, 113
- hash1
 - eth_device, 113
- hash_add
 - mprofiler.c, 588
- HASH_FUNC
 - mprofiler.c, 588
- hash_get
 - mprofiler.c, 588
- HASH_SIZE
 - mprofiler.c, 588
- hazards
 - config, 55
- hazardwait
 - runtime, 165
- head
 - channel.c, 661
- heads
 - ata_device, 22
- heads_per_cylinder
 - ata_device, 22
- height
 - INFOHEADER, 131
- hex

- rsp-server.c, 543
- hex2ascii
 - rsp-server.c, 544
- hex2reg
 - rsp-server.c, 544
- hexchars
 - rsp-server.c, 559
- hide_device_id
 - config, 55
- high32
 - common-i386.h, 325
- hist_exec, 125
 - addr, 125
 - next, 125
 - prev, 125
- hist_exec_tail
 - abstract.h, 243
 - execute.c, 375
 - execute.h, 267
- HISTEXEC_LEN
 - abstract.h, 231
- history
 - config, 55
- hit
 - bpbstat, 31
 - bticstat, 35
- hitdelay
 - config, 55
 - dmmu, 86
 - ic, 127
 - immu, 129
- hlen
 - vga_state, 180
- host
 - ata_device, 22
- host_len
 - dyn_page, 89
- host_page
 - dyn_page, 89
- htim
 - vga_state, 180
- hush
 - runtime, 165
- hw_enabled
 - dev_generic, 77
- i386-regs.h
 - CPU_STATE_REG, 380
 - NUM_T_REGS, 380
 - T0_REG, 380
 - T1_REG, 380
 - T2_REG, 380
- i386_gen_func_reloc
 - dyngen-i386.c, 360
- i386_gen_reloc
 - dyngen-i386.c, 360
- i386_get_real_func_len
 - dyngen-i386.c, 360
- IADDR_PAGE
 - immu.h, 585
- ic, 126
 - block_mask, 127
 - block_offset_mask, 127
 - blocksize, 127
 - blocksize_log2, 127
 - enabled, 127
 - hitdelay, 127
 - last_way, 127
 - lrus, 127
 - mem, 127
 - missdelay, 127
 - nsets, 127
 - nways, 127
 - set_mask, 127
 - tagaddr_mask, 127
 - tags, 127
 - ustates, 127
 - ustates_reload, 127
- ic_blocksize
 - icache-model.c, 199
- ic_enabled
 - icache-model.c, 199
- ic_end_sec
 - icache-model.c, 199
- ic_hitdelay
 - icache-model.c, 200
- ic_info
 - icache-model.c, 200
- ic_inv
 - icache-model.c, 200
 - icache-model.h, 204
- ic_missdelay
 - icache-model.c, 200
- ic_nsets
 - icache-model.c, 200
- ic_nways
 - icache-model.c, 200
- ic_simulate_fetch
 - icache-model.c, 201
 - icache-model.h, 204
- ic_start_sec
 - icache-model.c, 201
- ic_state
 - icache-model.c, 202
 - icache-model.h, 205
- ic_stats
 - stats.c, 282
 - stats.h, 285

- ic_ustates
 - icache-model.c, 201
- icache-model.c
 - ic_blocksize, 199
 - ic_enabled, 199
 - ic_end_sec, 199
 - ic_hitdelay, 200
 - ic_info, 200
 - ic_inv, 200
 - ic_missdelay, 200
 - ic_nsets, 200
 - ic_nways, 200
 - ic_simulate_fetch, 201
 - ic_start_sec, 201
 - ic_state, 202
 - ic_ustates, 201
 - MAX_IC_BLOCK_SIZE, 199
 - MAX_IC_SETS, 199
 - MAX_IC_WAYS, 199
 - MIN_IC_BLOCK_SIZE, 199
 - reg_ic_sec, 202
- icache-model.h
 - ic_inv, 204
 - ic_simulate_fetch, 204
 - ic_state, 205
 - reg_ic_sec, 204
- icomplet
 - cpu_state, 61
- IDENTIFY_DEVICE
 - atacmd.h, 624
- IDENTIFY_PACKET_DEVICE
 - atacmd.h, 624
- identifyfile
 - parse.c, 273
- IDLE
 - atacmd.h, 624
- IDLE_IMMEDIATE
 - atacmd.h, 624
- ier
 - dev_16450, 74
- ifr
 - eth_device, 113
- II_ADD
 - insn.h, 490
- II_AND
 - insn.h, 490
- II_BF
 - insn.h, 490
- II_CALL
 - insn.h, 490
- II_CMOV
 - insn.h, 490
- II_IS_LOAD
 - insn.h, 490
- II_IS_STORE
 - insn.h, 490
- II_LAST
 - insn.h, 490
- II_LB
 - insn.h, 490
- II_LH
 - insn.h, 490
- II_LRBB
 - insn.h, 490
- II_LW
 - insn.h, 490
- II_MASK
 - insn.h, 490
- II_MEM
 - insn.h, 490
- II_MEM_WIDTH
 - insn.h, 490
- II_MUL
 - insn.h, 490
- II_NOP
 - insn.h, 492
- II_OR
 - insn.h, 492
- II_REG
 - insn.h, 492
- II_SB
 - insn.h, 492
- II_SFREQ
 - insn.h, 492
- II_SFGE
 - insn.h, 492
- II_SFGT
 - insn.h, 492
- II_SFLE
 - insn.h, 492
- II_SFLT
 - insn.h, 492
- II_SFNE
 - insn.h, 492
- II_SH
 - insn.h, 492
- II_SIGNED
 - insn.h, 492
- ii_size
 - insn.h, 492
 - timings.c, 508
- II_SLL
 - insn.h, 492
- II_SRA
 - insn.h, 492
- II_SRL
 - insn.h, 492
- II_SUB

- insn.h, 492
- II_SW
 - insn.h, 492
- II_XOR
 - insn.h, 492
- iir
 - dev_16450, 74
- imagesize
 - INFOHEADER, 131
- imm_gen_op
 - dyn-rec.c, 335
- IMM_STATS
 - parse.c, 273
- immu, 128
 - enabled, 129
 - entrysize, 129
 - hitdelay, 129
 - lru_reload, 129
 - missdelay, 129
 - nsets, 129
 - nways, 129
 - page_mask, 129
 - page_offset_mask, 129
 - pagesize, 129
 - pagesize_log2, 129
 - set_mask, 129
 - ustates, 129
 - vpn_mask, 129
- immu.c
 - immu_enabled, 580
 - immu_end_sec, 580
 - immu_entrysize, 580
 - immu_find_tlbmr, 580
 - immu_hitdelay, 581
 - immu_missdelay, 581
 - immu_nsets, 581
 - immu_nways, 581
 - immu_pagesize, 581
 - immu_start_sec, 582
 - immu_state, 583
 - immu_translate, 582
 - immu_ustates, 582
 - itlb_status, 582
 - peek_into_itlb, 583
 - reg_immu_sec, 583
- immu.h
 - IADDR_PAGE, 585
 - immu_simulate_tlb, 585
 - immu_state, 586
 - immu_translate, 585
 - peek_into_itlb, 585
 - reg_immu_sec, 585
- immu_enabled
 - immu.c, 580
- immu_end_sec
 - immu.c, 580
- immu_entrysize
 - immu.c, 580
- immu_find_tlbmr
 - immu.c, 580
- IMMU_GOT_DISABLED
 - dyn-rec.h, 353
- IMMU_GOT_ENABLED
 - dyn-rec.h, 353
- immu_hitdelay
 - immu.c, 581
- immu_missdelay
 - immu.c, 581
- immu_nsets
 - immu.c, 581
- immu_nways
 - immu.c, 581
- immu_pagesize
 - immu.c, 581
- immu_retranslate
 - dyn-rec.c, 347
- immu_simulate_tlb
 - immu.h, 585
- immu_start_sec
 - immu.c, 582
- immu_state
 - immu.c, 583
 - immu.h, 586
- immu_stats
 - stats.c, 282
 - stats.h, 285
- immu_translate
 - immu.c, 582
 - immu.h, 585
- immu_ustates
 - immu.c, 582
- immustats_entry, 130
 - fetch_pagefaults, 130
 - fetch_tlbhit, 130
 - fetch_tlbmiss, 130
- importantcolours
 - INFOHEADER, 131
- in
 - gpio_device, 124
- in_file
 - generate.c, 379
- in_refresh
 - fb_state, 118
- in_reset
 - debug-unit.c, 523
- incorrect
 - bpbstat, 31
- index

- cuc_insn, 65
- mc, 147
- INFOHEADER, 131
 - bits, 131
 - compression, 131
 - height, 131
 - imagesize, 131
 - importantcolours, 131
 - ncolours, 131
 - planes, 131
 - size, 131
 - width, 131
 - xresolution, 131
 - yresolution, 131
- init
 - channel_ops, 39
 - mprofiler.c, 588
- init_bb_reloc
 - _cuc_func, 16
- init_breakpoints
 - labels.c, 269
 - labels.h, 271
- init_defconfig
 - sim-config.c, 802
 - sim-config.h, 811
- init_dyn_recomp
 - dyn-rec.c, 348
 - dyn-rec.h, 354
- init_labels
 - labels.c, 269
 - labels.h, 271
- INITIALIZE_DEVICE_PARAMETERS
 - atacmd.h, 624
- initstats
 - stats.c, 281
 - stats.h, 285
- insert_conditional_facts
 - adv.c, 441
 - cuc.h, 473
- insert_insns
 - cuc.h, 474
 - insn.c, 485
- INSN
 - cuc.h, 469
- insn
 - btic_entry, 34
 - cuc.h, 481
 - cuc_bb, 63
 - cuc_insn, 65
 - iqueue_entry, 132
 - load.c, 498
 - op_queue, 158
 - sstats_entry, 174
- insn.c
 - add_data_dep, 483
 - add_dep, 483
 - add_latches, 483
 - apply_edge_condition, 483
 - change_insn_type, 484
 - cmov_needed, 484
 - count_cmovs, 484
 - cse, 484
 - csm, 484
 - csm_gen, 484
 - cuc_insn_name, 484
 - dispose_list, 485
 - insert_insns, 485
 - insn_uses, 485
 - iteration, 487
 - known, 487
 - main_list, 487
 - optimize_cmov_more, 485
 - optimize_cmovs, 485
 - optimize_tree, 485
 - print_insns, 486
 - print_shared, 486
 - remove_dead, 486
 - remove_nops, 486
 - remove_trivial_regs, 487
 - search_csm, 487
 - set_io, 487
 - tmp_op, 487
 - tmp_opt, 487
 - unmark_tree, 487
- insn.h
 - change_insn_type, 492
 - cuc_insn_name, 492
 - II_ADD, 490
 - II_AND, 490
 - II_BF, 490
 - II_CALL, 490
 - II_CMOV, 490
 - II_IS_LOAD, 490
 - II_IS_STORE, 490
 - II_LAST, 490
 - II_LB, 490
 - II_LH, 490
 - II_LRBB, 490
 - II_LW, 490
 - II_MASK, 490
 - II_MEM, 490
 - II_MEM_WIDTH, 490
 - II_MUL, 490
 - II_NOP, 492
 - II_OR, 492
 - II_REG, 492
 - II_SB, 492
 - II_SFEQ, 492

- II_SFGE, 492
- II_SFGT, 492
- II_SFLE, 492
- II_SFLT, 492
- II_SFNE, 492
- II_SH, 492
- II_SIGNED, 492
- ii_size, 492
- II_SLL, 492
- II_SRA, 492
- II_SRL, 492
- II_SUB, 492
- II_SW, 492
- II_XOR, 492
- insn_size, 492
- insn_time, 492
- known, 493
- load_timing_table, 492
- print_shared, 492
- insn1
 - dstats_entry, 88
 - fstats_entry, 121
- insn2
 - dstats_entry, 88
 - fstats_entry, 121
- insn_addr
 - iqueue_entry, 132
 - op_queue, 158
- insn_ci
 - abstract.c, 228
 - abstract.h, 243
- insn_decode
 - or32.c, 433
- insn_ea
 - cpu_state, 60
- insn_extract
 - or32.c, 434
- insn_index
 - iqueue_entry, 132
 - op_queue, 158
 - or32.c, 434
- insn_indexs
 - dyn_page, 89
- insn_len
 - or32.c, 434
- insn_name
 - or32.c, 434
- insn_size
 - insn.h, 492
 - timings.c, 509
- insn_time
 - insn.h, 492
 - timings.c, 509
- insn_uses
 - insn.c, 485
- INSNAME_LEN
 - abstract.h, 231
- insns
 - dyn_page, 89
- insnset.c
 - INSTRUCTION, 384–392
- instr
 - mp_entry, 154
- INSTRUCTION
 - insnset.c, 384–392
- instructions
 - runtime, 165
- int_mask
 - eth_device, 113
- int_msk_a
 - dma_controller, 84
- int_msk_b
 - dma_controller, 84
- int_source
 - eth_device, 113
- int_src_a
 - dma_controller, 84
- int_src_b
 - dma_controller, 84
- int_val
 - param_val, 159
- inte
 - gpio_device, 124
- internal_or1ksim_time
 - libtoplevel.c, 564
- internals
 - ata_device, 22
- intrq
 - ata_device, 22
- ints
 - dev_16450, 74
 - gpio_device, 124
- iordy
 - ata_device, 22
- ipgr1
 - eth_device, 113
- ipgr2
 - eth_device, 113
- ipgt
 - eth_device, 113
- iprompt
 - runtime, 165
- iprompt_run
 - runtime, 165
- iqueue
 - cpu_state, 61
 - iqueue_entry, 132
 - insn, 132

- insn_addr, [132](#)
- insn_index, [132](#)
- iregs
 - dev_16450, [74](#)
- irq
 - ata_host, [26](#)
 - dev_16450, [74](#)
 - dma_controller, [84](#)
 - gpio_device, [124](#)
 - kbd_state, [144](#)
 - vga_state, [180](#)
- is_ata_hostadr
 - atahost.h, [657](#)
- is_power2
 - misc.c, [827](#)
 - misc.h, [828](#)
- isblank
 - isblank.c, [774](#)
 - port.h, [775](#)
- isblank.c
 - isblank, [774](#)
- isok
 - channel_ops, [39](#)
- issued_per_cycle
 - execute.c, [376](#)
- istat
 - dev_16450, [74](#)
- IT_BBEND
 - cuc.h, [469](#)
- IT_BBSTART
 - cuc.h, [469](#)
- IT_BRANCH
 - cuc.h, [469](#)
- IT_COND
 - cuc.h, [469](#)
- IT_CUT
 - cuc.h, [469](#)
- IT_FLAG1
 - cuc.h, [469](#)
- IT_FLAG2
 - cuc.h, [469](#)
- IT_INDELAY
 - cuc.h, [469](#)
- IT_LATCHED
 - cuc.h, [469](#)
- IT_MEMADD
 - cuc.h, [469](#)
- IT_MEMORY
 - cuc.h, [469](#)
- IT_OUTPUT
 - cuc.h, [469](#)
- IT_SIGNED
 - cuc.h, [469](#)
- IT_UNUSED
 - cuc.h, [469](#)
- IT_VOLATILE
 - cuc.h, [469](#)
- iteration
 - insn.c, [487](#)
- itlb_status
 - immu.c, [582](#)
- jitter
 - dev_16450, [74](#)
- job_queue
 - scheduler_struct, [168](#)
- join_bb
 - bb.c, [444](#)
- join_transfers
 - cuc/memory.c, [500](#)
- JTAG_CHAIN_DEBUG_UNIT
 - debug-unit.h, [524](#)
- JTAG_CHAIN_DEVELOPMENT
 - debug-unit.h, [524](#)
- JTAG_CHAIN_GLOBAL
 - debug-unit.h, [524](#)
- JTAG_CHAIN_TRACE
 - debug-unit.h, [524](#)
- JTAG_CHAIN_WISHBONE
 - debug-unit.h, [524](#)
- JTAG_PROXY_ACCESS_EXCEPTION
 - gdb.h, [530](#)
- JTAG_PROXY_COMMAND_NOT_-
IMPLEMENTED
 - gdb.h, [530](#)
- JTAG_PROXY_INVALID_ADDRESS
 - gdb.h, [530](#)
- JTAG_PROXY_INVALID_CHAIN
 - gdb.h, [530](#)
- JTAG_PROXY_INVALID_COMMAND
 - gdb.h, [530](#)
- JTAG_PROXY_INVALID_LENGTH
 - gdb.h, [530](#)
- JTAG_PROXY_NO_CONNECTION
 - gdb.h, [530](#)
- JTAG_PROXY_OUT_OF_MEMORY
 - gdb.h, [530](#)
- JTAG_PROXY_PROTOCOL_ERROR
 - gdb.h, [530](#)
- JTAG_PROXY_SERVER_TERMINATED
 - gdb.h, [530](#)
- jtag_request
 - gdbcomm.c, [533](#)
- jtr_chain_message, [133](#)
 - chain, [133](#)
 - command, [133](#)
 - length, [133](#)
- jtr_chain_response, [134](#)

- status, 134
- jtr_failure_response, 135
 - status, 135
- jtr_read_block_message, 136
 - address, 136
 - command, 136
 - length, 136
 - num_regs, 136
- jtr_read_block_response, 137
 - data, 137
 - num_regs, 137
 - status, 137
- jtr_read_message, 138
 - address, 138
 - command, 138
 - length, 138
- jtr_read_response, 139
 - data_h, 139
 - data_l, 139
 - status, 139
- jtr_write_block_message, 140
 - address, 140
 - command, 140
 - data, 140
 - length, 140
 - num_regs, 140
- jtr_write_block_response, 141
 - status, 141
- jtr_write_message, 142
 - address, 142
 - command, 142
 - data_h, 142
 - data_l, 142
 - length, 142
- jtr_write_response, 143
 - status, 143
- jump_local
 - op_queue, 158
- jump_local_loc
 - op_queue, 158
- kbd_baseaddr
 - ps2kbd.c, 753
- KBD_BAUD_RATE
 - ps2kbd.c, 753
- KBD_CCMD_DKI
 - ps2kbd.c, 753
- KBD_CCMD_EKI
 - ps2kbd.c, 753
- KBD_CCMD_RCB
 - ps2kbd.c, 753
- KBD_CCMD_ST1
 - ps2kbd.c, 753
- KBD_CCMD_ST2
 - ps2kbd.c, 753
- KBD_CCMD_WCB
 - ps2kbd.c, 753
- KBD_CCMDBYTE_EN
 - ps2kbd.c, 753
- KBD_CCMDBYTE_EN2
 - ps2kbd.c, 753
- KBD_CCMDBYTE_INT
 - ps2kbd.c, 753
- KBD_CCMDBYTE_INT2
 - ps2kbd.c, 753
- KBD_CCMDBYTE_SYS
 - ps2kbd.c, 753
- KBD_CCMDBYTE_XLAT
 - ps2kbd.c, 753
- KBD_CTRL
 - ps2kbd.c, 753
- KBD_DATA
 - ps2kbd.c, 753
- kbd_enabled
 - ps2kbd.c, 753
- kbd_info
 - ps2kbd.c, 753
- kbd_irq
 - ps2kbd.c, 753
- kbd_job
 - ps2kbd.c, 753
- KBD_KCMD_DK
 - ps2kbd.c, 753
- KBD_KCMD_ECHO
 - ps2kbd.c, 753
- KBD_KCMD_EK
 - ps2kbd.c, 753
- KBD_KCMD_RST
 - ps2kbd.c, 753
- KBD_KCMD_SRL
 - ps2kbd.c, 753
- KBD_KRESP_ACK
 - ps2kbd.c, 753
- KBD_KRESP_ECHO
 - ps2kbd.c, 753
- KBD_KRESP_RSTOK
 - ps2kbd.c, 753
- KBD_MAX_BUF
 - ps2kbd.c, 753
- kbd_put
 - ps2kbd.c, 754
- kbd_read8
 - ps2kbd.c, 754
- kbd_reset
 - ps2kbd.c, 754
- kbd_rxfile
 - ps2kbd.c, 754
- kbd_sec_end

- ps2kbd.c, 754
- kbd_sec_start
 - ps2kbd.c, 755
- KBD_SPACE
 - ps2kbd.c, 753
- kbd_state, 144
 - baseaddr, 144
 - buf, 144
 - buf_count, 144
 - buf_head, 144
 - buf_tail, 144
 - ccmd, 144
 - ccmdbyte, 144
 - enabled, 144
 - irq, 144
 - kcmd, 144
 - kresp, 144
 - rxfile, 144
 - rxfs, 144
 - slowdown, 144
- KBD_STATUS_A2
 - ps2kbd.c, 753
- KBD_STATUS_IBF
 - ps2kbd.c, 753
- KBD_STATUS_INH
 - ps2kbd.c, 753
- KBD_STATUS_MOBF
 - ps2kbd.c, 753
- KBD_STATUS_OBF
 - ps2kbd.c, 753
- KBD_STATUS_PERR
 - ps2kbd.c, 753
- KBD_STATUS_SYS
 - ps2kbd.c, 753
- KBD_STATUS_TO
 - ps2kbd.c, 753
- kbd_write8
 - ps2kbd.c, 755
- kcmd
 - kbd_state, 144
- KEEP_ENDIAN_LONG
 - coff.h, 249
- KEEP_ENDIAN_SHORT
 - coff.h, 250
- known
 - insn.c, 487
 - insn.h, 493
- kresp
 - kbd_state, 144
- l_addr
 - COFF_lineno, 46
- l_invalid
 - execute.c, 374
- l_inno
 - COFF_lineno, 46
- l_none
 - dyn32-defs.h, 357
 - or32.c, 434
- l_paddr
 - COFF_lineno, 46
- l_symndx
 - COFF_lineno, 46
- label_entry, 145
 - addr, 145
 - name, 145
 - next, 145
- label_hash
 - labels.c, 269
- LABELEND_CHAR
 - dumpverilog.c, 823
- LABELNAME_LEN
 - abstract.h, 231
- labels.c
 - add_breakpoint, 269
 - add_label, 269
 - breakpoints, 269
 - eval_label, 269
 - find_label, 269
 - get_label, 269
 - has_breakpoint, 269
 - init_breakpoints, 269
 - init_labels, 269
 - label_hash, 269
 - LABELS_HASH_SIZE, 269
 - print_breakpoints, 269
 - remove_breakpoint, 269
- labels.h
 - add_breakpoint, 271
 - add_label, 271
 - breakpoints, 271
 - eval_label, 271
 - find_label, 271
 - get_label, 271
 - has_breakpoint, 271
 - init_breakpoints, 271
 - init_labels, 271
 - print_breakpoints, 271
 - remove_breakpoint, 271
- LABELS_HASH_SIZE
 - labels.c, 269
- last
 - cuc_bb, 63
- last_used_reg
 - cuc_bb, 63
- last_way
 - ic, 127
- lba

- ata_device, 22
- lcr
 - dev_16450, 74
- LE16
 - abstract.h, 231
- len
 - rsp_buf, 162
- length
 - jtr_chain_message, 133
 - jtr_read_block_message, 136
 - jtr_read_message, 138
 - jtr_write_block_message, 140
 - jtr_write_message, 142
- letter_range
 - or32.c, 434
- letter_signed
 - or32.c, 434
- libtoplevel.c, 563
 - internal_orlksim_time, 564
 - orlksim_clock_rate, 564
 - orlksim_get_time_period, 564
 - orlksim_init, 564
 - orlksim_interrupt, 567
 - orlksim_is_le, 567
 - orlksim_reset_duration, 567
 - orlksim_run, 567
 - orlksim_set_time_point, 568
- line
 - dc_set, 70
- LINK_REGNO
 - arch.h, 290
- lo_buff
 - eth_device, 113
- load.c
 - build_insn, 495
 - conv, 498
 - cuc_load, 495
 - detect_locals, 496
 - expand_branch, 496
 - expand_calls, 496
 - expand_memory, 497
 - expand_signed, 497
 - insn, 498
 - negate_conditional, 497
 - num_insn, 498
 - print_cuc_insns, 497
 - reloc, 498
 - remove_dslots, 497
 - xchg_insn, 497
- load_hitdelay
 - config, 55
- load_missdelay
 - config, 55
- load_next_descriptor_when_done
 - dma_channel, 81
- load_timing_table
 - insn.h, 492
 - timings.c, 509
- loadcode
 - parse.c, 274
 - parse.h, 278
- loadcycles
 - runtime, 165
- loads_pagefaults
 - dmmustats_entry, 87
- loads_tlbhit
 - dmmustats_entry, 87
- loads_tlbmiss
 - dmmustats_entry, 87
- locs
 - dyn_page, 89
- log
 - cuc.h, 469
 - dev_memarea, 79
 - mem_config, 150
 - mem_ops, 152
- log2_int
 - misc.c, 827
 - misc.h, 828
- log_enabled
 - config, 55
- LONGEST
 - abstract.h, 231
- longlong_val
 - param_val, 159
- loopback
 - dev_16450, 74
- loopback_offset
 - eth_device, 113
- low32
 - common-i386.h, 325
- LRBB_REG
 - cuc.h, 469
- lru
 - bpb_entry, 30
 - btic_entry, 34
 - dc_set, 70
- lru_reload
 - dmmu, 86
 - immu, 129
- lrus
 - ic, 127
- LS_OP_CAST
 - op.c, 423
- LS_OP_FUNC
 - op.c, 423
- LS_OP_NAME
 - op.c, 423

- lsr
 - dev_16450, 74
- lur
 - _cuc_func, 16
- mac_address
 - eth_device, 113
- mac_int
 - eth_device, 113
- magic
 - COFF_AOUTHDR, 40
- main
 - dyngen.c, 363
 - generate.c, 378
 - toplevel-mprofile.c, 847
 - toplevel-profile.c, 850
 - toplevel.c, 862
- main_cuc
 - cuc.c, 457
 - cuc.h, 474
- main_list
 - insn.c, 487
- main_mprofiler
 - mprofiler.c, 588
 - mprofiler.h, 591
- main_profiler
 - profiler.c, 778
 - profiler.h, 781
- mainpage, 569
- mark_cut
 - cuc.h, 477
 - timings.c, 509
- mark_successors
 - adv.c, 441
- mask
 - adv.c, 441
 - spr_bit_def, 172
- masked_increase
 - dma.c, 693
- MATCHPOINTS_TO_NDP
 - spr-defs.h, 305
- MAX
 - cuc.h, 469
- max
 - cuc_insn, 65
- MAX_AUTOMATA_SIZE
 - or32.c, 432
- MAX_BB
 - cuc.h, 469
- max_bb_delay
 - timings.c, 509
- MAX_DC_BLOCK_SIZE
 - dcache-model.h, 196
- MAX_DC_SETS
 - dcache-model.h, 196
- MAX_DC_WAYS
 - dcache-model.h, 196
- max_delay
 - timings.c, 509
- MAX_FUNCS
 - profiler.h, 781
- MAX_GRP_S
 - spr-defs.h, 305
- MAX_IC_BLOCK_SIZE
 - icache-model.c, 199
- MAX_IC_SETS
 - icache-model.c, 199
- MAX_IC_WAYS
 - icache-model.c, 199
- MAX_INSNS
 - cuc.h, 469
- MAX_LEN
 - or32.c, 432
- MAX_MATCHPOINTS
 - spr-defs.h, 307
- max_op
 - adv.c, 441
- MAX_OP_TABLE_SIZE
 - or32.c, 432
- MAX_OPERANDS
 - abstract.h, 231
- MAX_PARAMS
 - dyngen.c, 363
- MAX_PREROLL
 - cuc.h, 469
- MAX_REGS
 - cuc.h, 469
- MAX_SBUF_LEN
 - sim-config.h, 810
- MAX_SKEW
 - 16450.c, 598
- MAX_SPRS
 - spr-defs.h, 307
- MAX_SPRS_PER_GRP
 - spr-defs.h, 307
- MAX_SPRS_PER_GRP_BITS
 - spr-defs.h, 307
- MAX_STACK
 - profiler.c, 778
- MAX_UNROLL
 - cuc.h, 469
- MAX_WATCHPOINTS
 - spr-defs.h, 307
- MAX_XTERM_ARGS
 - xterm.c, 676
- maximum_length
 - eth_device, 113
- maxstack

- profiler.c, 779
- mc, 146
 - ba_mask, 147
 - baseaddr, 147
 - csc, 147
 - csr, 147
 - enabled, 147
 - index, 147
 - mc_areas, 147
 - mc_area, 148
 - mem_config, 150
 - next, 147
 - poc, 147
 - tms, 147
- mc.c
 - MC_ADDR_SPACE, 742
 - mc_areas, 745
 - MC_BA_MASK, 742
 - MC_BA_MASK_VALID, 742
 - mc_baseaddr, 742
 - MC_CSC, 742
 - MC_CSC_BAS_OFFSET, 742
 - MC_CSC_BW_OFFSET, 742
 - MC_CSC_BW_WIDTH, 742
 - MC_CSC_EN_OFFSET, 742
 - MC_CSC_KRO_OFFSET, 742
 - MC_CSC_MEMTYPE_ASYNC, 742
 - MC_CSC_MEMTYPE_OFFSET, 742
 - MC_CSC_MEMTYPE_SDRAM, 742
 - MC_CSC_MEMTYPE_SSRAM, 742
 - MC_CSC_MEMTYPE_SYNC, 742
 - MC_CSC_MEMTYPE_WIDTH, 742
 - MC_CSC_MS_OFFSET, 742
 - MC_CSC_MS_WIDTH, 742
 - MC_CSC_PEN_OFFSET, 742
 - MC_CSC_SEL_OFFSET, 742
 - MC_CSC_SEL_WIDTH, 742
 - MC_CSC_VALID, 742
 - MC_CSC_WP_OFFSET, 742
 - MC_CSR, 742
 - MC_CSR_VALID, 742
 - mc_done, 742
 - mc_enabled, 742
 - mc_index, 743
 - MC_POC, 742
 - mc_poc, 743
 - MC_POC_EN_BW_OFFSET, 742
 - MC_POC_EN_BW_WIDTH, 742
 - MC_POC_EN_MEMTYPE_OFFSET, 742
 - MC_POC_EN_MEMTYPE_WIDTH, 742
 - MC_POC_VALID, 742
 - mc_read_word, 743
 - mc_reg_mem_area, 743
 - mc_reset, 743
 - mc_sec_end, 743
 - mc_sec_start, 744
 - mc_status, 744
 - MC_TMS, 742
 - MC_TMS_ASYNC_TRDV_OFFSET, 742
 - MC_TMS_ASYNC_TRDV_WIDTH, 742
 - MC_TMS_ASYNC_TRDZ_OFFSET, 742
 - MC_TMS_ASYNC_TRDZ_WIDTH, 742
 - MC_TMS_ASYNC_TWD_OFFSET, 742
 - MC_TMS_ASYNC_TWD_WIDTH, 742
 - MC_TMS_ASYNC_TWPW_OFFSET, 742
 - MC_TMS_ASYNC_TWPW_WIDTH, 742
 - MC_TMS_ASYNC_TWWD_OFFSET, 742
 - MC_TMS_ASYNC_TWWD_WIDTH, 742
 - MC_TMS_ASYNC_VALID, 742
 - MC_TMS_SDRAM_BL_OFFSET, 742
 - MC_TMS_SDRAM_BL_WIDTH, 742
 - MC_TMS_SDRAM_BT_OFFSET, 742
 - MC_TMS_SDRAM_CL_OFFSET, 742
 - MC_TMS_SDRAM_CL_WIDTH, 742
 - MC_TMS_SDRAM_OM_OFFSET, 742
 - MC_TMS_SDRAM_OM_WIDTH, 742
 - MC_TMS_SDRAM_TRCD_OFFSET, 742
 - MC_TMS_SDRAM_TRCD_WIDTH, 742
 - MC_TMS_SDRAM_TRFC_OFFSET, 742
 - MC_TMS_SDRAM_TRFC_WIDTH, 742
 - MC_TMS_SDRAM_TRP_OFFSET, 742
 - MC_TMS_SDRAM_TRP_WIDTH, 742
 - MC_TMS_SDRAM_TWR_OFFSET, 742
 - MC_TMS_SDRAM_TWR_WIDTH, 742
 - MC_TMS_SDRAM_VALID, 742
 - MC_TMS_SDRAM_WBL_OFFSET, 742
 - MC_TMS_SSRAM_VALID, 742
 - MC_TMS_SYNC_TRDV_OFFSET, 742
 - MC_TMS_SYNC_TRDV_WIDTH, 742
 - MC_TMS_SYNC_TRDZ_OFFSET, 742
 - MC_TMS_SYNC_TRDZ_WIDTH, 742
 - MC_TMS_SYNC_TTO_OFFSET, 742
 - MC_TMS_SYNC_TTO_WIDTH, 742
 - MC_TMS_SYNC_TWR_OFFSET, 742
 - MC_TMS_SYNC_TWR_WIDTH, 742
 - MC_TMS_SYNC_VALID, 742
 - MC_TMS_VALID, 742
 - mc_write_word, 744
 - mcs, 745
 - N_CE, 742
 - reg_mc_sec, 744
 - set_csc_tms, 745
- mc.h
 - mc_done, 746
 - mc_reg_mem_area, 746
 - reg_mc_sec, 747
 - MC_ADDR_SPACE
 - mc.c, 742

- mc_area, 148
 - abstract.c, 228
 - cs, 148
 - mc, 148
 - mem, 148
 - next, 148
- mc_areas
 - mc, 147
 - mc.c, 745
- MC_BA_MASK
 - mc.c, 742
- MC_BA_MASK_VALID
 - mc.c, 742
- mc_baseaddr
 - mc.c, 742
- MC_CSC
 - mc.c, 742
- MC_CSC_BAS_OFFSET
 - mc.c, 742
- MC_CSC_BW_OFFSET
 - mc.c, 742
- MC_CSC_BW_WIDTH
 - mc.c, 742
- MC_CSC_EN_OFFSET
 - mc.c, 742
- MC_CSC_KRO_OFFSET
 - mc.c, 742
- MC_CSC_MEMTYPE_ASYNC
 - mc.c, 742
- MC_CSC_MEMTYPE_OFFSET
 - mc.c, 742
- MC_CSC_MEMTYPE_SDRAM
 - mc.c, 742
- MC_CSC_MEMTYPE_SSRAM
 - mc.c, 742
- MC_CSC_MEMTYPE_SYNC
 - mc.c, 742
- MC_CSC_MEMTYPE_WIDTH
 - mc.c, 742
- MC_CSC_MS_OFFSET
 - mc.c, 742
- MC_CSC_MS_WIDTH
 - mc.c, 742
- MC_CSC_PEN_OFFSET
 - mc.c, 742
- MC_CSC_SEL_OFFSET
 - mc.c, 742
- MC_CSC_SEL_WIDTH
 - mc.c, 742
- MC_CSC_VALID
 - mc.c, 742
- MC_CSC_WP_OFFSET
 - mc.c, 742
- MC_CSR
 - mc.c, 742
- MC_CSR_VALID
 - mc.c, 742
- mc_done
 - mc.c, 742
 - mc.h, 746
- mc_enabled
 - mc.c, 742
- mc_index
 - mc.c, 743
- MC_POC
 - mc.c, 742
- mc_poc
 - mc.c, 743
- MC_POC_EN_BW_OFFSET
 - mc.c, 742
- MC_POC_EN_BW_WIDTH
 - mc.c, 742
- MC_POC_EN_MEMTYPE_OFFSET
 - mc.c, 742
- MC_POC_EN_MEMTYPE_WIDTH
 - mc.c, 742
- MC_POC_VALID
 - mc.c, 742
- mc_read_word
 - mc.c, 743
- mc_reg_mem_area
 - mc.c, 743
 - mc.h, 746
- mc_reset
 - mc.c, 743
- mc_sec_end
 - mc.c, 743
- mc_sec_start
 - mc.c, 744
- mc_status
 - mc.c, 744
- MC_TMS
 - mc.c, 742
- MC_TMS_ASYNC_TRDV_OFFSET
 - mc.c, 742
- MC_TMS_ASYNC_TRDV_WIDTH
 - mc.c, 742
- MC_TMS_ASYNC_TRDZ_OFFSET
 - mc.c, 742
- MC_TMS_ASYNC_TRDZ_WIDTH
 - mc.c, 742
- MC_TMS_ASYNC_TWD_OFFSET
 - mc.c, 742
- MC_TMS_ASYNC_TWD_WIDTH
 - mc.c, 742
- MC_TMS_ASYNC_TWPW_OFFSET
 - mc.c, 742
- MC_TMS_ASYNC_TWPW_WIDTH
 - mc.c, 742

- mc.c, [742](#)
- MC_TMS_ASYNC_TWWD_OFFSET
 - mc.c, [742](#)
- MC_TMS_ASYNC_TWWD_WIDTH
 - mc.c, [742](#)
- MC_TMS_ASYNC_VALID
 - mc.c, [742](#)
- MC_TMS_SDRAM_BL_OFFSET
 - mc.c, [742](#)
- MC_TMS_SDRAM_BL_WIDTH
 - mc.c, [742](#)
- MC_TMS_SDRAM_BT_OFFSET
 - mc.c, [742](#)
- MC_TMS_SDRAM_CL_OFFSET
 - mc.c, [742](#)
- MC_TMS_SDRAM_CL_WIDTH
 - mc.c, [742](#)
- MC_TMS_SDRAM_OM_OFFSET
 - mc.c, [742](#)
- MC_TMS_SDRAM_OM_WIDTH
 - mc.c, [742](#)
- MC_TMS_SDRAM_TRCD_OFFSET
 - mc.c, [742](#)
- MC_TMS_SDRAM_TRCD_WIDTH
 - mc.c, [742](#)
- MC_TMS_SDRAM_TRFC_OFFSET
 - mc.c, [742](#)
- MC_TMS_SDRAM_TRFC_WIDTH
 - mc.c, [742](#)
- MC_TMS_SDRAM_TRP_OFFSET
 - mc.c, [742](#)
- MC_TMS_SDRAM_TRP_WIDTH
 - mc.c, [742](#)
- MC_TMS_SDRAM_TWR_OFFSET
 - mc.c, [742](#)
- MC_TMS_SDRAM_TWR_WIDTH
 - mc.c, [742](#)
- MC_TMS_SDRAM_VALID
 - mc.c, [742](#)
- MC_TMS_SDRAM_WBL_OFFSET
 - mc.c, [742](#)
- MC_TMS_SSRAM_VALID
 - mc.c, [742](#)
- MC_TMS_SYNC_TRDV_OFFSET
 - mc.c, [742](#)
- MC_TMS_SYNC_TRDV_WIDTH
 - mc.c, [742](#)
- MC_TMS_SYNC_TRDZ_OFFSET
 - mc.c, [742](#)
- MC_TMS_SYNC_TRDZ_WIDTH
 - mc.c, [742](#)
- MC_TMS_SYNC_TTO_OFFSET
 - mc.c, [742](#)
- MC_TMS_SYNC_TTO_WIDTH
 - mc.c, [742](#)
- MC_TMS_SYNC_TWR_OFFSET
 - mc.c, [742](#)
- MC_TMS_SYNC_TWR_WIDTH
 - mc.c, [742](#)
- MC_TMS_SYNC_VALID
 - mc.c, [742](#)
- mc_write_word
 - mc.c, [744](#)
- mcr
 - dev_16450, [74](#)
- mcs
 - mc.c, [745](#)
- mdelay
 - runtime, [165](#)
- mdep
 - cuc_bb, [63](#)
- MEDIA_EJECT
 - atacmd.h, [624](#)
- MEDIA_LOCK
 - atacmd.h, [624](#)
- MEDIA_UNLOCK
 - atacmd.h, [624](#)
- mem
 - ata_host, [26](#)
 - ic, [127](#)
 - mc_area, [148](#)
 - mem_config, [150](#)
- mem_config
 - MT_PATTERN, [149](#)
 - MT_RANDOM, [149](#)
 - MT_UNKNOWN, [149](#)
- mem_config, [149](#)
 - baseaddr, [150](#)
 - ce, [150](#)
 - delayr, [150](#)
 - delayw, [150](#)
 - log, [150](#)
 - mc, [150](#)
 - mem, [150](#)
 - name, [150](#)
 - pattern, [150](#)
 - random_seed, [150](#)
 - size, [150](#)
 - type, [150](#)
- mem_cycles
 - runtime, [165](#)
- mem_ops, [151](#)
 - delayr, [152](#)
 - delayw, [152](#)
 - log, [152](#)
 - read_dat16, [152](#)

- read_dat32, 152
- read_dat8, 152
- readfunc16, 152
- readfunc32, 152
- readfunc8, 152
- write_dat16, 152
- write_dat32, 152
- write_dat8, 152
- writfunc16, 152
- writfunc32, 152
- writfunc8, 152
- writeprog32, 152
- writeprog32_dat, 152
- writeprog8, 152
- writeprog8_dat, 152
- mem_ordering_cmp
 - cuc/memory.c, 500
- mem_reset
 - peripheral/memory.c, 503
- memory.h
 - reg_memory_sec, 749
- memory_baseaddr
 - peripheral/memory.c, 503
- memory_ce
 - peripheral/memory.c, 503
- memory_delay
 - timings.c, 509
- memory_delayr
 - peripheral/memory.c, 503
- memory_delayw
 - peripheral/memory.c, 503
- memory_hash, 153
 - addr, 153
 - cnt, 153
 - next, 153
- MEMORY_LEN
 - parse.c, 273
- memory_log
 - peripheral/memory.c, 503
- memory_mc
 - peripheral/memory.c, 503
- memory_name
 - peripheral/memory.c, 503
- memory_order
 - _cuc_func, 16
 - config, 55
- memory_pattern
 - peripheral/memory.c, 503
- memory_random_seed
 - peripheral/memory.c, 503
- memory_sec_end
 - peripheral/memory.c, 504
- memory_sec_start
 - peripheral/memory.c, 504
- memory_size
 - peripheral/memory.c, 504
- memory_table_status
 - abstract.c, 220
 - abstract.h, 237
- memory_type
 - peripheral/memory.c, 504
- mfspr
 - sprs.c, 322
 - sprs.h, 323
- miiaddress
 - eth_device, 113
- miicommand
 - eth_device, 113
- miimoder
 - eth_device, 113
- miirx_data
 - eth_device, 113
- miistatus
 - eth_device, 113
- miitx_data
 - eth_device, 113
- MIN
 - 16450.c, 598
 - cuc.h, 469
 - or32.c, 432
- MIN_DC_BLOCK_SIZE
 - dcache-model.h, 196
- MIN_IC_BLOCK_SIZE
 - icache-model.c, 199
- MIN_MWDMA_CYCLE_TIME
 - atadevice-cmdi.h, 633
- MIN_PIO_CYCLE_TIME_IORDY
 - atadevice-cmdi.h, 633
- MIN_PIO_CYCLE_TIME_NO_IORDY
 - atadevice-cmdi.h, 633
- minimum_length
 - eth_device, 113
- misc.c
 - is_power2, 827
 - log2_int, 827
- misc.h
 - is_power2, 828
 - log2_int, 828
- miss
 - bpbstat, 31
 - bticstat, 35
- missdelay
 - config, 55
 - dmmu, 86
 - ic, 127
 - immu, 129
- mmu/dmmu.c, 570
- mmu/dmmu.h, 576

- mmu/immu.c, 579
- mmu/immu.h, 584
- MO_EXACT
 - cuc.h, 469
- MO_NONE
 - cuc.h, 469
- MO_STRONG
 - cuc.h, 469
- MO_WEAK
 - cuc.h, 469
- MODE_ACCESS
 - mprofiler.c, 588
- MODE_DETAIL
 - mprofiler.c, 588
- MODE_PRETTY
 - mprofiler.c, 588
- MODE_WIDTH
 - mprofiler.c, 588
- moder
 - eth_device, 113
- mp_entry, 154
 - addr, 154
 - instr, 154
 - next, 154
 - type, 154
- mp_hash
 - rsp-server.c, 559
- mp_hash_add
 - rsp-server.c, 544
- mp_hash_delete
 - rsp-server.c, 545
- mp_hash_init
 - rsp-server.c, 545
- mp_hash_lookup
 - rsp-server.c, 545
- MP_HASH_SIZE
 - rsp-server.c, 539
- mp_type
 - rsp-server.c, 540
- MPROF_16
 - profile.h, 831
- MPROF_32
 - profile.h, 831
- MPROF_8
 - profile.h, 831
- MPROF_FETCH
 - profile.h, 831
- mprof_fn
 - config, 55
- MPROF_READ
 - profile.h, 831
- MPROF_WRITE
 - profile.h, 831
- mprofentry_struct, 155
 - addr, 155
 - type, 155
- mprofile
 - config, 55
 - profile.c, 829
 - profile.h, 831
- mprofiler.c, 587
 - BUF_SIZE, 588
 - end_addr, 589
 - fprof, 589
 - group_bits, 589
 - hash, 589
 - hash_add, 588
 - HASH_FUNC, 588
 - hash_get, 588
 - HASH_SIZE, 588
 - init, 588
 - main_mprofiler, 588
 - MODE_ACCESS, 588
 - MODE_DETAIL, 588
 - MODE_PRETTY, 588
 - MODE_WIDTH, 588
 - nbits, 589
 - printout, 589
 - read_file, 589
 - start_addr, 589
- mprofiler.h, 591
 - main_mprofiler, 591
- msched
 - _cuc_func, 16
- msr
 - dev_16450, 74
- mstats_entry, 156
 - bf, 156
 - bnf, 156
 - bpb, 156
 - btic, 156
 - byteadd, 156
- MT_PATTERN
 - mem_config, 149
- MT_RANDOM
 - mem_config, 149
- MT_UNKNOWN
 - mem_config, 149
- MT_BURST
 - cuc.h, 469
- MT_BURSTE
 - cuc.h, 469
- MT_CALL
 - cuc.h, 469
- MT_LOAD
 - cuc.h, 469
- MT_SIGNED
 - cuc.h, 469

- MT_STORE
 - cuc.h, 469
- MT_WIDTH
 - cuc.h, 469
- mtspr
 - sprs.c, 322
 - sprs.h, 324
- mtype
 - _cuc_func, 16
- multissue
 - execute.c, 376
- mwdma
 - ata_device, 22
- N_CE
 - mc.c, 742
- n_descsz
 - elf32_note, 93
 - elf64_note, 102
- n_namesz
 - elf32_note, 93
 - elf64_note, 102
- n_type
 - elf32_note, 93
 - elf64_note, 102
- name
 - channel_factory, 38
 - config_param, 58
 - config_section, 59
 - cuc_known_insn, 66
 - dev_generic, 77
 - func_struct, 122
 - label_entry, 145
 - mem_config, 150
 - reloc, 161
 - sim_command, 169
 - spr_bit_def, 172
 - spr_def, 173
 - stack_struct, 175
 - tty.c, 673
- namein
 - file_channel, 120
- nameout
 - file_channel, 120
- nbits
 - mprofiler.c, 589
- ncolours
 - INFOHEADER, 131
- negate_conditional
 - cuc.h, 477
 - load.c, 497
- new_bb_cycles
 - timings.c, 509
- new_dp
 - dyn-rec.c, 348
 - dyn-rec.h, 354
- new_time
 - cuc_timings, 69
- next
 - _csm_list, 14
 - _dep_list_t, 17
 - breakpoint_entry, 33
 - channel_factory, 38
 - config_param, 58
 - config_section, 59
 - cuc_bb, 63
 - dev_memarea, 79
 - dma_controller, 84
 - gpio_device, 124
 - hist_exec, 125
 - label_entry, 145
 - mc, 147
 - mc_area, 148
 - memory_hash, 153
 - mp_entry, 154
 - op_queue, 158
 - sched_entry, 167
 - sim_reset_hook, 170
 - sim_stat, 171
 - vapi_handler, 178
- next_delay_insn
 - execute.c, 376
- nfdeps
 - _cuc_func, 16
- nfds
 - vapi.c, 870
- nfunccalls
 - profiler.c, 779
- nhandlers
 - vapi.c, 870
- ninsn
 - _csm_list, 14
 - cuc_bb, 63
- nmemory
 - cuc_bb, 63
- nmsched
 - _cuc_func, 16
- no_multicycle
 - config, 55
- nonblocking
 - tcp_channel, 176
- NOP
 - atacmd.h, 624
- NOP_CNT_RESET
 - spr-defs.h, 307
- NOP_EXIT
 - spr-defs.h, 307
- NOP_NOP

- spr-defs.h, 307
- NOP_PRINTF
 - spr-defs.h, 307
- NOP_PUTC
 - spr-defs.h, 307
- NOP_REPORT
 - spr-defs.h, 307
- NOP_REPORT_FIRST
 - spr-defs.h, 307
- NOP_REPORT_LAST
 - spr-defs.h, 307
- not_jump_loc
 - op_queue, 158
- nottaken
 - branchstat, 32
- NPC_REGNUM
 - rsp-server.c, 539
- nr_sect
 - ata_device, 22
- nsets
 - config, 55
 - dmmu, 86
 - ic, 127
 - immu, 129
- nshared
 - cuc_timings, 69
- nstack
 - profiler.c, 779
- NT_PRFPREG
 - elf.h, 260
- NT_PRPSINFO
 - elf.h, 260
- NT_PRSTATUS
 - elf.h, 260
- NT_TASKSTRUCT
 - elf.h, 260
- ntim
 - cuc_bb, 63
- ntotcalls
 - profiler.c, 779
- num_bb
 - _cuc_func, 16
- num_ids
 - vapi_handler, 178
- num_init_bb
 - _cuc_func, 16
- num_insn
 - cuc.h, 481
 - load.c, 498
- num_ones
 - or32.c, 434
- num_opcodes
 - or32.c, 435
- num_ops
 - op_queue, 158
- num_ops_param
 - op_queue, 158
- NUM_REGS
 - rsp-server.c, 540
- num_regs
 - jtr_read_block_message, 136
 - jtr_read_block_response, 137
 - jtr_write_block_message, 140
- num_runs
 - _cuc_func, 16
- NUM_T_REGS
 - i386-regs.h, 380
- nuncovered
 - or32.c, 435
- nways
 - config, 55
 - dmmu, 86
 - ic, 127
 - immu, 129
- oe
 - gpio_device, 124
- offset
 - BMP_HEADER, 29
 - eth_device, 113
- OP
 - op.c, 423
- op
 - cuc_insn, 65
- op-1t-op.h
 - glue, 393
- op-1t.h
 - OP_1T, 394
 - T, 394
- op-2t-op.h
 - glue, 395
- op-2t.h
 - OP_2T, 396
 - T, 396
- op-3t-op.h
 - glue, 397
- op-3t.h
 - OP_3T, 398
 - T, 398
- op-extend-op.h
 - glue, 401
- op-ff1-op.h
 - glue, 402
- op-i386.h
 - asm, 403
 - FORCE_RET, 403
 - OP_JUMP, 403
 - SPEEDY_CALL, 403

- op-mac-op.h
 - glue, 405
- op-mftspr-op.h
 - op_mtspr_imm_clear, 406
- op-support.c
 - do_jump, 408
 - op_support_analysis, 408
 - op_support_nop_exit, 408
 - op_support_nop_printf, 408
 - op_support_nop_report, 408
 - op_support_nop_report_imm, 408
 - op_support_nop_reset, 408
- op-support.h
 - do_jump, 409
 - op_support_analysis, 409
 - op_support_nop_exit, 409
 - op_support_nop_printf, 410
 - op_support_nop_report, 410
 - op_support_nop_report_imm, 410
 - op_support_nop_reset, 410
 - upd_reg_from_t, 410
- op-swhb-op.h
 - glue, 411
- op-t-reg-mov-op.h
 - glue, 415
- op.c
 - __op_param1, 429
 - __op_param2, 429
 - __op_param3, 429
 - __or_dynop, 423
 - asm, 423
 - COMP, 423
 - COMP_CAST, 423
 - COMP_NAME, 423
 - do_sched_wrap, 423
 - do_sched_wrap_delay, 424
 - enter_dyn_code, 424
 - EXT_CAST, 423
 - EXT_NAME, 423
 - EXT_TYPE, 423
 - LS_OP_CAST, 423
 - LS_OP_FUNC, 423
 - LS_OP_NAME, 423
 - OP, 423
 - op_add_pc, 424
 - op_analysis, 424
 - OP_CAST, 423
 - op_check_delay_slot, 424
 - op_check_flag, 424
 - op_check_flag_delay, 424
 - op_check_not_flag, 425
 - op_check_not_flag_delay, 425
 - op_clear_delay_insn, 425
 - op_clear_flag, 425
 - op_clear_pc_delay, 425
 - op_do_jump_delay, 425
 - op_do_sched, 425
 - op_do_sched_delay, 425
 - OP_EXTRA, 423
 - OP_FILE, 423
 - op_illegal, 425
 - op_illegal_delay, 427
 - op_jump_imm, 427
 - op_join_mem_cycles, 427
 - op_macc, 427
 - op_move_gpr10_pc_delay, 427
 - op_move_gpr11_pc_delay, 427
 - op_move_gpr12_pc_delay, 427
 - op_move_gpr13_pc_delay, 427
 - op_move_gpr14_pc_delay, 427
 - op_move_gpr15_pc_delay, 427
 - op_move_gpr16_pc_delay, 427
 - op_move_gpr17_pc_delay, 427
 - op_move_gpr18_pc_delay, 427
 - op_move_gpr19_pc_delay, 427
 - op_move_gpr1_pc_delay, 427
 - op_move_gpr20_pc_delay, 427
 - op_move_gpr21_pc_delay, 427
 - op_move_gpr22_pc_delay, 427
 - op_move_gpr23_pc_delay, 427
 - op_move_gpr24_pc_delay, 427
 - op_move_gpr25_pc_delay, 427
 - op_move_gpr26_pc_delay, 427
 - op_move_gpr27_pc_delay, 427
 - op_move_gpr28_pc_delay, 427
 - op_move_gpr29_pc_delay, 427
 - op_move_gpr2_pc_delay, 427
 - op_move_gpr30_pc_delay, 427
 - op_move_gpr31_pc_delay, 427
 - op_move_gpr3_pc_delay, 427
 - op_move_gpr4_pc_delay, 427
 - op_move_gpr5_pc_delay, 427
 - op_move_gpr6_pc_delay, 427
 - op_move_gpr7_pc_delay, 427
 - op_move_gpr8_pc_delay, 427
 - op_move_gpr9_pc_delay, 427
 - OP_NAME, 423
 - op_nop_exit, 427
 - op_nop_printf, 428
 - op_nop_report, 428
 - op_nop_report_imm, 428
 - op_nop_reset, 428
 - OP_PARAM1, 423
 - OP_PARAM2, 423
 - OP_PARAM3, 423
 - op_prep_rfe, 428
 - op_prep_sys, 428
 - op_prep_sys_delay, 428

- op_prep_trap, 429
- op_prep_trap_delay, 429
- op_set_delay_insn, 429
- op_set_flag, 429
- op_set_pc_delay_imm, 429
- op_set_pc_delay_pc, 429
- op_set_pc_pc_delay, 429
- op_store_insn_ea, 429
- op_store_link_addr_gpr, 429
- prep_except, 429
- S_FUNC, 423
- S_OP_NAME, 423
- save_t_bound, 429
- OP_1T
 - op-1t.h, 394
- OP_2T
 - op-2t.h, 396
- OP_3T
 - op-3t.h, 398
- op_add_pc
 - op.c, 424
- op_analysis
 - op.c, 424
- OP_CAST
 - op.c, 423
- op_check_delay_slot
 - op.c, 424
- op_check_flag
 - op.c, 424
- op_check_flag_delay
 - op.c, 424
- op_check_not_flag
 - op.c, 425
- op_check_not_flag_delay
 - op.c, 425
- op_clear_delay_insn
 - op.c, 425
- op_clear_flag
 - op.c, 425
- op_clear_pc_delay
 - op.c, 425
- op_data
 - or32.c, 435
- op_do_jump_delay
 - op.c, 425
- op_do_sched
 - op.c, 425
- op_do_sched_delay
 - op.c, 425
- OP_EXTRA
 - op.c, 423
- OP_FILE
 - op.c, 423
- OP_FUNC_PARAM_PREFIX
 - dyngen.c, 363
- OP_FUNC_PREFIX
 - dyngen.c, 363
- op_illegal
 - op.c, 425
- op_illegal_delay
 - op.c, 427
- op_jump_imm
 - op.c, 427
- op_join_mem_cycles
 - except.c, 293
 - op.c, 427
- OP_JUMP
 - op-i386.h, 403
- op_macc
 - op.c, 427
- OP_MEM_ACCESS
 - abstract.h, 231
- op_move_gpr10_pc_delay
 - op.c, 427
- op_move_gpr11_pc_delay
 - op.c, 427
- op_move_gpr12_pc_delay
 - op.c, 427
- op_move_gpr13_pc_delay
 - op.c, 427
- op_move_gpr14_pc_delay
 - op.c, 427
- op_move_gpr15_pc_delay
 - op.c, 427
- op_move_gpr16_pc_delay
 - op.c, 427
- op_move_gpr17_pc_delay
 - op.c, 427
- op_move_gpr18_pc_delay
 - op.c, 427
- op_move_gpr19_pc_delay
 - op.c, 427
- op_move_gpr1_pc_delay
 - op.c, 427
- op_move_gpr20_pc_delay
 - op.c, 427
- op_move_gpr21_pc_delay
 - op.c, 427
- op_move_gpr22_pc_delay
 - op.c, 427
- op_move_gpr23_pc_delay
 - op.c, 427
- op_move_gpr24_pc_delay
 - op.c, 427
- op_move_gpr25_pc_delay
 - op.c, 427
- op_move_gpr26_pc_delay
 - op.c, 427

- op_move_gpr27_pc_delay
 - op.c, 427
- op_move_gpr28_pc_delay
 - op.c, 427
- op_move_gpr29_pc_delay
 - op.c, 427
- op_move_gpr2_pc_delay
 - op.c, 427
- op_move_gpr30_pc_delay
 - op.c, 427
- op_move_gpr31_pc_delay
 - op.c, 427
- op_move_gpr3_pc_delay
 - op.c, 427
- op_move_gpr4_pc_delay
 - op.c, 427
- op_move_gpr5_pc_delay
 - op.c, 427
- op_move_gpr6_pc_delay
 - op.c, 427
- op_move_gpr7_pc_delay
 - op.c, 427
- op_move_gpr8_pc_delay
 - op.c, 427
- op_move_gpr9_pc_delay
 - op.c, 427
- op_mtspr_imm_clear
 - op-mftspr-op.h, 406
- OP_NAME
 - op.c, 423
- op_nop_exit
 - op.c, 427
- op_nop_printf
 - op.c, 428
- op_nop_report
 - op.c, 428
- op_nop_report_imm
 - op.c, 428
- op_nop_reset
 - op.c, 428
- OP_PARAM1
 - op.c, 423
- OP_PARAM2
 - op.c, 423
- OP_PARAM3
 - op.c, 423
- op_prep_rfe
 - op.c, 428
- op_prep_sys
 - op.c, 428
- op_prep_sys_delay
 - op.c, 428
- op_prep_trap
 - op.c, 429
- op_prep_trap_delay
 - op.c, 429
- op_queue, 157
 - insn, 158
 - insn_addr, 158
 - insn_index, 158
 - jump_local, 158
 - jump_local_loc, 158
 - next, 158
 - not_jump_loc, 158
 - num_ops, 158
 - num_ops_param, 158
 - ops, 158
 - ops_len, 158
 - ops_param, 158
 - ops_param_len, 158
 - param, 158
 - param_num, 158
 - param_type, 158
 - prev, 158
 - reg_t, 158
 - tflags, 158
 - xref, 158
- op_set_delay_insn
 - op.c, 429
- op_set_flag
 - op.c, 429
- op_set_pc_delay_imm
 - op.c, 429
- op_set_pc_delay_pc
 - op.c, 429
- op_set_pc_pc_delay
 - op.c, 429
- op_start
 - or32.c, 435
- op_store_insn_ea
 - op.c, 429
- op_store_link_addr_gpr
 - op.c, 429
- op_support_analysis
 - op-support.c, 408
 - op-support.h, 409
- op_support_nop_exit
 - op-support.c, 408
 - op-support.h, 409
- op_support_nop_printf
 - op-support.c, 408
 - op-support.h, 410
- op_support_nop_report
 - op-support.c, 408
 - op-support.h, 410
- op_support_nop_report_imm
 - op-support.c, 408
 - op-support.h, 410

- op_support_nop_reset
 - op-support.c, 408
 - op-support.h, 410
- open
 - channel_ops, 39
- open_file
 - atadevice.c, 637
- open_local
 - atadevice.c, 638
- open_obj
 - bff, 28
- OPERANDNAME_LEN
 - abstract.h, 231
- ops
 - channel, 37
 - channel_factory, 38
 - dev_memarea, 79
 - op_queue, 158
- OPS_ENLARGE_BY
 - dyn-rec.c, 335
- ops_len
 - op_queue, 158
- ops_param
 - op_queue, 158
- ops_param_len
 - op_queue, 158
- opt
 - cuc_insn, 65
- OPT_BB
 - cuc.h, 469
- OPT_CONST
 - cuc.h, 469
- OPT_DEST
 - cuc.h, 469
- OPT_JUMP
 - cuc.h, 469
- OPT_LRBB
 - cuc.h, 469
- OPT_NONE
 - cuc.h, 469
- OPT_REF
 - cuc.h, 469
- OPT_REGISTER
 - cuc.h, 469
- optimize_bb
 - bb.c, 445
 - cuc.h, 477
- optimize_cmov_more
 - insn.c, 485
- optimize_cmovs
 - cuc.h, 477
 - insn.c, 485
- optimize_tree
 - cuc.h, 477
- insn.c, 485
- option_char
 - cuc.c, 463
- options_cmd
 - cuc.c, 459
- OR1K_JTAG_COMMAND_CHAIN
 - gdb.h, 530
- OR1K_JTAG_COMMAND_READ
 - gdb.h, 530
- OR1K_JTAG_COMMAND_READ_BLOCK
 - gdb.h, 530
- OR1K_JTAG_COMMAND_WRITE
 - gdb.h, 530
- OR1K_JTAG_COMMAND_WRITE_BLOCK
 - gdb.h, 530
- or1k_jtag_errors
 - gdb.h, 529
- or1k_jtag_proxy_protocol_commands
 - gdb.h, 530
- OR1K_MEM_VERILOG_FOOTER
 - dumpverilog.c, 823
- OR1K_MEM_VERILOG_HEADER
 - dumpverilog.c, 823
- or1k_mstats
 - stats.c, 282
 - stats.h, 285
- OR1K_TRAP_INSTR
 - rsp-server.c, 540
- or1ksim.h, 592
 - OR1KSIM_RC_BADINIT, 592
 - OR1KSIM_RC_BRKPT, 592
 - OR1KSIM_RC_OK, 592
 - or1ksim_clock_rate, 592
 - or1ksim_get_time_period, 592
 - or1ksim_init, 593
 - or1ksim_interrupt, 593
 - or1ksim_is_le, 593
 - or1ksim_rc, 592
 - or1ksim_reset_duration, 593
 - or1ksim_run, 593
 - or1ksim_set_time_point, 594
- OR1KSIM_RC_BADINIT
 - or1ksim.h, 592
- OR1KSIM_RC_BRKPT
 - or1ksim.h, 592
- OR1KSIM_RC_OK
 - or1ksim.h, 592
- or1ksim_clock_rate
 - libtoplevel.c, 564
 - or1ksim.h, 592
- or1ksim_get_time_period
 - libtoplevel.c, 564
 - or1ksim.h, 592
- or1ksim_init

- libtoplevel.c, 564
- or1ksim.h, 593
- or1ksim_interrupt
 - libtoplevel.c, 567
 - or1ksim.h, 593
- or1ksim_is_le
 - libtoplevel.c, 567
 - or1ksim.h, 593
- or1ksim_rc
 - or1ksim.h, 592
- or1ksim_reset_duration
 - libtoplevel.c, 567
 - or1ksim.h, 593
- OR1KSIM_RSP_PROTOCOL
 - rsp-server.c, 540
- OR1KSIM_RSP_SERVICE
 - rsp-server.c, 540
- or1ksim_run
 - libtoplevel.c, 567
 - or1ksim.h, 593
- or1ksim_set_time_point
 - libtoplevel.c, 568
 - or1ksim.h, 594
- or32.c
 - automata, 435
 - build_automata, 432
 - cover_insn, 432
 - curpass, 435
 - destruct_automata, 432
 - disassemble_index, 433
 - disassemble_insn, 433
 - disassembled, 435
 - disassembled_str, 435
 - EF, 432
 - EFI, 432
 - EFN, 432
 - extend_imm, 433
 - insn_decode, 433
 - insn_extract, 434
 - insn_index, 434
 - insn_len, 434
 - insn_name, 434
 - l_none, 434
 - letter_range, 434
 - letter_signed, 434
 - MAX_AUTOMATA_SIZE, 432
 - MAX_LEN, 432
 - MAX_OP_TABLE_SIZE, 432
 - MIN, 432
 - num_ones, 434
 - num_opcodes, 435
 - nuncovered, 435
 - op_data, 435
 - op_start, 435
 - or32_debug, 434
 - or32_extract, 434
 - or32_letters, 435
 - or32_opcodes, 435
 - or32_print_immediate, 434
 - or32_print_register, 434
 - parse_params, 434
 - range_cache, 436
 - ti, 436
- or32_debug
 - or32.c, 434
- or32_extract
 - or32.c, 434
- or32_letters
 - or32.c, 435
- or32_opcodes
 - or32.c, 435
- or32_print_immediate
 - or32.c, 434
- or32_print_register
 - or32.c, 434
- or_page
 - dyn_page, 89
- oraddr_t
 - arch.h, 291
- orig_time
 - _cuc_func, 16
- orreg_t
 - arch.h, 291
- orsim_dbcl_set
 - debug.c, 817
- orsim_dbcl_set_name
 - debug.c, 817
 - debug.h, 820
- orsim_dbg_log
 - debug.c, 817
 - debug.h, 821
- osize
 - _csm_list, 14
- out
 - gpio_device, 124
- out_file
 - generate.c, 379
- out_lines
 - generate.c, 379
- output_call
 - generate.c, 378
- output_function
 - generate.c, 379
- output_verilog
 - verilog.c, 511
 - verilog.h, 513
- p_align

- elf32_phdr, [94](#)
- elf64_phdr, [103](#)
- p_filesz
 - elf32_phdr, [94](#)
 - elf64_phdr, [103](#)
- p_flags
 - elf32_phdr, [94](#)
 - elf64_phdr, [103](#)
- p_memsz
 - elf32_phdr, [94](#)
 - elf64_phdr, [103](#)
- p_offset
 - elf32_phdr, [94](#)
 - elf64_phdr, [103](#)
- p_paddr
 - elf32_phdr, [94](#)
 - elf64_phdr, [103](#)
- p_type
 - elf32_phdr, [94](#)
 - elf64_phdr, [103](#)
- p_vaddr
 - elf32_phdr, [94](#)
 - elf64_phdr, [103](#)
- PACKET
 - atacmd.h, [624](#)
- packet
 - ata_device, [22](#)
- packet_length
 - eth_device, [113](#)
- packetlen
 - eth_device, [113](#)
- page_mask
 - dmmu, [86](#)
 - immu, [129](#)
- page_offset_mask
 - dmmu, [86](#)
 - immu, [129](#)
- pagesize
 - dmmu, [86](#)
 - immu, [129](#)
- pagesize_log2
 - dmmu, [86](#)
 - immu, [129](#)
- pal
 - fb_state, [118](#)
- palette
 - vga_state, [180](#)
- param
 - op_queue, [158](#)
 - sched_entry, [167](#)
- param_num
 - op_queue, [158](#)
- param_t
 - sim-config.h, [810](#)
- param_type
 - op_queue, [158](#)
- param_val, [159](#)
 - addr_val, [159](#)
 - int_val, [159](#)
 - longlong_val, [159](#)
 - str_val, [159](#)
- PARAMS
 - dyn32-defs.h, [357](#)
 - simpl32-defs.h, [439](#)
- params
 - config_section, [59](#)
- paramt_addr
 - sim-config.h, [810](#)
- paramt_int
 - sim-config.h, [810](#)
- paramt_longlong
 - sim-config.h, [810](#)
- paramt_none
 - sim-config.h, [810](#)
- paramt_str
 - sim-config.h, [810](#)
- paramt_word
 - sim-config.h, [810](#)
- parse.c
 - addprogram, [273](#)
 - freemem, [277](#)
 - identifyfile, [273](#)
 - IMM_STATS, [273](#)
 - loadcode, [274](#)
 - MEMORY_LEN, [273](#)
 - readfile_coff, [275](#)
 - readfile_elf, [275](#)
 - readsyms_coff, [276](#)
 - rstrip, [276](#)
 - transl_error, [277](#)
 - transl_table, [277](#)
 - translate, [276](#)
- parse.h
 - loadcode, [278](#)
 - rstrip, [279](#)
- parse_args
 - sim-config.c, [802](#)
 - sim-config.h, [811](#)
- parse_baud
 - tty.c, [673](#)
- parse_dbchs
 - debug.c, [817](#)
 - debug.h, [821](#)
- parse_params
 - or32.c, [434](#)
- pattern
 - mem_config, [150](#)
- pc

- cpu_state, 61
- pc_delay
 - cpu_state, 61
- pcnext
 - execute.c, 376
 - execute.h, 267
- pctr
 - ata_host, 26
- pdiagi
 - ata_device, 22
- pdiago
 - ata_device, 22
- peek_into_dtlb
 - dmmu.c, 574
 - dmmu.h, 577
- peek_into_itlb
 - immu.c, 583
 - immu.h, 585
- peripheral/16450.c, 595
- peripheral/16450.h, 617
- peripheral/atacmd.h, 621
- peripheral/atadevice-cmdi.c, 625
- peripheral/atadevice-cmdi.h, 630
- peripheral/atadevice.c, 635
- peripheral/atadevice.h, 639
- peripheral/atahost.c, 644
- peripheral/atahost.h, 654
- peripheral/channels/channel.c, 660
- peripheral/channels/channel.h, 662
- peripheral/channels/fd.c, 663
- peripheral/channels/fd.h, 665
- peripheral/channels/file.c, 666
- peripheral/channels/file.h, 668
- peripheral/channels/generic.c, 727
- peripheral/channels/generic.h, 729
- peripheral/channels/tcp.c, 669
- peripheral/channels/tcp.h, 671
- peripheral/channels/tty.c, 672
- peripheral/channels/tty.h, 674
- peripheral/channels/xterm.c, 675
- peripheral/channels/xterm.h, 677
- peripheral/crc32.c, 678
- peripheral/crc32.h, 679
- peripheral/dma-defs.h, 680
- peripheral/dma.c, 687
- peripheral/dma.h, 694
- peripheral/eth.c, 696
- peripheral/eth.h, 711
- peripheral/fb.c, 713
- peripheral/fb.h, 719
- peripheral/fields.h, 720
- peripheral/generic.c, 722
- peripheral/generic.h, 728
- peripheral/gpio.c, 730
- peripheral/gpio.h, 737
- peripheral/mc.c, 738
- peripheral/mc.h, 746
- peripheral/memory.c, 502
 - mem_reset, 503
 - memory_baseaddr, 503
 - memory_ce, 503
 - memory_delayr, 503
 - memory_delayw, 503
 - memory_log, 503
 - memory_mc, 503
 - memory_name, 503
 - memory_pattern, 503
 - memory_random_seed, 503
 - memory_sec_end, 504
 - memory_sec_start, 504
 - memory_size, 504
 - memory_type, 504
 - reg_memory_sec, 505
 - simmem_read16, 505
 - simmem_read32, 506
 - simmem_read8, 506
 - simmem_read_zero16, 506
 - simmem_read_zero32, 506
 - simmem_read_zero8, 506
 - simmem_write16, 506
 - simmem_write32, 506
 - simmem_write8, 506
 - simmem_write_null16, 506
 - simmem_write_null32, 506
 - simmem_write_null8, 506
- peripheral/memory.h, 748
- peripheral/ps2kbd.c, 750
- peripheral/ps2kbd.h, 757
- peripheral/vga.c, 758
- peripheral/vga.h, 764
- PF_R
 - elf.h, 260
- PF_W
 - elf.h, 260
- PF_X
 - elf.h, 260
- pftr0
 - ata_host, 26
- pftr1
 - ata_host, 26
- pic
 - config, 55
 - fb_state, 118
- pic.c
 - clear_interrupt, 766
 - pic_edge_trigger, 766
 - pic_enabled, 766
 - pic_ints_en, 766

- pic_rep_int, 766
- pic_reset, 767
- pic_state, 768
- pic_state_int, 768
- reg_pic_sec, 767
- report_interrupt, 767
- pic.h
 - clear_interrupt, 769
 - pic_ints_en, 769
 - pic_reset, 769
 - reg_pic_sec, 769
 - report_interrupt, 770
- pic/pic.c, 765
- pic/pic.h, 769
- pic_edge_trigger
 - pic.c, 766
- pic_enabled
 - pic.c, 766
- pic_ints_en
 - pic.c, 766
 - pic.h, 769
- pic_lines
 - cpu_state, 61
- pic_rep_int
 - pic.c, 766
- pic_reset
 - pic.c, 767
 - pic.h, 769
- pic_state
 - pic.c, 768
- pic_state_int
 - pic.c, 768
- pics
 - vga_state, 180
- pid
 - xterm_channel, 181
- pinindex
 - vga_state, 180
- pio
 - ata_device, 22
- pio_mode
 - ata_device, 22
- PIO_MODE0_T1
 - atahost.c, 646
- pio_mode0_t1
 - ata_host, 26
- PIO_MODE0_T2
 - atahost.c, 646
- pio_mode0_t2
 - ata_host, 26
- PIO_MODE0_T4
 - atahost.c, 646
- pio_mode0_t4
 - ata_host, 26
- PIO_MODE0_TEOC
 - atahost.c, 646
- pio_mode0_teoc
 - ata_host, 26
- planes
 - INFOHEADER, 131
- pm
 - config, 55
- pm.c
 - pm_enabled, 771
 - pm_reset, 771
 - reg_pm_sec, 771
- pm.h
 - pm_reset, 773
 - reg_pm_sec, 773
- pm/pm.c, 771
- pm/pm.h, 773
- pm_enabled
 - pm.c, 771
- pm_reset
 - pm.c, 771
 - pm.h, 773
- poc
 - mc, 147
- port.h
 - isblank, 775
 - PRi16, 775
 - PRi8, 775
 - strndup, 775
- port/isblank.c, 774
- port/port.h, 775
- port/strndup.c, 776
- port_number
 - tcp_channel, 176
- POWERUP_IN_STANDBY_FEATURE_SET_-
SPINUP
 - atacmd.h, 624
- PPC_REGNUM
 - rsp-server.c, 540
- preloaded
 - channel.c, 661
- prep_except
 - op.c, 429
- preroll
 - cuc_timings, 69
- preunroll_bb
 - cuc.c, 459
- preunroll_loop
 - bb.c, 445
 - cuc.h, 478
- prev
 - cuc_bb, 63
 - hist_exec, 125
 - op_queue, 158

- PRIdREG
 - arch.h, 290
- print_bb_num
 - bb.c, 445
 - cuc.h, 478
- print_breakpoints
 - labels.c, 269
 - labels.h, 271
- print_config
 - sim-config.c, 802
 - sim-config.h, 811
- print_cuc_bb
 - bb.c, 446
 - cuc.h, 478
- print_cuc_insns
 - cuc.h, 478
 - load.c, 497
- print_deps
 - verilog.c, 511
- print_insn_exec
 - sim-cmd.c, 786
- print_insn_v
 - verilog.c, 512
- print_insns
 - cuc.h, 479
 - insn.c, 486
- print_op_v
 - verilog.c, 512
- print_option
 - cuc.c, 462
- print_shared
 - insn.c, 486
 - insn.h, 492
- print_turn_off_dep
 - verilog.c, 512
- PRINTF
 - sim-config.h, 810
- prntotherstats
 - stats.c, 281
- printout
 - mprofiler.c, 589
- printstats
 - stats.c, 281
 - stats.h, 285
- priv_dat
 - vapi_handler, 178
- PRiX16
 - port.h, 775
- PRiX8
 - port.h, 775
- PRiXADDR
 - arch.h, 290
- PRiXREG
 - arch.h, 291
- prof_acquire
 - profiler.c, 778
 - profiler.h, 782
- prof_cycles
 - profiler.c, 779
 - profiler.h, 782
- prof_fn
 - config, 55
- prof_func
 - profiler.c, 779
 - profiler.h, 782
- prof_nfuncs
 - profiler.c, 779
 - profiler.h, 782
- prof_print
 - profiler.c, 778
- prof_set
 - profiler.c, 779
 - profiler.h, 782
- profile
 - config, 55
- profile.c
 - mprofile, 829
- profile.h
 - MPROF_16, 831
 - MPROF_32, 831
 - MPROF_8, 831
 - MPROF_FETCH, 831
 - MPROF_READ, 831
 - MPROF_WRITE, 831
 - mprofile, 831
- profiler.c, 777
 - cumulative, 779
 - fprof, 779
 - main_profiler, 778
 - MAX_STACK, 778
 - maxstack, 779
 - nfuncalls, 779
 - nstack, 779
 - ntotcalls, 779
 - prof_acquire, 778
 - prof_cycles, 779
 - prof_func, 779
 - prof_nfuncs, 779
 - prof_print, 778
 - prof_set, 779
 - quiet, 779
 - stack, 779
- profiler.h, 781
 - main_profiler, 781
 - MAX_FUNCS, 781
 - prof_acquire, 782
 - prof_cycles, 782
 - prof_func, 782

- prof_nfuncs, 782
- prof_set, 782
- proto_num
 - rsp-server.c, 559
- protocol_clean
 - gdbcomm.c, 534
- ps2kbd.c
 - code, 756
 - kbd_baseaddr, 753
 - KBD_BAUD_RATE, 753
 - KBD_CCMD_DKI, 753
 - KBD_CCMD_EKI, 753
 - KBD_CCMD_RCB, 753
 - KBD_CCMD_ST1, 753
 - KBD_CCMD_ST2, 753
 - KBD_CCMD_WCB, 753
 - KBD_CCMDBYTE_EN, 753
 - KBD_CCMDBYTE_EN2, 753
 - KBD_CCMDBYTE_INT, 753
 - KBD_CCMDBYTE_INT2, 753
 - KBD_CCMDBYTE_SYS, 753
 - KBD_CCMDBYTE_XLAT, 753
 - KBD_CTRL, 753
 - KBD_DATA, 753
 - kbd_enabled, 753
 - kbd_info, 753
 - kbd_irq, 753
 - kbd_job, 753
 - KBD_KCMD_DK, 753
 - KBD_KCMD_ECHO, 753
 - KBD_KCMD_EK, 753
 - KBD_KCMD_RST, 753
 - KBD_KCMD_SRL, 753
 - KBD_KRESP_ACK, 753
 - KBD_KRESP_ECHO, 753
 - KBD_KRESP_RSTOK, 753
 - KBD_MAX_BUF, 753
 - kbd_put, 754
 - kbd_read8, 754
 - kbd_reset, 754
 - kbd_rxfile, 754
 - kbd_sec_end, 754
 - kbd_sec_start, 755
 - KBD_SPACE, 753
 - KBD_STATUS_A2, 753
 - KBD_STATUS_IBF, 753
 - KBD_STATUS_INH, 753
 - KBD_STATUS_MOBF, 753
 - KBD_STATUS_OBF, 753
 - KBD_STATUS_PERR, 753
 - KBD_STATUS_SYS, 753
 - KBD_STATUS_TO, 753
 - kbd_write8, 755
 - reg_kbd_sec, 755
 - scan_decode, 756
 - scan_table, 756
 - shift, 756
- ps2kbd.h
 - reg_kbd_sec, 757
- PT_DYNAMIC
 - elf.h, 260
- PT_HIPROC
 - elf.h, 260
- PT_INTERP
 - elf.h, 260
- PT_LOAD
 - elf.h, 260
- PT_LOPROC
 - elf.h, 260
- PT_NOTE
 - elf.h, 260
- PT_NULL
 - elf.h, 260
- PT_PHDR
 - elf.h, 260
- PT_SHLIB
 - elf.h, 260
- pdrig
 - gpio_device, 124
- put_packet
 - rsp-server.c, 545
- put_rsp_char
 - rsp-server.c, 546
- put_str_packet
 - rsp-server.c, 546
- QUEUE_DEPTH
 - atadevice-cmdi.h, 633
- quiet
 - profiler.c, 779
- R_386_32
 - elf.h, 260
- R_386_COPY
 - elf.h, 260
- R_386_GLOB_DAT
 - elf.h, 260
- R_386_GOT32
 - elf.h, 260
- R_386_GOTOFF
 - elf.h, 260
- R_386_GOTPC
 - elf.h, 260
- R_386_JMP_SLOT
 - elf.h, 260
- R_386_NONE
 - elf.h, 260
- R_386_NUM

- elf.h, 260
- R_386_PC32
 - elf.h, 260
- R_386_PLT32
 - elf.h, 260
- R_386_RELATIVE
 - elf.h, 260
- R_68K_16
 - elf.h, 260
- R_68K_32
 - elf.h, 260
- R_68K_8
 - elf.h, 260
- R_68K_COPY
 - elf.h, 260
- R_68K_GLOB_DAT
 - elf.h, 260
- R_68K_GOT16
 - elf.h, 260
- R_68K_GOT16O
 - elf.h, 260
- R_68K_GOT32
 - elf.h, 260
- R_68K_GOT32O
 - elf.h, 260
- R_68K_GOT8
 - elf.h, 260
- R_68K_GOT8O
 - elf.h, 260
- R_68K_JMP_SLOT
 - elf.h, 260
- R_68K_NONE
 - elf.h, 260
- R_68K_PC16
 - elf.h, 260
- R_68K_PC32
 - elf.h, 260
- R_68K_PC8
 - elf.h, 260
- R_68K_PLT16
 - elf.h, 260
- R_68K_PLT16O
 - elf.h, 260
- R_68K_PLT32
 - elf.h, 260
- R_68K_PLT32O
 - elf.h, 260
- R_68K_PLT8
 - elf.h, 260
- R_68K_PLT8O
 - elf.h, 260
- R_68K_RELATIVE
 - elf.h, 260
- r_addend
 - elf32_rela, 96
 - elf64_rela, 105
- r_info
 - elf32_rel, 95
 - elf32_rela, 96
 - elf64_rel, 104
 - elf64_rela, 105
- r_offset
 - elf32_rel, 95
 - elf32_rela, 96
 - elf64_rel, 104
 - elf64_rela, 105
- r_symndx
 - COFF_reloc, 47
- r_type
 - COFF_reloc, 47
- r_vaddr
 - COFF_reloc, 47
- raddr
 - stack_struct, 175
- random_seed
 - mem_config, 150
- range
 - raw_stats, 160
- range_cache
 - or32.c, 436
- RAW_RANGE
 - stats.h, 285
- raw_stats, 160
 - range, 160
 - reg, 160
 - stats.c, 283
 - stats.h, 285
- read
 - channel_ops, 39
- READ_BUFFER
 - atacmd.h, 624
- read_dat16
 - mem_ops, 152
- read_dat32
 - mem_ops, 152
- read_dat8
 - mem_ops, 152
- READ_DMA
 - atacmd.h, 624
- READ_DMA_QUEUED
 - atacmd.h, 624
- read_file
 - mprofiler.c, 589
- read_func
 - vapi_handler, 178
- READ_MULTIPLE
 - atacmd.h, 624
- READ_NATIVE_MAX_ADDRESS

- atacmd.h, 624
- read_packet
 - vapi.c, 867
- read_script_file
 - sim-config.c, 803
- READ_SECTOR
 - atacmd.h, 624
- READ_SECTORS
 - atacmd.h, 624
- read_up
 - config, 55
- READ_VERIFY_SECTOR
 - atacmd.h, 624
- READ_VERIFY_SECTORS
 - atacmd.h, 624
- readfile_coff
 - parse.c, 275
- readfile_elf
 - parse.c, 275
- readfunc16
 - mem_ops, 152
- readfunc32
 - mem_ops, 152
- readfunc8
 - mem_ops, 152
- readhit
 - cachestats_entry, 36
- readmiss
 - cachestats_entry, 36
- readsyms_coff
 - parse.c, 276
- rebuild_fds
 - vapi.c, 867
- rec-i386.h
 - get_sp, 437
- rec_stack_base
 - dyn-rec.h, 356
- recalc_cnts
 - cuc.h, 479
 - timings.c, 509
- recalc_last_used_reg
 - bb.c, 446
- RECED_PAGE_ENLARGE_BY
 - dyn-rec.c, 335
- receiveing
 - dev_16450, 74
- recheck_immu
 - dyn-rec.c, 348
 - dyn-rec.h, 355
- RECOMMENDED_MWDMA_CYCLE_TIME
 - atadevice-cmdi.h, 633
- recompile_delay_insn
 - dyn-rec.c, 348
- recompile_insn
 - dyn-rec.c, 349
- recompile_page
 - dyn-rec.c, 349
 - dyn-rec.h, 355
- recv_break
 - dev_16450, 74
- reenter_int
 - sim-cmd.c, 787
- REF
 - cuc.h, 469
- ref
 - _csm_list, 14
 - _dep_list_t, 17
 - cuc_shared_item, 67
- REF_BB
 - cuc.h, 469
- REF_I
 - cuc.h, 469
- referenced
 - dma_channel, 81
- refresh
 - fb_state, 118
- refresh_count
 - fb_state, 118
- REFRESH_DIVIDER
 - fb.c, 714
- refresh_rate
 - fb_state, 118
 - vga_state, 180
- reg
 - cpu_state, 60
 - raw_stats, 160
- reg2hex
 - rsp-server.c, 546
- reg_ata_sec
 - atahost.c, 650
 - atahost.h, 658
- reg_bpb_sec
 - branch-predict.c, 186
 - branch-predict.h, 189
- REG_C
 - arch.h, 291
- reg_config_param
 - sim-config.c, 803
 - sim-config.h, 811
- reg_config_sec
 - sim-config.c, 805
 - sim-config.h, 813
- reg_config_secs
 - sim-config.c, 805
 - sim-config.h, 813
- reg_cpu_sec
 - cpu-config.c, 208
 - cpu-config.h, 210

- reg_cuc_sec
 - cuc.c, [462](#)
 - cuc.h, [479](#)
- reg_dc_sec
 - dcache-model.c, [193](#)
 - dcache-model.h, [197](#)
- reg_debug_sec
 - debug-unit.c, [521](#)
 - debug-unit.h, [527](#)
- reg_dep
 - bb.c, [446](#)
 - cuc.h, [479](#)
- reg_dep_rec
 - bb.c, [446](#)
- reg_dma_sec
 - dma.c, [693](#)
 - dma.h, [695](#)
- reg_dmmu_sec
 - dmmu.c, [574](#)
 - dmmu.h, [577](#)
- reg_ethernet_sec
 - eth.c, [709](#)
 - eth.h, [711](#)
- reg_fb_sec
 - fb.c, [717](#)
 - fb.h, [719](#)
- reg_generic_sec
 - generic.c, [725](#)
 - generic.h, [728](#)
- reg_gpio_sec
 - gpio.c, [736](#)
 - gpio.h, [737](#)
- reg_ic_sec
 - icache-model.c, [202](#)
 - icache-model.h, [204](#)
- reg_immu_sec
 - immu.c, [583](#)
 - immu.h, [585](#)
- reg_kbd_sec
 - ps2kbd.c, [755](#)
 - ps2kbd.h, [757](#)
- reg_mc_sec
 - mc.c, [744](#)
 - mc.h, [747](#)
- reg_mem_area
 - abstract.c, [220](#)
 - abstract.h, [237](#)
- reg_memory_sec
 - memory.h, [749](#)
 - peripheral/memory.c, [505](#)
- reg_pic_sec
 - pic.c, [767](#)
 - pic.h, [769](#)
- reg_pm_sec
 - pm.c, [771](#)
 - pm.h, [773](#)
- reg_sim_reset
 - toplevel-support.c, [853](#)
 - toplevel-support.h, [858](#)
- reg_sim_sec
 - sim-config.c, [806](#)
- reg_sim_stat
 - sim-cmd.c, [787](#)
 - sim-cmd.h, [799](#)
- reg_t
 - op_queue, [158](#)
- reg_uart_sec
 - 16450.c, [604](#)
 - 16450.h, [619](#)
- reg_vapi_sec
 - vapi.c, [867](#)
 - vapi.h, [872](#)
- reg_vga_sec
 - vga.c, [760](#)
 - vga.h, [764](#)
- register_memoryarea_mask
 - abstract.c, [221](#)
- regs
 - ata_device, [22](#)
 - ata_host, [26](#)
 - dev_16450, [74](#)
 - dma_channel, [81](#)
 - dma_controller, [84](#)
 - eth_device, [113](#)
- reloc, [161](#)
 - addend, [161](#)
 - cuc.h, [481](#)
 - func_offset, [161](#)
 - load.c, [498](#)
 - name, [161](#)
 - type, [161](#)
- relocate_bb
 - bb.c, [446](#)
- remove_breakpoint
 - labels.c, [269](#)
 - labels.h, [271](#)
- remove_dead
 - cuc.h, [480](#)
 - insn.c, [486](#)
- remove_dead_bb
 - bb.c, [446](#)
 - cuc.h, [480](#)
- remove_dslots
 - load.c, [497](#)
- remove_nops
 - cuc.h, [480](#)
 - insn.c, [486](#)
- remove_trivial_regs

- cuc.h, 480
- insn.c, 487
- report_interrupt
 - pic.c, 767
 - pic.h, 770
- reserved1
 - BMP_HEADER, 29
- reserved2
 - BMP_HEADER, 29
- reset_cycles
 - runtime, 165
- reset_hook
 - sim_reset_hook, 170
- reset_instructions
 - runtime, 165
- RET_OPCODE
 - dyngen-i386.c, 360
- ret_spr
 - spr-dump.c, 309
- rev
 - ata_host, 26
- rfd
 - eth_device, 113
- RGPIO_AUX
 - gpio.c, 732
- RGPIO_CTRL
 - gpio.c, 732
- RGPIO_CTRL_ECLK
 - gpio.c, 732
- RGPIO_CTRL_INTE
 - gpio.c, 732
- RGPIO_CTRL_INTS
 - gpio.c, 732
- RGPIO_CTRL_NEC
 - gpio.c, 732
- RGPIO_IN
 - gpio.c, 732
- RGPIO_INTE
 - gpio.c, 732
- RGPIO_INTS
 - gpio.c, 732
- RGPIO_OE
 - gpio.c, 732
- RGPIO_OUT
 - gpio.c, 732
- RGPIO_PTRIG
 - gpio.c, 732
- RISCOP_RESET
 - debug-unit.c, 515
- RISCOP_STALL
 - debug-unit.c, 515
- roll_loop
 - bb.c, 446
- rsp
 - rsp-server.c, 559
- rsp-server.c
 - ascii2hex, 541
 - BP_HARDWARE, 540
 - BP_MEMORY, 540
 - client_fd, 559
 - client_waiting, 559
 - GDB_BUF_MAX, 539
 - get_packet, 541
 - get_rsp_char, 541
 - handle_rsp, 542
 - hex, 543
 - hex2ascii, 544
 - hex2reg, 544
 - hexchars, 559
 - mp_hash, 559
 - mp_hash_add, 544
 - mp_hash_delete, 545
 - mp_hash_init, 545
 - mp_hash_lookup, 545
 - MP_HASH_SIZE, 539
 - mp_type, 540
 - NPC_REGNUM, 539
 - NUM_REGS, 540
 - OR1K_TRAP_INSTR, 540
 - OR1KSIM_RSP_PROTOCOL, 540
 - OR1KSIM_RSP_SERVICE, 540
 - PPC_REGNUM, 540
 - proto_num, 559
 - put_packet, 545
 - put_rsp_char, 546
 - put_str_packet, 546
 - reg2hex, 546
 - rsp, 559
 - rsp_client_close, 547
 - rsp_client_request, 547
 - rsp_command, 548
 - rsp_continue, 549
 - rsp_continue_generic, 549
 - rsp_continue_with_signal, 550
 - rsp_exception, 550
 - rsp_init, 550
 - rsp_insert_matchpoint, 551
 - rsp_query, 551
 - rsp_read_all_regs, 552
 - rsp_read_mem, 552
 - rsp_read_reg, 553
 - rsp_remove_matchpoint, 553
 - rsp_report_exception, 554
 - rsp_restart, 554
 - rsp_server_close, 554
 - rsp_server_request, 554
 - rsp_set, 555
 - rsp_step, 555

- rsp_step_generic, 555
- rsp_step_with_signal, 556
- RSP_TRACE, 540
- rsp_unescape, 556
- rsp_vpkt, 556
- rsp_write_all_regs, 557
- rsp_write_mem, 557
- rsp_write_mem_bin, 557
- rsp_write_reg, 558
- server_fd, 559
- set_npc, 558
- sigval, 559
- SR_REGNUM, 540
- start_addr, 559
- TARGET_SIGNAL_ALRM, 541
- TARGET_SIGNAL_BUS, 541
- TARGET_SIGNAL_FPE, 541
- TARGET_SIGNAL_ILL, 541
- TARGET_SIGNAL_INT, 541
- TARGET_SIGNAL_NONE, 541
- TARGET_SIGNAL_PWR, 541
- TARGET_SIGNAL_SEGV, 541
- TARGET_SIGNAL_TRAP, 541
- TARGET_SIGNAL_USR2, 541
- target_signal, 540
- WP_ACCESS, 540
- WP_READ, 540
- WP_WRITE, 540
- rsp-server.h
 - handle_rsp, 560
 - rsp_exception, 561
 - rsp_init, 562
- rsp_buf, 162
 - data, 162
 - len, 162
- rsp_client_close
 - rsp-server.c, 547
- rsp_client_request
 - rsp-server.c, 547
- rsp_command
 - rsp-server.c, 548
- rsp_continue
 - rsp-server.c, 549
- rsp_continue_generic
 - rsp-server.c, 549
- rsp_continue_with_signal
 - rsp-server.c, 550
- rsp_enabled
 - config, 55
- rsp_exception
 - rsp-server.c, 550
 - rsp-server.h, 561
- rsp_init
 - rsp-server.c, 550
 - rsp-server.h, 562
- rsp_insert_matchpoint
 - rsp-server.c, 551
- rsp_port
 - config, 55
- rsp_query
 - rsp-server.c, 551
- rsp_read_all_regs
 - rsp-server.c, 552
- rsp_read_mem
 - rsp-server.c, 552
- rsp_read_reg
 - rsp-server.c, 553
- rsp_remove_matchpoint
 - rsp-server.c, 553
- rsp_report_exception
 - rsp-server.c, 554
- rsp_restart
 - rsp-server.c, 554
- rsp_server_close
 - rsp-server.c, 554
- rsp_server_request
 - rsp-server.c, 554
- rsp_set
 - rsp-server.c, 555
- rsp_step
 - rsp-server.c, 555
- rsp_step_generic
 - rsp-server.c, 555
- rsp_step_with_signal
 - rsp-server.c, 556
- RSP_TRACE
 - rsp-server.c, 540
- rsp_unescape
 - rsp-server.c, 556
- rsp_vpkt
 - rsp-server.c, 556
- rsp_write_all_regs
 - rsp-server.c, 557
- rsp_write_mem
 - rsp-server.c, 557
- rsp_write_mem_bin
 - rsp-server.c, 557
- rsp_write_reg
 - rsp-server.c, 558
- rtl
 - cuc_known_insn, 66
- rtx_sock
 - eth_device, 113
- rtx_type
 - eth_device, 113
- run_sched_out_of_line
 - dyn-rec.c, 349
 - dyn-rec.h, 355

- runtime, 163
 - cpu, 165
 - cuc, 165
 - cycle_duration, 165
 - cycles, 165
 - enabled, 165
 - end_cycles, 165
 - ext_int, 165
 - fexe_log, 165
 - filename, 165
 - fiprof, 165
 - fout, 165
 - fprof, 165
 - hazardwait, 165
 - hush, 165
 - instructions, 165
 - iprompt, 165
 - iprompt_run, 165
 - loadcycles, 165
 - mdelay, 165
 - mem_cycles, 165
 - reset_cycles, 165
 - reset_instructions, 165
 - server_port, 165
 - sim, 165
 - sim-config.c, 808
 - sim-config.h, 814
 - stalled, 165
 - storecycles, 165
 - supercycles, 165
 - time_point, 165
 - vapi, 165
 - vapi_file, 165
- rx
 - eth_device, 113
- rx_buff
 - eth_device, 113
- rx_channel
 - eth_device, 113
- rxb
 - ata_host, 26
- rxbuf
 - dev_16450, 74
- rxbuf_full
 - dev_16450, 74
- rxbuf_head
 - dev_16450, 74
- rxbuf_tail
 - dev_16450, 74
- rxfd
 - eth_device, 113
- rxfile
 - eth_device, 113
 - kbd_state, 144
- rxfs
 - kbd_state, 144
- rxser
 - dev_16450, 74
- s_flags
 - COFF_scnhdr, 48
- S_FUNC
 - op.c, 423
- s_innoptr
 - COFF_scnhdr, 48
- s_name
 - COFF_scnhdr, 48
- s_nlnno
 - COFF_scnhdr, 48
- s_nreloc
 - COFF_scnhdr, 48
- S_OP_NAME
 - op.c, 423
- s_paddr
 - COFF_scnhdr, 48
- s_relptr
 - COFF_scnhdr, 48
- s_scnptr
 - COFF_scnhdr, 48
- s_size
 - COFF_scnhdr, 48
- s_vaddr
 - COFF_scnhdr, 48
- same_transfers
 - cuc/memory.c, 500
- save_t_bound
 - op.c, 429
- saved_regs
 - _cuc_func, 16
- sbp_bf_fwd
 - config, 55
- sbp_bnf_fwd
 - config, 55
- sbuf_buf
 - execute.c, 376
- sbuf_count
 - execute.c, 376
- sbuf_head
 - execute.c, 376
- sbuf_len
 - config, 55
- sbuf_load
 - execute.c, 374
- sbuf_prev_cycles
 - execute.c, 376
- sbuf_store
 - execute.c, 375
- sbuf_tail

- execute.c, 376
- sbuf_total_cyc
 - execute.c, 376
 - execute.h, 267
- sbuf_wait_cyc
 - execute.c, 376
 - execute.h, 267
- scan_decode
 - ps2kbd.c, 756
- scan_table
 - ps2kbd.c, 756
- sched-i386.h
 - set_sched_cycle, 438
- sched.c
 - DECLARE_DEBUG_CHANNEL, 833
 - do_scheduler, 833
 - sched_add, 833
 - sched_find_remove, 833
 - sched_guard, 833
 - SCHED_HEAP_SIZE, 833
 - sched_init, 833
 - sched_next_insn, 833
 - sched_reset, 833
 - SCHED_TIME_MAX, 833
 - scheduler, 833
- sched.h
 - do_scheduler, 835
 - SCHED_ADD, 834
 - sched_add, 835
 - SCHED_FIND_REMOVE, 834
 - sched_find_remove, 835
 - sched_init, 835
 - sched_next_insn, 835
 - sched_reset, 835
 - scheduler, 835
- SCHED_ADD
 - sched.h, 834
- sched_add
 - sched.c, 833
 - sched.h, 835
- sched_entry, 167
 - func, 167
 - next, 167
 - param, 167
 - time, 167
- SCHED_FIND_REMOVE
 - sched.h, 834
- sched_find_remove
 - sched.c, 833
 - sched.h, 835
- sched_guard
 - sched.c, 833
- SCHED_HEAP_SIZE
 - sched.c, 833
- sched_init
 - sched.c, 833
 - sched.h, 835
- sched_next_insn
 - sched.c, 833
 - sched.h, 835
- sched_reset
 - sched.c, 833
 - sched.h, 835
- SCHED_TIME_MAX
 - sched.c, 833
- sched_timer_job
 - tick.c, 840
- schedule_memory
 - cuc.h, 480
 - cuc/memory.c, 500
- scheduler
 - sched.c, 833
 - sched.h, 835
- scheduler_struct, 168
 - free_job_queue, 168
 - job_queue, 168
- scr
 - dev_16450, 74
- SD
 - stats.c, 281
- search_csm
 - insn.c, 487
- sec_end
 - config_section, 59
- sec_start
 - config_section, 59
- sections
 - sim-config.c, 808
- sector_count
 - ata_device, 22
- sector_number
 - ata_device, 22
- sectors
 - ata_device, 22
- sectors_per_track
 - ata_device, 22
- SECURITY_DISABLE_PASSWORD
 - atacmd.h, 624
- SECURITY_ERASE_PREPARE
 - atacmd.h, 624
- SECURITY_ERASE_UNIT
 - atacmd.h, 624
- SECURITY_FREEZE_LOCK
 - atacmd.h, 624
- SECURITY_SET_PASSWORD
 - atacmd.h, 624
- SECURITY_UNLOCK
 - atacmd.h, 624

- SEEK
 - atacmd.h, 624
- selected_tim
 - cuc_bb, 63
- SELMAG
 - elf.h, 260
- send_char
 - 16450.c, 605
- server_fd
 - gdbcomm.c, 534
 - rsp-server.c, 559
 - vapi.c, 870
- server_ip
 - gdbcomm.c, 534
- server_port
 - config, 55
 - gdbcomm.c, 534
 - runtime, 165
- server_request
 - vapi.c, 867
- serverIP
 - vapi.c, 870
- SERVICE
 - atacmd.h, 624
- set_config
 - sim-config.c, 806
- set_config_command
 - sim-config.c, 807
 - sim-config.h, 814
- set_csc_tms
 - mc.c, 745
- set_devint_reg
 - debug-unit.c, 522
- set_direct16
 - abstract.c, 221
 - abstract.h, 238
- set_direct32
 - abstract.c, 222
 - abstract.h, 238
- set_direct8
 - abstract.c, 222
 - abstract.h, 239
- set_dma_nd_i
 - dma.c, 693
 - dma.h, 695
- set_dma_req_i
 - dma.c, 693
 - dma.h, 695
- SET_FEATURES
 - atacmd.h, 624
- SET_FEATURES_REQUIRED_AFTER_
POWER_UP
 - atadevice-cmdi.h, 633
- SET_FIELD
 - fields.h, 721
- SET_FLAG
 - fields.h, 721
- set_func_deps
 - cuc.c, 462
- set_insnbrkpoint
 - trace.c, 286
 - trace.h, 288
- set_io
 - cuc.h, 481
 - insn.c, 487
- set_mask
 - dmmu, 86
 - ic, 127
 - immu, 129
- SET_MAX
 - atacmd.h, 624
- SET_MAX_ADDRESS
 - atacmd.h, 624
- SET_MAX_FREEZE_LOCK
 - atacmd.h, 624
- SET_MAX_LOCK
 - atacmd.h, 624
- SET_MAX_SET_PASSWORD
 - atacmd.h, 624
- SET_MAX_UNLOCK
 - atacmd.h, 624
- set_mem16
 - abstract.c, 223
 - abstract.h, 239
- set_mem32
 - abstract.c, 223
 - abstract.h, 240
- set_mem8
 - abstract.c, 224
 - abstract.h, 240
- set_mem_16_inv
 - abstract.c, 224
- set_mem_16_inv_direct
 - abstract.c, 225
- set_mem_32_inv
 - abstract.c, 225
- set_mem_32_inv_direct
 - abstract.c, 225
- set_mem_8_inv
 - abstract.c, 226
- set_mem_8_inv_direct
 - abstract.c, 226
- set_mem_valid
 - abstract.c, 226
 - abstract.h, 241
- SET_MULTIPLE_MODE
 - atacmd.h, 624
- set_npc

- rsp-server.c, 558
- SET_OV_FLAG_FN
 - execute.h, 262
- set_pc
 - common-i386.h, 325
- set_pc_delay_gpr
 - dyn-rec.c, 350
- set_program32
 - abstract.c, 226
 - abstract.h, 241
- set_program8
 - abstract.c, 226
 - abstract.h, 241
- set_sched_cycle
 - sched-i386.h, 438
- set_stall_state
 - debug-unit.c, 523
 - debug-unit.h, 528
- SET_TRANSFER_MODE_SECTOR_COUNT_-REG
 - atacmd.h, 624
- setsim_mem16
 - abstract.c, 226
 - abstract.h, 241
- setsim_mem32
 - abstract.c, 227
 - abstract.h, 242
- setsim_mem8
 - abstract.c, 227
 - abstract.h, 242
- setsim_reg
 - execute.c, 375
 - execute.h, 267
- sh_addr
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_addralign
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_entsize
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_flags
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_info
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_link
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_name
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_offset
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_size
 - elf32_shdr, 97
 - elf64_shdr, 106
- sh_type
 - elf32_shdr, 97
 - elf64_shdr, 106
- shared
 - cuc_timings, 69
- SHF_ALLOC
 - elf.h, 260
- SHF_EXECINSTR
 - elf.h, 260
- SHF_MASKPROC
 - elf.h, 260
- SHF_WRITE
 - elf.h, 260
- shift
 - ps2kbd.c, 756
- shift_fprintf
 - generate.c, 379
- ship_gprs_out_t
 - dyn-rec.c, 350
- ship_t_out
 - dyn-rec.c, 350
- SHN_ABS
 - elf.h, 260
- SHN_COMMON
 - elf.h, 260
- SHN_HIPROC
 - elf.h, 260
- SHN_HIRESERVE
 - elf.h, 260
- SHN_LOPROC
 - elf.h, 260
- SHN_LORESERVE
 - elf.h, 260
- SHN_UNDEF
 - elf.h, 260
- SHT_DYNAMIC
 - elf.h, 260
- SHT_DYNSYM
 - elf.h, 260
- SHT_HASH
 - elf.h, 260
- SHT_HIPROC
 - elf.h, 260
- SHT_HIUSER
 - elf.h, 260
- SHT_LOPROC
 - elf.h, 260
- SHT_LOUSER

- elf.h, 260
- SHT_NOBITS
 - elf.h, 260
- SHT_NOTE
 - elf.h, 260
- SHT_NULL
 - elf.h, 260
- SHT_NUM
 - elf.h, 260
- SHT_PROGBITS
 - elf.h, 260
- SHT_REL
 - elf.h, 260
- SHT_RELA
 - elf.h, 260
- SHT_SHLIB
 - elf.h, 260
- SHT_STRTAB
 - elf.h, 260
- SHT_SYMTAB
 - elf.h, 260
- sigs
 - ata_device, 22
- sigsegv_addr
 - dyn-rec.c, 351
- sigsegv_state
 - dyn-rec.c, 351
- sigval
 - rsp-server.c, 559
- sim
 - config, 55
 - runtime, 165
- sim-cmd.c, 783
 - check_insn_exec, 785
 - handle_sim_command, 785
 - print_insn_exec, 786
 - reenter_int, 787
 - reg_sim_stat, 787
 - sim_cmd_break, 788
 - sim_cmd_breaks, 788
 - sim_cmd_cm, 788
 - sim_cmd_cuc, 788
 - sim_cmd_de, 791
 - sim_cmd_debug, 791
 - sim_cmd_dh, 791
 - sim_cmd_dm, 791
 - sim_cmd_dv, 792
 - sim_cmd_help, 792
 - sim_cmd_hist, 792
 - sim_cmd_info, 793
 - sim_cmd_mprofile, 793
 - sim_cmd_pc, 793
 - sim_cmd_pm, 794
 - sim_cmd_pr, 794
 - sim_cmd_profile, 794
 - sim_cmd_quit, 794
 - sim_cmd_r, 795
 - sim_cmd_reset, 795
 - sim_cmd_run, 795
 - sim_cmd_set, 796
 - sim_cmd_setdbch, 796
 - sim_cmd_stall, 796
 - sim_cmd_stats, 796
 - sim_cmd_trace, 797
 - sim_cmd_unstall, 797
 - sim_commands, 798
 - sim_stats, 798
 - strip_space, 797
 - to_insn_num, 798
- sim-cmd.h, 799
 - handle_sim_command, 799
 - reg_sim_stat, 799
- sim-config.c, 800
 - base_include, 802
 - config, 808
 - cur_section, 808
 - init_defconfig, 802
 - parse_args, 802
 - print_config, 802
 - read_script_file, 803
 - reg_config_param, 803
 - reg_config_sec, 805
 - reg_config_secs, 805
 - reg_sim_sec, 806
 - runtime, 808
 - sections, 808
 - set_config, 806
 - set_config_command, 807
 - sim_clkcycle, 807
 - sim_debug, 807
 - sim_exe_log, 807
 - sim_exe_log_end, 808
 - sim_exe_log_fn, 808
 - sim_exe_log_marker, 808
 - sim_exe_log_start, 808
 - sim_exe_log_type, 808
 - sim_history, 808
 - sim_mprof_fn, 808
 - sim_mprofile, 808
 - sim_prof_fn, 808
 - sim_profile, 808
 - sim_verbose, 808
 - switch_param, 808
- sim-config.h, 809
 - CHECK_INT_TIME, 810
 - config, 814
 - cur_section, 814
 - do_stats, 814

- EXE_LOG_HARDWARE, 810
- EXE_LOG_SIMPLE, 810
- EXE_LOG_SOFTWARE, 810
- init_defconfig, 811
- MAX_SBUF_LEN, 810
- param_t, 810
- paramt_addr, 810
- paramt_int, 810
- paramt_longlong, 810
- paramt_none, 810
- paramt_str, 810
- paramt_word, 810
- parse_args, 811
- print_config, 811
- PRINTF, 810
- reg_config_param, 811
- reg_config_sec, 813
- reg_config_secs, 813
- runtime, 814
- set_config_command, 814
- STR_SIZE, 810
- sim_clkcycle
 - sim-config.c, 807
- sim_cmd_break
 - sim-cmd.c, 788
- sim_cmd_breaks
 - sim-cmd.c, 788
- sim_cmd_cm
 - sim-cmd.c, 788
- sim_cmd_cuc
 - sim-cmd.c, 788
- sim_cmd_de
 - sim-cmd.c, 791
- sim_cmd_debug
 - sim-cmd.c, 791
- sim_cmd_dh
 - sim-cmd.c, 791
- sim_cmd_dm
 - sim-cmd.c, 791
- sim_cmd_dv
 - sim-cmd.c, 792
- sim_cmd_help
 - sim-cmd.c, 792
- sim_cmd_hist
 - sim-cmd.c, 792
- sim_cmd_info
 - sim-cmd.c, 793
- sim_cmd_mprofile
 - sim-cmd.c, 793
- sim_cmd_pc
 - sim-cmd.c, 793
- sim_cmd_pm
 - sim-cmd.c, 794
- sim_cmd_pr
 - sim-cmd.c, 794
- sim_cmd_profile
 - sim-cmd.c, 794
- sim_cmd_quit
 - sim-cmd.c, 794
- sim_cmd_r
 - sim-cmd.c, 795
- sim_cmd_reset
 - sim-cmd.c, 795
- sim_cmd_run
 - sim-cmd.c, 795
- sim_cmd_set
 - sim-cmd.c, 796
- sim_cmd_setdbch
 - sim-cmd.c, 796
- sim_cmd_stall
 - sim-cmd.c, 796
- sim_cmd_stats
 - sim-cmd.c, 796
- sim_cmd_trace
 - sim-cmd.c, 797
- sim_cmd_unstall
 - sim-cmd.c, 797
- sim_command, 169
 - cmd_handle, 169
 - name, 169
- sim_commands
 - sim-cmd.c, 798
- sim_debug
 - sim-config.c, 807
- sim_done
 - toplevel-support.c, 854
 - toplevel-support.h, 859
- sim_exe_log
 - sim-config.c, 807
- sim_exe_log_end
 - sim-config.c, 808
- sim_exe_log_fn
 - sim-config.c, 808
- sim_exe_log_marker
 - sim-config.c, 808
- sim_exe_log_start
 - sim-config.c, 808
- sim_exe_log_type
 - sim-config.c, 808
- sim_history
 - sim-config.c, 808
- sim_init
 - toplevel-support.c, 854
 - toplevel-support.h, 859
- sim_mprof_fn
 - sim-config.c, 808
- sim_mprofile
 - sim-config.c, 808

- sim_prof_fn
 - sim-config.c, 808
- sim_profile
 - sim-config.c, 808
- sim_reset
 - toplevel-support.c, 855
 - toplevel-support.h, 860
- sim_reset_hook, 170
 - dat, 170
 - next, 170
 - reset_hook, 170
- sim_reset_hooks
 - toplevel-support.c, 856
- sim_stat, 171
 - dat, 171
 - next, 171
 - stat_func, 171
- sim_stats
 - sim-cmd.c, 798
- sim_verbose
 - sim-config.c, 808
- simgetstr
 - simprintf.c, 837
- simmem_read16
 - peripheral/memory.c, 505
- simmem_read32
 - peripheral/memory.c, 506
- simmem_read8
 - peripheral/memory.c, 506
- simmem_read_zero16
 - peripheral/memory.c, 506
- simmem_read_zero32
 - peripheral/memory.c, 506
- simmem_read_zero8
 - peripheral/memory.c, 506
- simmem_write16
 - peripheral/memory.c, 506
- simmem_write32
 - peripheral/memory.c, 506
- simmem_write8
 - peripheral/memory.c, 506
- simmem_write_null16
 - peripheral/memory.c, 506
- simmem_write_null32
 - peripheral/memory.c, 506
- simmem_write_null8
 - peripheral/memory.c, 506
- simpl32-defs.h
 - PARAMS, 439
- simplify_bb
 - bb.c, 446
- simprintf
 - simprintf.c, 837
 - simprintf.h, 838
- simprintf.c
 - FMTLEN, 837
 - fmtstr, 837
 - simgetstr, 837
 - simprintf, 837
 - STACK_ARGS, 837
- simprintf.h
 - simprintf, 838
- size
 - _csm_list, 14
 - ata_device, 22
 - BMP_HEADER, 29
 - cuc_timing_table, 68
 - cuc_timings, 69
 - dev_generic, 77
 - dev_memarea, 79
 - INFOHEADER, 131
 - mem_config, 150
- size_mask
 - dev_memarea, 79
- size_sect
 - ata_device, 22
- sizei
 - cuc_timing_table, 68
- skew
 - dev_16450, 74
- sl_entosz
 - COFF_slib, 49
- sl_pathndx
 - COFF_slib, 49
- SLEEP
 - atacmd.h, 624
- slowdown
 - kbd_state, 144
- SMART
 - atacmd.h, 624
- SMART_ATTRIBITE_AUTOSAVE
 - atacmd.h, 624
- SMART_DISABLE_OPERATIONS
 - atacmd.h, 624
- SMART_ENABLE_OPERATIONS
 - atacmd.h, 624
- SMART_EXECUTE_OFFLINE_IMMEDIATE
 - atacmd.h, 624
- SMART_READ_DATA
 - atacmd.h, 624
- SMART_READ_LOG
 - atacmd.h, 624
- SMART_RETURN_STATUS
 - atacmd.h, 624
- SMART_SAVE_ATTRIBUTE_VALUES
 - atacmd.h, 624
- SMART_WRITE_LOG
 - atacmd.h, 624

- socket_fd
 - tcp_channel, 176
- sockif
 - eth_device, 113
- source
 - dma_channel, 81
- source_mask
 - dma_channel, 81
- SPEEDY_CALL
 - op-i386.h, 403
- spr-defs.h
 - MATCHPOINTS_TO_NDP, 305
 - MAX_GRP, 305
 - MAX_MATCHPOINTS, 307
 - MAX_SPRS, 307
 - MAX_SPRS_PER_GRP, 307
 - MAX_SPRS_PER_GRP_BITS, 307
 - MAX_WATCHPOINTS, 307
 - NOP_CNT_RESET, 307
 - NOP_EXIT, 307
 - NOP_NOP, 307
 - NOP_PRINTF, 307
 - NOP_PUTC, 307
 - NOP_REPORT, 307
 - NOP_REPORT_FIRST, 307
 - NOP_REPORT_LAST, 307
 - SPR_CPUCFGR, 307
 - SPR_CPUCFGR_CGF, 307
 - SPR_CPUCFGR_NSGF, 307
 - SPR_CPUCFGR_OB32S, 307
 - SPR_CPUCFGR_OB64S, 307
 - SPR_CPUCFGR_OF32S, 307
 - SPR_CPUCFGR_OF64S, 307
 - SPR_CPUCFGR_OV64S, 307
 - SPR_CPUCFGR_RES, 307
 - SPR_DCBFR, 307
 - SPR_DCBIR, 307
 - SPR_DCBLR, 307
 - SPR_DCBPR, 307
 - SPR_DCBWR, 307
 - SPR_DCCFGR, 307
 - SPR_DCCFGR_CBFRI, 307
 - SPR_DCCFGR_CBIRI, 307
 - SPR_DCCFGR_CBLRI, 307
 - SPR_DCCFGR_CBPRI, 307
 - SPR_DCCFGR_CBS, 307
 - SPR_DCCFGR_CBS_OFF, 307
 - SPR_DCCFGR_CBWBRI, 307
 - SPR_DCCFGR_CCRI, 307
 - SPR_DCCFGR_CWS, 307
 - SPR_DCCFGR_NCS, 307
 - SPR_DCCFGR_NCS_OFF, 307
 - SPR_DCCFGR_NCW, 307
 - SPR_DCCFGR_NCW_OFF, 307
 - SPR_DCCR, 307
 - SPR_DCCR_EW, 307
 - SPR_DCFGR, 307
 - SPR_DCFGR_NDP, 307
 - SPR_DCFGR_NDP1, 307
 - SPR_DCFGR_NDP2, 307
 - SPR_DCFGR_NDP3, 307
 - SPR_DCFGR_NDP4, 307
 - SPR_DCFGR_NDP5, 307
 - SPR_DCFGR_NDP6, 307
 - SPR_DCFGR_NDP7, 307
 - SPR_DCFGR_NDP8, 307
 - SPR_DCFGR_WPCI, 307
 - SPR_DCR, 307
 - SPR_DCR_BASE, 307
 - SPR_DCR_CC, 307
 - SPR_DCR_CC_EQUAL, 307
 - SPR_DCR_CC_GREAT, 307
 - SPR_DCR_CC_GREATE, 307
 - SPR_DCR_CC_LESS, 307
 - SPR_DCR_CC_LESSE, 307
 - SPR_DCR_CC_MASKED, 307
 - SPR_DCR_CC_NEQUAL, 307
 - SPR_DCR_CT, 307
 - SPR_DCR_CT_DISABLED, 307
 - SPR_DCR_CT_IFEA, 307
 - SPR_DCR_CT_LD, 307
 - SPR_DCR_CT_LEA, 307
 - SPR_DCR_CT_LSD, 307
 - SPR_DCR_CT_LSEA, 307
 - SPR_DCR_CT_SD, 307
 - SPR_DCR_CT_SEA, 307
 - SPR_DCR_DP, 307
 - SPR_DCR_LAST, 307
 - SPR_DCR_SC, 307
 - SPR_DMMUCFGR, 307
 - SPR_DMMUCFGR_CRI, 307
 - SPR_DMMUCFGR_HTR, 307
 - SPR_DMMUCFGR_NAE, 307
 - SPR_DMMUCFGR_NTS, 307
 - SPR_DMMUCFGR_NTS_OFF, 307
 - SPR_DMMUCFGR_NTW, 307
 - SPR_DMMUCFGR_NTW_OFF, 307
 - SPR_DMMUCFGR_PRI, 307
 - SPR_DMMUCFGR_TEIRI, 307
 - SPR_DMMUCR, 307
 - SPR_DMMUCR_P1S, 307
 - SPR_DMMUCR_P2S, 307
 - SPR_DMMUCR_PADDR_WIDTH, 307
 - SPR_DMMUCR_VADDR_WIDTH, 307
 - SPR_DMR1, 307
 - SPR_DMR1_BT, 307
 - SPR_DMR1_CW, 307
 - SPR_DMR1_CW0, 307

SPR_DMR1_CW0_AND, 307
SPR_DMR1_CW0_OR, 307
SPR_DMR1_CW1, 307
SPR_DMR1_CW1_AND, 307
SPR_DMR1_CW1_OR, 307
SPR_DMR1_CW2, 307
SPR_DMR1_CW2_AND, 307
SPR_DMR1_CW2_OR, 307
SPR_DMR1_CW3, 307
SPR_DMR1_CW3_AND, 307
SPR_DMR1_CW3_OR, 307
SPR_DMR1_CW4, 307
SPR_DMR1_CW4_AND, 307
SPR_DMR1_CW4_OR, 307
SPR_DMR1_CW5, 307
SPR_DMR1_CW5_AND, 307
SPR_DMR1_CW5_OR, 307
SPR_DMR1_CW6, 307
SPR_DMR1_CW6_AND, 307
SPR_DMR1_CW6_OR, 307
SPR_DMR1_CW7, 307
SPR_DMR1_CW7_AND, 307
SPR_DMR1_CW7_OR, 307
SPR_DMR1_CW8, 307
SPR_DMR1_CW8_AND, 307
SPR_DMR1_CW8_OR, 307
SPR_DMR1_CW9, 307
SPR_DMR1_CW9_AND, 307
SPR_DMR1_CW9_OR, 307
SPR_DMR1_RES1, 307
SPR_DMR1_RES2, 307
SPR_DMR1_ST, 307
SPR_DMR2, 307
SPR_DMR2_AWTC, 307
SPR_DMR2_AWTC_OFF, 307
SPR_DMR2_WBS, 307
SPR_DMR2_WBS_OFF, 307
SPR_DMR2_WCE0, 307
SPR_DMR2_WCE1, 307
SPR_DMR2_WGB, 307
SPR_DMR2_WGB_OFF, 307
SPR_DRR, 307
SPR_DRR_AE, 307
SPR_DRR_BUSEE, 307
SPR_DRR_DME, 307
SPR_DRR_DPFE, 307
SPR_DRR_IE, 307
SPR_DRR_IIE, 307
SPR_DRR_IME, 307
SPR_DRR_IPFE, 307
SPR_DRR_RE, 307
SPR_DRR_RSTE, 307
SPR_DRR_SCE, 307
SPR_DRR_TE, 307
SPR_DRR_TTE, 307
SPR_DSR, 307
SPR_DSR_AE, 307
SPR_DSR_BUSEE, 307
SPR_DSR_DME, 307
SPR_DSR_DPFE, 307
SPR_DSR_IE, 307
SPR_DSR_IIE, 307
SPR_DSR_IME, 307
SPR_DSR_IPFE, 307
SPR_DSR_RE, 307
SPR_DSR_RSTE, 307
SPR_DSR_SCE, 307
SPR_DSR_SSE, 307
SPR_DSR_TE, 307
SPR_DSR_TTE, 307
SPR_DTLBMR_BASE, 307
SPR_DTLBMR_CID, 307
SPR_DTLBMR_LAST, 307
SPR_DTLBMR_LRU, 307
SPR_DTLBMR_PL1, 307
SPR_DTLBMR_V, 307
SPR_DTLBMR_VPN, 307
SPR_DTLBTR_A, 307
SPR_DTLBTR_BASE, 307
SPR_DTLBTR_CC, 307
SPR_DTLBTR_CI, 307
SPR_DTLBTR_D, 307
SPR_DTLBTR_LAST, 307
SPR_DTLBTR_PPN, 307
SPR_DTLBTR_SRE, 307
SPR_DTLBTR_SWE, 307
SPR_DTLBTR_URE, 307
SPR_DTLBTR_UWE, 307
SPR_DTLBTR_WBC, 307
SPR_DTLBTR_WOM, 307
SPR_DVR, 307
SPR_DWCRO, 307
SPR_DWCR1, 307
SPR_DWCR_COUNT, 307
SPR_DWCR_MATCH, 307
SPR_DWCR_MATCH_OFF, 307
SPR_EEAR_BASE, 307
SPR_EEAR_LAST, 307
SPR_EPCR_BASE, 307
SPR_EPCR_LAST, 307
SPR_ESR_BASE, 307
SPR_ESR_LAST, 307
SPR_ICBIR, 307
SPR_ICBLR, 307
SPR_ICBPR, 307
SPR_ICCFGR, 307
SPR_ICCFGR_CBIRI, 307
SPR_ICCFGR_CBLRI, 307

SPR_ICCFGR_CBPRI, 307
SPR_ICCFGR_CBS, 307
SPR_ICCFGR_CBS_OFF, 307
SPR_ICCFGR_CCRI, 307
SPR_ICCFGR_NCS, 307
SPR_ICCFGR_NCS_OFF, 307
SPR_ICCFGR_NCW, 307
SPR_ICCFGR_NCW_OFF, 307
SPR_ICCR, 307
SPR_ICCR_EW, 307
SPR_ICR_BASE, 307
SPR_ICR_LAST, 307
SPR_IMMUCFGR, 307
SPR_IMMUCFGR_CRI, 307
SPR_IMMUCFGR_HTR, 307
SPR_IMMUCFGR_NAE, 307
SPR_IMMUCFGR_NTS, 307
SPR_IMMUCFGR_NTS_OFF, 307
SPR_IMMUCFGR_NTW, 307
SPR_IMMUCFGR_NTW_OFF, 307
SPR_IMMUCFGR_PRI, 307
SPR_IMMUCFGR_TEIRI, 307
SPR_IMMUCR, 307
SPR_IMMUCR_P1S, 307
SPR_IMMUCR_P2S, 307
SPR_IMMUCR_PADDR_WIDTH, 307
SPR_IMMUCR_VADDR_WIDTH, 307
SPR_ITLBMR_BASE, 307
SPR_ITLBMR_CID, 307
SPR_ITLBMR_LAST, 307
SPR_ITLBMR_LRU, 307
SPR_ITLBMR_PL1, 307
SPR_ITLBMR_V, 307
SPR_ITLBMR_VPN, 307
SPR_ITLBTR_A, 307
SPR_ITLBTR_BASE, 307
SPR_ITLBTR_CC, 307
SPR_ITLBTR_CI, 307
SPR_ITLBTR_D, 307
SPR_ITLBTR_LAST, 307
SPR_ITLBTR_PPN, 307
SPR_ITLBTR_SXE, 307
SPR_ITLBTR_UXE, 307
SPR_ITLBTR_WBC, 307
SPR_ITLBTR_WOM, 307
SPR_MACHI, 307
SPR_MACLO, 307
SPR_NPC, 307
SPR_PCCFGR, 307
SPR_PCCR, 307
SPR_PCMR, 307
SPR_PCMR_BS, 307
SPR_PCMR_CISM, 307
SPR_PCMR_CIU, 307
SPR_PCMR_CP, 307
SPR_PCMR_DCM, 307
SPR_PCMR_DDS, 307
SPR_PCMR_DTLBM, 307
SPR_PCMR_ICM, 307
SPR_PCMR_IF, 307
SPR_PCMR_IFS, 307
SPR_PCMR_ITLBM, 307
SPR_PCMR_LA, 307
SPR_PCMR_LSUS, 307
SPR_PCMR_SA, 307
SPR_PCMR_UMRA, 307
SPR_PCMR_WPE, 307
SPR_PICMR, 307
SPR_PICMR_IUM, 307
SPR_PICPR, 307
SPR_PICPR_IPRIO, 307
SPR_PICSR, 307
SPR_PICSR_IS, 307
SPR_PMR, 307
SPR_PMR_DCGE, 307
SPR_PMR_DME, 307
SPR_PMR_SDF, 307
SPR_PMR_SME, 307
SPR_PMR_SUME, 307
SPR_PPC, 307
SPR_SR, 307
SPR_SR_CE, 307
SPR_SR_CID, 307
SPR_SR_CY, 307
SPR_SR_DCE, 307
SPR_SR_DME, 307
SPR_SR_DSX, 307
SPR_SR_EPH, 307
SPR_SR_F, 307
SPR_SR_FO, 307
SPR_SR_ICE, 307
SPR_SR_IEE, 307
SPR_SR_IME, 307
SPR_SR_LEE, 307
SPR_SR_OV, 307
SPR_SR_OVE, 307
SPR_SR_RES, 307
SPR_SR_SM, 307
SPR_SR_SUMRA, 307
SPR_SR_TEE, 307
SPR_TTCR, 307
SPR_TTCR_PERIOD, 307
SPR_TTMR, 307
SPR_TTMR_CR, 307
SPR_TTMR_IE, 307
SPR_TTMR_IP, 307
SPR_TTMR_M, 307
SPR_TTMR_PERIOD, 307

- SPR_TTMR_RT, 307
- SPR_TTMR_SR, 307
- SPR_UPR, 307
- SPR_UPR_CUP, 307
- SPR_UPR_DCP, 307
- SPR_UPR_DMP, 307
- SPR_UPR_DUP, 307
- SPR_UPR_ICP, 307
- SPR_UPR_IMP, 307
- SPR_UPR_MP, 307
- SPR_UPR_PCUP, 307
- SPR_UPR_PICP, 307
- SPR_UPR_PMP, 307
- SPR_UPR_RES, 307
- SPR_UPR_TTP, 307
- SPR_UPR_UP, 307
- SPR_VR, 307
- SPR_VR_CFG, 307
- SPR_VR_CFG_OFF, 307
- SPR_VR_RES, 307
- SPR_VR_REV, 307
- SPR_VR_REV_OFF, 307
- SPR_VR_VER, 307
- SPR_VR_VER_OFF, 307
- SPRGROUP_D, 307
- SPRGROUP_DC, 307
- SPRGROUP_DMMU, 307
- SPRGROUP_IC, 307
- SPRGROUP_IMMU, 307
- SPRGROUP_MAC, 307
- SPRGROUP_PC, 307
- SPRGROUP_PIC, 307
- SPRGROUP_PM, 307
- SPRGROUP_SYS, 307
- SPRGROUP_TT, 307
- spr-dump.c
 - dump_spr, 309
 - ret_spr, 309
 - spr_cpucfgr, 309
 - spr_d_group, 309
 - spr_dc_group, 310
 - spr_dccfgr, 310
 - spr_dccr, 310
 - spr_dcfgr, 310
 - spr_dmmu_group, 311
 - spr_dmmucfgr, 311
 - spr_dmmucr, 311
 - spr_dmr1, 311
 - spr_dmr2, 312
 - spr_drr, 312
 - spr_dsr, 312
 - spr_dtlbmr, 313
 - spr_dtlbtr, 313
 - spr_dwcr, 313
 - spr_groups, 313
 - spr_ic_group, 314
 - spr_icefgr, 314
 - spr_iccr, 314
 - spr_immu_group, 314
 - spr_immucfgr, 315
 - spr_immucr, 315
 - spr_itlbmr, 315
 - spr_itlbtr, 315
 - spr_mac_group, 316
 - spr_one_val, 316
 - spr_pc_group, 316
 - spr_pcmr, 316
 - spr_pic_group, 317
 - spr_picmr, 317
 - spr_pm_group, 317
 - spr_pmr, 317
 - spr_sr, 317
 - spr_sys_group, 318
 - spr_tt_group, 318
 - spr_ttmr, 318
 - spr_upr, 319
 - spr_vr, 319
- spr-dump.h
 - dump_spr, 320
- spr_bit_def, 172
 - mask, 172
 - name, 172
- SPR_CPUCFGR
 - spr-defs.h, 307
- spr_cpucfgr
 - spr-dump.c, 309
- SPR_CPUCFGR_CGF
 - spr-defs.h, 307
- SPR_CPUCFGR_NSGF
 - spr-defs.h, 307
- SPR_CPUCFGR_OB32S
 - spr-defs.h, 307
- SPR_CPUCFGR_OB64S
 - spr-defs.h, 307
- SPR_CPUCFGR_OF32S
 - spr-defs.h, 307
- SPR_CPUCFGR_OF64S
 - spr-defs.h, 307
- SPR_CPUCFGR_OV64S
 - spr-defs.h, 307
- SPR_CPUCFGR_RES
 - spr-defs.h, 307
- spr_d_group
 - spr-dump.c, 309
- spr_dc_group
 - spr-dump.c, 310
- SPR_DCBFR
 - spr-defs.h, 307

- SPR_DCBIR
 - spr-defs.h, 307
- SPR_DCBLR
 - spr-defs.h, 307
- SPR_DCBPR
 - spr-defs.h, 307
- SPR_DCBWR
 - spr-defs.h, 307
- SPR_DCCFGR
 - spr-defs.h, 307
- spr_dccfgr
 - spr-dump.c, 310
- SPR_DCCFGR_CBFRI
 - spr-defs.h, 307
- SPR_DCCFGR_CBIRI
 - spr-defs.h, 307
- SPR_DCCFGR_CBLRI
 - spr-defs.h, 307
- SPR_DCCFGR_CBPRI
 - spr-defs.h, 307
- SPR_DCCFGR_CBS
 - spr-defs.h, 307
- SPR_DCCFGR_CBS_OFF
 - spr-defs.h, 307
- SPR_DCCFGR_CBWBRI
 - spr-defs.h, 307
- SPR_DCCFGR_CCRI
 - spr-defs.h, 307
- SPR_DCCFGR_CWS
 - spr-defs.h, 307
- SPR_DCCFGR_NCS
 - spr-defs.h, 307
- SPR_DCCFGR_NCS_OFF
 - spr-defs.h, 307
- SPR_DCCFGR_NCW
 - spr-defs.h, 307
- SPR_DCCFGR_NCW_OFF
 - spr-defs.h, 307
- SPR_DCCR
 - spr-defs.h, 307
- spr_dccr
 - spr-dump.c, 310
- SPR_DCCR_EW
 - spr-defs.h, 307
- SPR_DCFGR
 - spr-defs.h, 307
- spr_dcfgr
 - spr-dump.c, 310
- SPR_DCFGR_NDP
 - spr-defs.h, 307
- SPR_DCFGR_NDP1
 - spr-defs.h, 307
- SPR_DCFGR_NDP2
 - spr-defs.h, 307
- SPR_DCFGR_NDP3
 - spr-defs.h, 307
- SPR_DCFGR_NDP4
 - spr-defs.h, 307
- SPR_DCFGR_NDP5
 - spr-defs.h, 307
- SPR_DCFGR_NDP6
 - spr-defs.h, 307
- SPR_DCFGR_NDP7
 - spr-defs.h, 307
- SPR_DCFGR_NDP8
 - spr-defs.h, 307
- SPR_DCFGR_WPCI
 - spr-defs.h, 307
- SPR_DCR
 - spr-defs.h, 307
- SPR_DCR_BASE
 - spr-defs.h, 307
- SPR_DCR_CC
 - spr-defs.h, 307
- SPR_DCR_CC_EQUAL
 - spr-defs.h, 307
- SPR_DCR_CC_GREAT
 - spr-defs.h, 307
- SPR_DCR_CC_GREATER
 - spr-defs.h, 307
- SPR_DCR_CC_LESS
 - spr-defs.h, 307
- SPR_DCR_CC_LESSE
 - spr-defs.h, 307
- SPR_DCR_CC_MASKED
 - spr-defs.h, 307
- SPR_DCR_CC_NEQUAL
 - spr-defs.h, 307
- SPR_DCR_CT
 - spr-defs.h, 307
- SPR_DCR_CT_DISABLED
 - spr-defs.h, 307
- SPR_DCR_CT_IFEA
 - spr-defs.h, 307
- SPR_DCR_CT_LD
 - spr-defs.h, 307
- SPR_DCR_CT_LEA
 - spr-defs.h, 307
- SPR_DCR_CT_LSD
 - spr-defs.h, 307
- SPR_DCR_CT_LSEA
 - spr-defs.h, 307
- SPR_DCR_CT_SD
 - spr-defs.h, 307
- SPR_DCR_CT_SEA
 - spr-defs.h, 307
- SPR_DCR_DP
 - spr-defs.h, 307

- SPR_DCR_LAST
 - spr-defs.h, 307
- SPR_DCR_SC
 - spr-defs.h, 307
- spr_def, 173
 - bits, 173
 - from_spr, 173
 - name, 173
 - to_spr, 173
- spr_dmmu_group
 - spr-dump.c, 311
- SPR_DMMUCFGR
 - spr-defs.h, 307
- spr_dmmucfgr
 - spr-dump.c, 311
- SPR_DMMUCFGR_CRI
 - spr-defs.h, 307
- SPR_DMMUCFGR_HTR
 - spr-defs.h, 307
- SPR_DMMUCFGR_NAE
 - spr-defs.h, 307
- SPR_DMMUCFGR_NTS
 - spr-defs.h, 307
- SPR_DMMUCFGR_NTS_OFF
 - spr-defs.h, 307
- SPR_DMMUCFGR_NTW
 - spr-defs.h, 307
- SPR_DMMUCFGR_NTW_OFF
 - spr-defs.h, 307
- SPR_DMMUCFGR_PRI
 - spr-defs.h, 307
- SPR_DMMUCFGR_TEIRI
 - spr-defs.h, 307
- SPR_DMMUCR
 - spr-defs.h, 307
- spr_dmmucr
 - spr-dump.c, 311
- SPR_DMMUCR_P1S
 - spr-defs.h, 307
- SPR_DMMUCR_P2S
 - spr-defs.h, 307
- SPR_DMMUCR_PADDR_WIDTH
 - spr-defs.h, 307
- SPR_DMMUCR_VADDR_WIDTH
 - spr-defs.h, 307
- SPR_DMR1
 - spr-defs.h, 307
- spr_dmr1
 - spr-dump.c, 311
- SPR_DMR1_BT
 - spr-defs.h, 307
- SPR_DMR1_CW
 - spr-defs.h, 307
- SPR_DMR1_CW0
 - spr-defs.h, 307
- SPR_DMR1_CW0_AND
 - spr-defs.h, 307
- SPR_DMR1_CW0_OR
 - spr-defs.h, 307
- SPR_DMR1_CW1
 - spr-defs.h, 307
- SPR_DMR1_CW1_AND
 - spr-defs.h, 307
- SPR_DMR1_CW1_OR
 - spr-defs.h, 307
- SPR_DMR1_CW2
 - spr-defs.h, 307
- SPR_DMR1_CW2_AND
 - spr-defs.h, 307
- SPR_DMR1_CW2_OR
 - spr-defs.h, 307
- SPR_DMR1_CW3
 - spr-defs.h, 307
- SPR_DMR1_CW3_AND
 - spr-defs.h, 307
- SPR_DMR1_CW3_OR
 - spr-defs.h, 307
- SPR_DMR1_CW4
 - spr-defs.h, 307
- SPR_DMR1_CW4_AND
 - spr-defs.h, 307
- SPR_DMR1_CW4_OR
 - spr-defs.h, 307
- SPR_DMR1_CW5
 - spr-defs.h, 307
- SPR_DMR1_CW5_AND
 - spr-defs.h, 307
- SPR_DMR1_CW5_OR
 - spr-defs.h, 307
- SPR_DMR1_CW6
 - spr-defs.h, 307
- SPR_DMR1_CW6_AND
 - spr-defs.h, 307
- SPR_DMR1_CW6_OR
 - spr-defs.h, 307
- SPR_DMR1_CW7
 - spr-defs.h, 307
- SPR_DMR1_CW7_AND
 - spr-defs.h, 307
- SPR_DMR1_CW7_OR
 - spr-defs.h, 307
- SPR_DMR1_CW8
 - spr-defs.h, 307
- SPR_DMR1_CW8_AND
 - spr-defs.h, 307
- SPR_DMR1_CW8_OR
 - spr-defs.h, 307
- SPR_DMR1_CW9
 - spr-defs.h, 307

- spr-defs.h, 307
- SPR_DMR1_CW9_AND
 - spr-defs.h, 307
- SPR_DMR1_CW9_OR
 - spr-defs.h, 307
- SPR_DMR1_RES1
 - spr-defs.h, 307
- SPR_DMR1_RES2
 - spr-defs.h, 307
- SPR_DMR1_ST
 - spr-defs.h, 307
- SPR_DMR2
 - spr-defs.h, 307
- spr_dmr2
 - spr-dump.c, 312
- SPR_DMR2_AWTC
 - spr-defs.h, 307
- SPR_DMR2_AWTC_OFF
 - spr-defs.h, 307
- SPR_DMR2_WBS
 - spr-defs.h, 307
- SPR_DMR2_WBS_OFF
 - spr-defs.h, 307
- SPR_DMR2_WCE0
 - spr-defs.h, 307
- SPR_DMR2_WCE1
 - spr-defs.h, 307
- SPR_DMR2_WGB
 - spr-defs.h, 307
- SPR_DMR2_WGB_OFF
 - spr-defs.h, 307
- SPR_DRR
 - spr-defs.h, 307
- spr_drr
 - spr-dump.c, 312
- SPR_DRR_AE
 - spr-defs.h, 307
- SPR_DRR_BUSEE
 - spr-defs.h, 307
- SPR_DRR_DME
 - spr-defs.h, 307
- SPR_DRR_DPFE
 - spr-defs.h, 307
- SPR_DRR_IE
 - spr-defs.h, 307
- SPR_DRR_IIE
 - spr-defs.h, 307
- SPR_DRR_IME
 - spr-defs.h, 307
- SPR_DRR_IPFE
 - spr-defs.h, 307
- SPR_DRR_RE
 - spr-defs.h, 307
- SPR_DRR_RSTE
 - spr-defs.h, 307
- spr-defs.h, 307
- SPR_DRR_SCE
 - spr-defs.h, 307
- SPR_DRR_TE
 - spr-defs.h, 307
- SPR_DRR_TTE
 - spr-defs.h, 307
- SPR_DSR
 - spr-defs.h, 307
- spr_dsr
 - spr-dump.c, 312
- SPR_DSR_AE
 - spr-defs.h, 307
- SPR_DSR_BUSEE
 - spr-defs.h, 307
- SPR_DSR_DME
 - spr-defs.h, 307
- SPR_DSR_DPFE
 - spr-defs.h, 307
- SPR_DSR_IE
 - spr-defs.h, 307
- SPR_DSR_IIE
 - spr-defs.h, 307
- SPR_DSR_IME
 - spr-defs.h, 307
- SPR_DSR_IPFE
 - spr-defs.h, 307
- SPR_DSR_RE
 - spr-defs.h, 307
- SPR_DSR_RSTE
 - spr-defs.h, 307
- SPR_DSR_SCE
 - spr-defs.h, 307
- SPR_DSR_SSE
 - spr-defs.h, 307
- SPR_DSR_TE
 - spr-defs.h, 307
- SPR_DSR_TTE
 - spr-defs.h, 307
- spr_dtlbmr
 - spr-dump.c, 313
- SPR_DTLBMR_BASE
 - spr-defs.h, 307
- SPR_DTLBMR_CID
 - spr-defs.h, 307
- SPR_DTLBMR_LAST
 - spr-defs.h, 307
- SPR_DTLBMR_LRU
 - spr-defs.h, 307
- SPR_DTLBMR_PL1
 - spr-defs.h, 307
- SPR_DTLBMR_V
 - spr-defs.h, 307
- SPR_DTLBMR_VPN

- spr-defs.h, 307
- spr_dtlbtr
 - spr-dump.c, 313
- SPR_DTLBTR_A
 - spr-defs.h, 307
- SPR_DTLBTR_BASE
 - spr-defs.h, 307
- SPR_DTLBTR_CC
 - spr-defs.h, 307
- SPR_DTLBTR_CI
 - spr-defs.h, 307
- SPR_DTLBTR_D
 - spr-defs.h, 307
- SPR_DTLBTR_LAST
 - spr-defs.h, 307
- SPR_DTLBTR_PPN
 - spr-defs.h, 307
- SPR_DTLBTR_SRE
 - spr-defs.h, 307
- SPR_DTLBTR_SWE
 - spr-defs.h, 307
- SPR_DTLBTR_URE
 - spr-defs.h, 307
- SPR_DTLBTR_UWE
 - spr-defs.h, 307
- SPR_DTLBTR_WBC
 - spr-defs.h, 307
- SPR_DTLBTR_WOM
 - spr-defs.h, 307
- SPR_DVR
 - spr-defs.h, 307
- spr_dwcr
 - spr-dump.c, 313
- SPR_DWCR0
 - spr-defs.h, 307
- SPR_DWCR1
 - spr-defs.h, 307
- SPR_DWCR_COUNT
 - spr-defs.h, 307
- SPR_DWCR_MATCH
 - spr-defs.h, 307
- SPR_DWCR_MATCH_OFF
 - spr-defs.h, 307
- SPR_EEAR_BASE
 - spr-defs.h, 307
- SPR_EEAR_LAST
 - spr-defs.h, 307
- SPR_EPCR_BASE
 - spr-defs.h, 307
- SPR_EPCR_LAST
 - spr-defs.h, 307
- SPR_ESR_BASE
 - spr-defs.h, 307
- SPR_ESR_LAST
 - spr-defs.h, 307
- spr_defs.h, 307
- spr_groups
 - spr-dump.c, 313
- spr_ic_group
 - spr-dump.c, 314
- SPR_ICBIR
 - spr-defs.h, 307
- SPR_ICBLR
 - spr-defs.h, 307
- SPR_ICBPR
 - spr-defs.h, 307
- SPR_ICCFGR
 - spr-defs.h, 307
- spr_iccfgr
 - spr-dump.c, 314
- SPR_ICCFGR_CBIRI
 - spr-defs.h, 307
- SPR_ICCFGR_CBLRI
 - spr-defs.h, 307
- SPR_ICCFGR_CBPRI
 - spr-defs.h, 307
- SPR_ICCFGR_CBS
 - spr-defs.h, 307
- SPR_ICCFGR_CBS_OFF
 - spr-defs.h, 307
- SPR_ICCFGR_CCRI
 - spr-defs.h, 307
- SPR_ICCFGR_NCS
 - spr-defs.h, 307
- SPR_ICCFGR_NCS_OFF
 - spr-defs.h, 307
- SPR_ICCFGR_NCW
 - spr-defs.h, 307
- SPR_ICCFGR_NCW_OFF
 - spr-defs.h, 307
- SPR_ICCR
 - spr-defs.h, 307
- spr_iccr
 - spr-dump.c, 314
- SPR_ICCR_EW
 - spr-defs.h, 307
- SPR_ICR_BASE
 - spr-defs.h, 307
- SPR_ICR_LAST
 - spr-defs.h, 307
- spr_immu_group
 - spr-dump.c, 314
- SPR_IMMUCFGR
 - spr-defs.h, 307
- spr_immucfgr
 - spr-dump.c, 315
- SPR_IMMUCFGR_CRI
 - spr-defs.h, 307
- SPR_IMMUCFGR_HTR
 - spr-defs.h, 307

- spr-defs.h, 307
- SPR_IMMUCFGR_NAE
 - spr-defs.h, 307
- SPR_IMMUCFGR_NTS
 - spr-defs.h, 307
- SPR_IMMUCFGR_NTS_OFF
 - spr-defs.h, 307
- SPR_IMMUCFGR_NTW
 - spr-defs.h, 307
- SPR_IMMUCFGR_NTW_OFF
 - spr-defs.h, 307
- SPR_IMMUCFGR_PRI
 - spr-defs.h, 307
- SPR_IMMUCFGR_TEIRI
 - spr-defs.h, 307
- SPR_IMMUCR
 - spr-defs.h, 307
- spr_immucr
 - spr-dump.c, 315
- SPR_IMMUCR_P1S
 - spr-defs.h, 307
- SPR_IMMUCR_P2S
 - spr-defs.h, 307
- SPR_IMMUCR_PADDR_WIDTH
 - spr-defs.h, 307
- SPR_IMMUCR_VADDR_WIDTH
 - spr-defs.h, 307
- spr_itlblr
 - spr-dump.c, 315
- SPR_ITLBMR_BASE
 - spr-defs.h, 307
- SPR_ITLBMR_CID
 - spr-defs.h, 307
- SPR_ITLBMR_LAST
 - spr-defs.h, 307
- SPR_ITLBMR_LRU
 - spr-defs.h, 307
- SPR_ITLBMR_PL1
 - spr-defs.h, 307
- SPR_ITLBMR_V
 - spr-defs.h, 307
- SPR_ITLBMR_VPN
 - spr-defs.h, 307
- spr_itlbtr
 - spr-dump.c, 315
- SPR_ITLBTR_A
 - spr-defs.h, 307
- SPR_ITLBTR_BASE
 - spr-defs.h, 307
- SPR_ITLBTR_CC
 - spr-defs.h, 307
- SPR_ITLBTR_CI
 - spr-defs.h, 307
- SPR_ITLBTR_D
 - spr-defs.h, 307
- SPR_ITLBTR_LAST
 - spr-defs.h, 307
- SPR_ITLBTR_PPN
 - spr-defs.h, 307
- SPR_ITLBTR_SXE
 - spr-defs.h, 307
- SPR_ITLBTR_UXE
 - spr-defs.h, 307
- SPR_ITLBTR_WBC
 - spr-defs.h, 307
- SPR_ITLBTR_WOM
 - spr-defs.h, 307
- spr_mac_group
 - spr-dump.c, 316
- SPR_MACHI
 - spr-defs.h, 307
- SPR_MACLO
 - spr-defs.h, 307
- SPR_NPC
 - spr-defs.h, 307
- spr_one_val
 - spr-dump.c, 316
- spr_pc_group
 - spr-dump.c, 316
- SPR_PCCFGR
 - spr-defs.h, 307
- SPR_PCCR
 - spr-defs.h, 307
- SPR_PCMR
 - spr-defs.h, 307
- spr_pcmr
 - spr-dump.c, 316
- SPR_PCMR_BS
 - spr-defs.h, 307
- SPR_PCMR_CISM
 - spr-defs.h, 307
- SPR_PCMR_CIU
 - spr-defs.h, 307
- SPR_PCMR_CP
 - spr-defs.h, 307
- SPR_PCMR_DCM
 - spr-defs.h, 307
- SPR_PCMR_DDS
 - spr-defs.h, 307
- SPR_PCMR_DTLBM
 - spr-defs.h, 307
- SPR_PCMR_ICM
 - spr-defs.h, 307
- SPR_PCMR_IF
 - spr-defs.h, 307
- SPR_PCMR_IFS
 - spr-defs.h, 307
- SPR_PCMR_ITLBM

- spr-defs.h, 307
- SPR_PCMR_LA
 - spr-defs.h, 307
- SPR_PCMR_LSUS
 - spr-defs.h, 307
- SPR_PCMR_SA
 - spr-defs.h, 307
- SPR_PCMR_UMRA
 - spr-defs.h, 307
- SPR_PCMR_WPE
 - spr-defs.h, 307
- spr_pic_group
 - spr-dump.c, 317
- SPR_PICMR
 - spr-defs.h, 307
- spr_picmr
 - spr-dump.c, 317
- SPR_PICMR_IUM
 - spr-defs.h, 307
- SPR_PICPR
 - spr-defs.h, 307
- SPR_PICPR_IPRIO
 - spr-defs.h, 307
- SPR_PICSR
 - spr-defs.h, 307
- SPR_PICSR_IS
 - spr-defs.h, 307
- spr_pm_group
 - spr-dump.c, 317
- SPR_PMR
 - spr-defs.h, 307
- spr_pmr
 - spr-dump.c, 317
- SPR_PMR_DCGE
 - spr-defs.h, 307
- SPR_PMR_DME
 - spr-defs.h, 307
- SPR_PMR_SDF
 - spr-defs.h, 307
- SPR_PMR_SME
 - spr-defs.h, 307
- SPR_PMR_SUME
 - spr-defs.h, 307
- SPR_PPC
 - spr-defs.h, 307
- spr_read_ttc_r
 - tick.c, 840
 - tick.h, 843
- SPR_SR
 - spr-defs.h, 307
- spr_sr
 - spr-dump.c, 317
- SPR_SR_CE
 - spr-defs.h, 307
- SPR_SR_CID
 - spr-defs.h, 307
- SPR_SR_CY
 - spr-defs.h, 307
- SPR_SR_DCE
 - spr-defs.h, 307
- SPR_SR_DME
 - spr-defs.h, 307
- SPR_SR_DSX
 - spr-defs.h, 307
- SPR_SR_EPH
 - spr-defs.h, 307
- SPR_SR_F
 - spr-defs.h, 307
- SPR_SR_FO
 - spr-defs.h, 307
- SPR_SR_ICE
 - spr-defs.h, 307
- SPR_SR_IEE
 - spr-defs.h, 307
- SPR_SR_IME
 - spr-defs.h, 307
- SPR_SR_LEE
 - spr-defs.h, 307
- SPR_SR_OV
 - spr-defs.h, 307
- SPR_SR_OVE
 - spr-defs.h, 307
- SPR_SR_RES
 - spr-defs.h, 307
- SPR_SR_SM
 - spr-defs.h, 307
- SPR_SR_SUMRA
 - spr-defs.h, 307
- SPR_SR_TEE
 - spr-defs.h, 307
- spr_sys_group
 - spr-dump.c, 318
- spr_tt_group
 - spr-dump.c, 318
- SPR_TTCR
 - spr-defs.h, 307
- SPR_TTCR_PERIOD
 - spr-defs.h, 307
- SPR_TTMR
 - spr-defs.h, 307
- spr_ttmr
 - spr-dump.c, 318
- SPR_TTMR_CR
 - spr-defs.h, 307
- SPR_TTMR_IE
 - spr-defs.h, 307
- SPR_TTMR_IP
 - spr-defs.h, 307

- SPR_TTMR_M
 - spr-defs.h, 307
- SPR_TTMR_PERIOD
 - spr-defs.h, 307
- SPR_TTMR_RT
 - spr-defs.h, 307
- SPR_TTMR_SR
 - spr-defs.h, 307
- SPR_UPR
 - spr-defs.h, 307
- spr_upr
 - spr-dump.c, 319
- SPR_UPR_CUP
 - spr-defs.h, 307
- SPR_UPR_DCP
 - spr-defs.h, 307
- SPR_UPR_DMP
 - spr-defs.h, 307
- SPR_UPR_DUP
 - spr-defs.h, 307
- SPR_UPR_ICP
 - spr-defs.h, 307
- SPR_UPR_IMP
 - spr-defs.h, 307
- SPR_UPR_MP
 - spr-defs.h, 307
- SPR_UPR_PCUP
 - spr-defs.h, 307
- SPR_UPR_PICP
 - spr-defs.h, 307
- SPR_UPR_PMP
 - spr-defs.h, 307
- SPR_UPR_RES
 - spr-defs.h, 307
- SPR_UPR_TTP
 - spr-defs.h, 307
- SPR_UPR_UP
 - spr-defs.h, 307
- SPR_VR
 - spr-defs.h, 307
- spr_vr
 - spr-dump.c, 319
- SPR_VR_CFG
 - spr-defs.h, 307
- SPR_VR_CFG_OFF
 - spr-defs.h, 307
- SPR_VR_RES
 - spr-defs.h, 307
- SPR_VR_REV
 - spr-defs.h, 307
- SPR_VR_REV_OFF
 - spr-defs.h, 307
- SPR_VR_VER
 - spr-defs.h, 307
- SPR_VR_VER_OFF
 - spr-defs.h, 307
- spr_write_tter
 - tick.c, 840
 - tick.h, 843
- spr_write_ttmr
 - tick.c, 840
 - tick.h, 844
- SPRGROUP_D
 - spr-defs.h, 307
- SPRGROUP_DC
 - spr-defs.h, 307
- SPRGROUP_DMMU
 - spr-defs.h, 307
- SPRGROUP_IC
 - spr-defs.h, 307
- SPRGROUP_IMMU
 - spr-defs.h, 307
- SPRGROUP_MAC
 - spr-defs.h, 307
- SPRGROUP_PC
 - spr-defs.h, 307
- SPRGROUP_PIC
 - spr-defs.h, 307
- SPRGROUP_PM
 - spr-defs.h, 307
- SPRGROUP_SYS
 - spr-defs.h, 307
- SPRGROUP_TT
 - spr-defs.h, 307
- sprsprs
 - cpu_state, 60
- sprsprs.c
 - audio_cnt, 322
 - DECLARE_DEBUG_CHANNEL, 322
 - fo, 322
 - mfspr, 322
 - mtspr, 322
 - sprsprs_status, 322
- sprsprs.h
 - dump_spr, 323
 - mfspr, 323
 - mtspr, 324
 - sprsprs_status, 324
- sprsprs_status
 - sprsprs.c, 322
 - sprsprs.h, 324
- SR_REGNUM
 - rsp-server.c, 540
- sstats
 - stats.c, 283
- sstats_entry, 174
 - cnt_dynamic, 174
 - insn, 174

- SSTATS_LEN
 - stats.c, 281
- st_info
 - elf32_sym, 98
 - elf64_sym, 107
- st_name
 - elf32_sym, 98
 - elf64_sym, 107
- st_other
 - elf32_sym, 98
 - elf64_sym, 107
- st_shndx
 - elf32_sym, 98
 - elf64_sym, 107
- st_size
 - elf32_sym, 98
 - elf64_sym, 107
- st_value
 - elf32_sym, 98
 - elf64_sym, 107
- stack
 - profiler.c, 779
- STACK_ARGS
 - simprintf.c, 837
- STACK_SIZE
 - abstract.h, 231
- stack_struct, 175
 - addr, 175
 - cycles, 175
 - name, 175
 - raddr, 175
- stalled
 - runtime, 165
- STANDBY
 - atacmd.h, 624
- STANDBY_IMMEDIATE
 - atacmd.h, 624
- start_addr
 - _cuc_func, 16
 - mprofiler.c, 589
 - rsp-server.c, 559
- stat
 - ata_host, 26
 - vga_state, 180
- stat_func
 - sim_stat, 171
- state
 - ata_device, 22
 - eth_device, 113
- stats.c
 - adddstats, 281
 - addfstats, 281
 - addsstats, 281
 - dc_stats, 282
 - dmmu_stats, 282
 - dstats, 282
 - DSTATS_LEN, 281
 - fstats, 282
 - FSTATS_LEN, 281
 - func_unit_str, 282
 - ic_stats, 282
 - immu_stats, 282
 - initstats, 281
 - or1k_mstats, 282
 - printotherstats, 281
 - printstats, 281
 - raw_stats, 283
 - SD, 281
 - sstats, 283
 - SSTATS_LEN, 281
- stats.h
 - adddstats, 285
 - addfstats, 285
 - addsstats, 285
 - dc_stats, 285
 - dmmu_stats, 285
 - ic_stats, 285
 - immu_stats, 285
 - initstats, 285
 - or1k_mstats, 285
 - printstats, 285
 - RAW_RANGE, 285
 - raw_stats, 285
- status
 - ata_device, 22
 - channel_ops, 39
 - jtr_chain_response, 134
 - jtr_failure_response, 135
 - jtr_read_block_response, 137
 - jtr_read_response, 139
 - jtr_write_block_response, 141
 - jtr_write_response, 143
- STB_GLOBAL
 - elf.h, 260
- STB_LOCAL
 - elf.h, 260
- STB_WEAK
 - elf.h, 260
- store_hitdelay
 - config, 55
- store_missdelay
 - config, 55
- storecycles
 - runtime, 165
- stores_pagefaults
 - dmmustats_entry, 87
- stores_tlbhit
 - dmmustats_entry, 87

- stores_tlbmiss
 - dmmustats_entry, 87
- STR_SIZE
 - sim-config.h, 810
- str_val
 - param_val, 159
- stream
 - ata_device, 22
- strip_space
 - sim-cmd.c, 797
- strndup
 - port.h, 775
 - strndup.c, 776
- strndup.c
 - strndup, 776
- strstrip
 - parse.c, 276
 - parse.h, 279
- STT_FILE
 - elf.h, 260
- STT_FUNC
 - elf.h, 260
- STT_NOTYPE
 - elf.h, 260
- STT_OBJECT
 - elf.h, 260
- STT_SECTION
 - elf.h, 260
- supercycles
 - runtime, 165
- superscalar
 - config, 55
- support/dbchs.h, 815
- support/debug.c, 816
- support/debug.h, 818
- support/dumpverilog.c, 822
- support/dumpverilog.h, 825
- support/misc.c, 827
- support/misc.h, 828
- support/profile.c, 829
- support/profile.h, 830
- support/sched.c, 832
- support/sched.h, 834
- support/simprintf.c, 836
- support/simprintf.h, 838
- SUPPORT_APM
 - atadevice-cmdi.h, 633
- SUPPORT_CFA
 - atadevice-cmdi.h, 633
- SUPPORT_DEVICE_RESET_CMD
 - atadevice-cmdi.h, 633
- SUPPORT_DOWNLOAD_MICROCODE
 - atadevice-cmdi.h, 633
- SUPPORT_HOST_PROTECTED_AREA
 - atadevice-cmdi.h, 633
- SUPPORT_LOOKAHEAD
 - atadevice-cmdi.h, 633
- SUPPORT_NOP_CMD
 - atadevice-cmdi.h, 633
- SUPPORT_POWER_MANAGEMENT
 - atadevice-cmdi.h, 633
- SUPPORT_POWER_UP_IN_STANDBY_MODE
 - atadevice-cmdi.h, 633
- SUPPORT_READ_BUFFER_CMD
 - atadevice-cmdi.h, 633
- SUPPORT_READ_WRITE_DMA_QUEUED
 - atadevice-cmdi.h, 633
- SUPPORT_RELEASE_INTERRUPT
 - atadevice-cmdi.h, 633
- SUPPORT_REMOVABLE_MEDIA
 - atadevice-cmdi.h, 633
- SUPPORT_REMOVABLE_MEDIA_-
 - NOTIFICATION
 - atadevice-cmdi.h, 633
- SUPPORT_SECURITY_MODE
 - atadevice-cmdi.h, 633
- SUPPORT_SERVICE_INTERRUPT
 - atadevice-cmdi.h, 633
- SUPPORT_SET_MAX
 - atadevice-cmdi.h, 633
- SUPPORT_SMART
 - atadevice-cmdi.h, 633
- SUPPORT_WRITE_BUFFER_CMD
 - atadevice-cmdi.h, 633
- SUPPORT_WRITE_CACHE
 - atadevice-cmdi.h, 633
- SWAP_ENDIAN_LONG
 - coff.h, 250
- SWAP_ENDIAN_SHORT
 - coff.h, 250
- switch_param
 - sim-config.c, 808
- swptr
 - dma_channel, 81
- sz
 - dma_channel, 81
- T
 - op-1t.h, 394
 - op-2t.h, 396
 - op-3t.h, 398
- T0_REG
 - i386-regs.h, 380
- T1_REG
 - i386-regs.h, 380
- T2_REG
 - i386-regs.h, 380
- T_NONE

- dyn-rec.c, 335
- tagaddr
 - dc_set, 70
- tagaddr_mask
 - ic, 127
- tags
 - ic, 127
- taken
 - bpb_entry, 30
 - branchstat, 32
- TARGET_SIGNAL_ALRM
 - rsp-server.c, 541
- TARGET_SIGNAL_BUS
 - rsp-server.c, 541
- TARGET_SIGNAL_FPE
 - rsp-server.c, 541
- TARGET_SIGNAL_ILL
 - rsp-server.c, 541
- TARGET_SIGNAL_INT
 - rsp-server.c, 541
- TARGET_SIGNAL_NONE
 - rsp-server.c, 541
- TARGET_SIGNAL_PWR
 - rsp-server.c, 541
- TARGET_SIGNAL_SEGV
 - rsp-server.c, 541
- TARGET_SIGNAL_TRAP
 - rsp-server.c, 541
- TARGET_SIGNAL_USR2
 - rsp-server.c, 541
- target_signal
 - rsp-server.c, 540
- tcp.c
 - tcp_channel_ops, 670
 - tcp_init, 670
 - tcp_open, 670
 - tcp_read, 670
 - tcp_write, 670
 - wait_for_tcp_connect, 670
- tcp.h
 - tcp_channel_ops, 671
- tcp_channel, 176
 - connected, 176
 - fds, 176
 - nonblocking, 176
 - port_number, 176
 - socket_fd, 176
- tcp_channel_ops
 - tcp.c, 670
 - tcp.h, 671
- tcp_init
 - tcp.c, 670
- tcp_level
 - gdbcomm.c, 534
- vapi.c, 870
- tcp_open
 - tcp.c, 670
- tcp_read
 - tcp.c, 670
- tcp_write
 - tcp.c, 670
- temp
 - vapi_handler, 178
- TEST_FLAG
 - fields.h, 721
- text_start
 - COFF_AOUTHDR, 40
- TFLAG_DST
 - dyn-rec.c, 335
- TFLAG_SAVED
 - dyn-rec.c, 335
- TFLAG_SOURCED
 - dyn-rec.c, 335
- TFLAG_SRC
 - dyn-rec.c, 335
- tflags
 - op_queue, 158
- ti
 - or32.c, 436
- tick.c
 - cycles_start, 842
 - sched_timer_job, 840
 - spr_read_ttc, 840
 - spr_write_ttc, 840
 - spr_write_ttmr, 840
 - tick_count, 842
 - tick_one_shot, 841
 - tick_raise_except, 841
 - tick_reset, 841
 - tick_restart, 841
- tick.h
 - spr_read_ttc, 843
 - spr_write_ttc, 843
 - spr_write_ttmr, 844
 - tick_reset, 844
- tick/tick.c, 839
- tick/tick.h, 843
- tick_count
 - tick.c, 842
- tick_one_shot
 - tick.c, 841
- tick_raise_except
 - tick.c, 841
- tick_reset
 - tick.c, 841
 - tick.h, 844
- tick_restart
 - tick.c, 841

- tim
 - cuc_bb, 63
- tim_comp
 - cuc.c, 462
- time
 - sched_entry, 167
- time_point
 - runtime, 165
- timing_table
 - timings.c, 509
- timings
 - _cuc_func, 16
- timings.c
 - analyse_timings, 508
 - bb_size, 508
 - cut_tree, 508
 - ii_size, 508
 - insn_size, 509
 - insn_time, 509
 - load_timing_table, 509
 - mark_cut, 509
 - max_bb_delay, 509
 - max_delay, 509
 - memory_delay, 509
 - new_bb_cycles, 509
 - recalc_cnts, 509
 - timing_table, 509
- timings_fn
 - config, 55
- tmp
 - _cuc_func, 16
 - cuc_bb, 63
 - cuc_insn, 65
- tmp_op
 - insn.c, 487
- tmp_opt
 - insn.c, 487
- tms
 - mc, 147
- to
 - cuc_conv, 64
- to_insn_num
 - sim-cmd.c, 798
- to_spr
 - spr_def, 173
- toplevel-mprofile.c, 845
 - main, 847
- toplevel-profile.c, 848
 - main, 850
- toplevel-support.c, 851
 - check_int, 852
 - ctrl_c, 853
 - reg_sim_reset, 853
 - sim_done, 854
 - sim_init, 854
 - sim_reset, 855
 - sim_reset_hooks, 856
- toplevel-support.h, 857
 - check_int, 857
 - ctrl_c, 858
 - reg_sim_reset, 858
 - sim_done, 859
 - sim_init, 859
 - sim_reset, 860
- toplevel.c, 862
 - main, 862
- total_size
 - dma_channel, 81
- TRACE
 - debug.h, 820
- trace.c
 - set_insnbrkpoint, 286
- trace.h
 - set_insnbrkpoint, 288
- TRACE_
 - debug.h, 820
- TRACE_ON
 - debug.h, 820
- trans_direction
 - dev_generic, 77
- trans_size
 - dev_generic, 77
- transl_error
 - parse.c, 277
- transl_table
 - parse.c, 277
- translate
 - parse.c, 276
- TRUE
 - gdbcomm.h, 535
- ts_bound
 - dyn_page, 89
- tsize
 - COFF_AOUTHDR, 40
- tty.c
 - baud_table, 673
 - DEFAULT_BAUD, 673
 - DEFAULT_TTY_DEVICE, 673
 - name, 673
 - parse_baud, 673
 - tty_channel_ops, 673
 - tty_init, 673
 - tty_open, 673
 - value, 673
- tty.h
 - tty_channel_ops, 674
- tty_channel, 177
 - fds, 177

- tty_channel_ops
 - tty.c, [673](#)
 - tty.h, [674](#)
- tty_init
 - tty.c, [673](#)
- tty_open
 - tty.c, [673](#)
- tx
 - eth_device, [113](#)
- tx_bd_num
 - eth_device, [113](#)
- tx_buff
 - eth_device, [113](#)
- tx_channel
 - eth_device, [113](#)
- txb
 - ata_host, [26](#)
- txbuf
 - dev_16450, [74](#)
- txbuf_full
 - dev_16450, [74](#)
- txbuf_head
 - dev_16450, [74](#)
- txbuf_tail
 - dev_16450, [74](#)
- txfd
 - eth_device, [113](#)
- txfile
 - eth_device, [113](#)
- txser
 - dev_16450, [74](#)
- type
 - ata_device, [22](#)
 - BMP_HEADER, [29](#)
 - config_param, [58](#)
 - cuc_bb, [63](#)
 - cuc_insn, [65](#)
 - mem_config, [150](#)
 - mp_entry, [154](#)
 - mprofentry_struct, [155](#)
 - reloc, [161](#)
- TYPE_FILE
 - atadevice.h, [642](#)
- TYPE_LOCAL
 - atadevice.h, [642](#)
- TYPE_NO_CONNECT
 - atadevice.h, [642](#)
- uart16550
 - dev_16450, [74](#)
- uart_16550
 - 16450.c, [605](#)
- uart_add_char
 - 16450.c, [605](#)
- UART_ADDR_SPACE
 - 16450.c, [598](#)
- uart_baseaddr
 - 16450.c, [605](#)
- UART_BREAK_COUNT
 - 16450.c, [598](#)
- uart_channel
 - 16450.c, [605](#)
- uart_char_clock
 - 16450.c, [605](#)
- UART_CHAR_TIMEOUT
 - 16450.c, [598](#)
- uart_check_char
 - 16450.c, [606](#)
- uart_check_rdi
 - 16450.c, [606](#)
- uart_check_rlsi
 - 16450.c, [606](#)
- uart_clear_int
 - 16450.c, [607](#)
- UART_CLOCK_DIVIDER
 - 16450.c, [598](#)
- UART_DLH
 - 16450.c, [598](#)
- UART_DLL
 - 16450.c, [598](#)
- uart_enabled
 - 16450.c, [607](#)
- UART_FCR
 - 16450.c, [598](#)
- UART_FCR_FIE
 - 16450.c, [598](#)
- UART_FCR_RRXFI
 - 16450.c, [598](#)
- UART_FCR_RTXFI
 - 16450.c, [598](#)
- UART_FGETC_SLOWDOWN
 - 16450.c, [598](#)
- UART_FIFO_TRIGGER
 - 16450.c, [598](#)
- UART_IER
 - 16450.c, [599](#)
- UART_IER_MSI
 - 16450.c, [601](#)
- UART_IER_RDI
 - 16450.c, [601](#)
- UART_IER_RLSI
 - 16450.c, [601](#)
- UART_IER_THRI
 - 16450.c, [601](#)
- UART_IIR
 - 16450.c, [601](#)
- UART_IIR_CTI
 - 16450.c, [601](#)

UART_IIR_ID
16450.c, 601

UART_IIR_MSI
16450.c, 601

UART_IIR_NO_INT
16450.c, 601

UART_IIR_RDI
16450.c, 601

UART_IIR_RLSI
16450.c, 601

UART_IIR_THRI
16450.c, 601

uart_int_cti
16450.c, 608

uart_int_msi
16450.c, 608

uart_int_rdi
16450.c, 608

uart_int_rlsi
16450.c, 609

uart_int_thri
16450.c, 609

uart_irq
16450.c, 610

uart_jitter
16450.c, 610

UART_LCR
16450.c, 601

UART_LCR_DLAB
16450.c, 601

UART_LCR_EPAR
16450.c, 601

UART_LCR_PARITY
16450.c, 601

UART_LCR_RESET
16450.c, 601

UART_LCR_SBC
16450.c, 601

UART_LCR_SPAR
16450.c, 601

UART_LCR_STOP
16450.c, 601

UART_LCR_WLEN5
16450.c, 601

UART_LCR_WLEN6
16450.c, 601

UART_LCR_WLEN7
16450.c, 601

UART_LCR_WLEN8
16450.c, 601

uart_loopback
16450.c, 610

UART_LSR
16450.c, 601

UART_LSR_BREAK
16450.c, 601

UART_LSR_FRAME
16450.c, 601

UART_LSR_OVRRUN
16450.c, 601

UART_LSR_PARITY
16450.c, 601

UART_LSR_RDRDY
16450.c, 601

UART_LSR_RXERR
16450.c, 601

UART_LSR_TXBUFE
16450.c, 601

UART_LSR_TXSERE
16450.c, 601

UART_MAX_FIFO_LEN
16450.c, 601

UART_MCR
16450.c, 601

UART_MCR_AUX1
16450.c, 602

UART_MCR_AUX2
16450.c, 602

UART_MCR_DTR
16450.c, 602

UART_MCR_LOOP
16450.c, 602

UART_MCR_RTS
16450.c, 602

UART_MSR
16450.c, 602

UART_MSR_CTS
16450.c, 602

UART_MSR_DCD
16450.c, 602

UART_MSR_DCTS
16450.c, 602

UART_MSR_DDCD
16450.c, 602

UART_MSR_DDSR
16450.c, 602

UART_MSR_DSR
16450.c, 602

UART_MSR_RI
16450.c, 602

UART_MSR_TERI
16450.c, 602

uart_newway
16450.c, 610

uart_next_int
16450.c, 611

uart_read_byte
16450.c, 611

- uart_rcv_break
 - 16450.c, [611](#)
- uart_rcv_break_start
 - 16450.c, [611](#)
- uart_rcv_break_stop
 - 16450.c, [612](#)
- uart_rcv_char
 - 16450.c, [612](#)
- uart_reset
 - 16450.c, [612](#)
 - 16450.h, [620](#)
- UART_RXBUF
 - 16450.c, [602](#)
- uart_sched_rcv_check
 - 16450.c, [613](#)
- UART_SCR
 - 16450.c, [602](#)
- uart_sec_end
 - 16450.c, [613](#)
- uart_sec_start
 - 16450.c, [614](#)
- uart_send_break
 - 16450.c, [614](#)
- uart_status
 - 16450.c, [615](#)
 - 16450.h, [620](#)
- uart_tx_send
 - 16450.c, [615](#)
- UART_TXBUF
 - 16450.c, [602](#)
- UART_VALID_FCR
 - 16450.c, [602](#)
- UART_VALID_IER
 - 16450.c, [602](#)
- UART_VALID_IIR
 - 16450.c, [602](#)
- UART_VALID_LCR
 - 16450.c, [602](#)
- UART_VALID_LSR
 - 16450.c, [602](#)
- UART_VALID_MCR
 - 16450.c, [602](#)
- UART_VALID_MSR
 - 16450.c, [602](#)
- UART_VAPI_BUF_LEN
 - 16450.c, [602](#)
- uart_vapi_cmd
 - 16450.c, [615](#)
- uart_vapi_id
 - 16450.c, [616](#)
- uart_vapi_read
 - 16450.c, [616](#)
- uart_write_byte
 - 16450.c, [616](#)
- ULONGEST
 - abstract.h, [231](#)
- unmark_tree
 - insn.c, [487](#)
- unroll
 - cuc_timings, [69](#)
- unrolled
 - cuc_bb, [63](#)
- uorreg_t
 - arch.h, [291](#)
- upd_reg_from_t
 - op-support.h, [410](#)
- upd_sim_cycles
 - common-i386.h, [325](#)
- update_pc
 - execute.c, [375](#)
- used_regs
 - _cuc_func, [16](#)
- useless_x86
 - common-i386.h, [325](#)
- ustates
 - config, [55](#)
 - dmmu, [86](#)
 - ic, [127](#)
 - immu, [129](#)
- ustates_reload
 - ic, [127](#)
- val3232
 - common-i386.h, [325](#)
- val64
 - common-i386.h, [325](#)
- valid
 - dev_memarea, [79](#)
- value
 - dev_generic, [77](#)
 - tty.c, [673](#)
- vapi
 - config, [55](#)
 - dev_16450, [74](#)
 - runtime, [165](#)
- vapi.c
 - add_handler, [866](#)
 - fds, [870](#)
 - find_handler, [866](#)
 - get_server_socket, [866](#)
 - handler_fits_id, [867](#)
 - nfds, [870](#)
 - nhandlers, [870](#)
 - read_packet, [867](#)
 - rebuild_fds, [867](#)
 - reg_vapi_sec, [867](#)
 - server_fd, [870](#)
 - server_request, [867](#)

- serverIP, 870
- tcp_level, 870
- vapi_check, 867
- vapi_done, 868
- vapi_enabled, 868
- vapi_hide_device_id, 868
- vapi_init, 868
- vapi_install_handler, 868
- vapi_install_multi_handler, 869
- vapi_log_enabled, 869
- vapi_log_fn, 869
- vapi_num_unconnected, 869
- vapi_read_stream, 869
- vapi_request, 869
- vapi_send, 869
- vapi_server_port, 870
- vapi_write_log_file, 870
- vapi_write_stream, 870
- write_packet, 870
- vapi.h
 - reg_vapi_sec, 872
 - VAPI_COMMAND_END, 871
 - VAPI_COMMAND_REQUEST, 871
 - VAPI_COMMAND_SEND, 871
 - vapi_check, 872
 - VAPI_COMMAND, 871
 - vapi_done, 872
 - vapi_init, 872
 - vapi_install_handler, 873
 - vapi_install_multi_handler, 873
 - VAPI_MAX_DEVID, 871
 - vapi_num_unconnected, 873
 - vapi_send, 873
 - vapi_write_log_file, 873
- vapi/vapi.c, 865
- vapi/vapi.h, 871
- VAPI_COMMAND_END
 - vapi.h, 871
- VAPI_COMMAND_REQUEST
 - vapi.h, 871
- VAPI_COMMAND_SEND
 - vapi.h, 871
- vapi_buf
 - dev_16450, 74
- vapi_buf_head_ptr
 - dev_16450, 74
- vapi_buf_tail_ptr
 - dev_16450, 74
- vapi_check
 - vapi.c, 867
 - vapi.h, 872
- VAPI_COMMAND
 - vapi.h, 871
- vapi_done
 - vapi.c, 868
 - vapi.h, 872
- vapi_enabled
 - vapi.c, 868
- vapi_file
 - runtime, 165
- vapi_fn
 - config, 55
- vapi_handler, 178
 - base_id, 178
 - fd, 178
 - next, 178
 - num_ids, 178
 - priv_dat, 178
 - read_func, 178
 - temp, 178
- vapi_hide_device_id
 - vapi.c, 868
- vapi_id
 - config, 55
 - dev_16450, 74
 - dma_controller, 84
- vapi_init
 - vapi.c, 868
 - vapi.h, 872
- vapi_install_handler
 - vapi.c, 868
 - vapi.h, 873
- vapi_install_multi_handler
 - vapi.c, 869
 - vapi.h, 873
- vapi_log_enabled
 - vapi.c, 869
- vapi_log_fn
 - vapi.c, 869
- VAPI_MAX_DEVID
 - vapi.h, 871
- vapi_num_unconnected
 - vapi.c, 869
 - vapi.h, 873
- vapi_read_stream
 - vapi.c, 869
- vapi_request
 - vapi.c, 869
- vapi_send
 - vapi.c, 869
 - vapi.h, 873
- vapi_server_port
 - vapi.c, 870
- vapi_write_log_file
 - vapi.c, 870
 - vapi.h, 873
- vapi_write_stream
 - vapi.c, 870

- vbar
 - vga_state, 180
- vbindex
 - vga_state, 180
- verbose
 - config, 55
- verify_memoryarea
 - abstract.c, 228
 - abstract.h, 243
- verilog.c
 - branch_index, 511
 - find_lsc_index, 511
 - func_index, 511
 - GEN, 511
 - generate_main, 511
 - output_verilog, 511
 - print_deps, 511
 - print_insn_v, 512
 - print_op_v, 512
 - print_turn_off_dep, 512
- verilog.h
 - generate_main, 513
 - output_verilog, 513
- vga.c
 - reg_vga_sec, 760
 - VGA_ADDR_SPACE, 760
 - vga_baseaddr, 761
 - VGA_CLUTA, 760
 - VGA_CLUTB, 760
 - VGA_CTRL, 760
 - VGA_CTRL_CD, 760
 - VGA_CTRL_PC, 760
 - VGA_CTRL_VEN, 760
 - vga_dump_image, 761
 - vga_enabled, 761
 - vga_filename, 761
 - VGA_HTIM, 760
 - VGA_HVLEN, 760
 - vga_irq, 761
 - vga_job, 762
 - VGA_MASK, 760
 - vga_read32, 762
 - vga_refresh_rate, 762
 - vga_reset, 762
 - vga_sec_end, 762
 - vga_sec_start, 763
 - VGA_STAT, 760
 - VGA_VBARA, 760
 - VGA_VBARB, 760
 - VGA_VTIM, 760
 - vga_write32, 763
- vga.h
 - reg_vga_sec, 764
- VGA_ADDR_SPACE
 - vga.c, 760
- vga_baseaddr
 - vga.c, 761
- VGA_CLUTA
 - vga.c, 760
- VGA_CLUTB
 - vga.c, 760
- VGA_CTRL
 - vga.c, 760
- VGA_CTRL_CD
 - vga.c, 760
- VGA_CTRL_PC
 - vga.c, 760
- VGA_CTRL_VEN
 - vga.c, 760
- vga_dump_image
 - vga.c, 761
- vga_enabled
 - vga.c, 761
- vga_filename
 - vga.c, 761
- VGA_HTIM
 - vga.c, 760
- VGA_HVLEN
 - vga.c, 760
- vga_irq
 - vga.c, 761
- vga_job
 - vga.c, 762
- VGA_MASK
 - vga.c, 760
- vga_read32
 - vga.c, 762
- vga_refresh_rate
 - vga.c, 762
- vga_reset
 - vga.c, 762
- vga_sec_end
 - vga.c, 762
- vga_sec_start
 - vga.c, 763
- VGA_STAT
 - vga.c, 760
- vga_state, 179
 - baseaddr, 180
 - ctrl, 180
 - enabled, 180
 - filename, 180
 - hlen, 180
 - htim, 180
 - irq, 180
 - palette, 180
 - pics, 180
 - pindex, 180

- refresh_rate, 180
- stat, 180
- vbar, 180
- vbindx, 180
- vlen, 180
- vtim, 180
- VGA_VBARA
 - vga.c, 760
- VGA_VBARB
 - vga.c, 760
- VGA_VTIM
 - vga.c, 760
- vga_write32
 - vga.c, 763
- vlen
 - vga_state, 180
- vpn_mask
 - dmmu, 86
 - immu, 129
- vstamp
 - COFF_AOUTHDR, 40
- vtim
 - vga_state, 180
- wait_for_tcp_connect
 - tcp.c, 670
- waiting_for_dma
 - eth_device, 113
- WARN
 - debug.h, 820
- WARN_
 - debug.h, 820
- WARN_ON
 - debug.h, 820
- WARNING
 - cpu-config.c, 207
- way
 - bpb_entry, 30
 - btic_entry, 34
 - dc_set, 70
- wfds
 - eth_device, 113
- width
 - INFOHEADER, 131
- word_enabled
 - dev_generic, 77
- words_transferred
 - dma_channel, 81
- working
 - eth_device, 113
- WP_ACCESS
 - rsp-server.c, 540
- WP_READ
 - rsp-server.c, 540
- WP_WRITE
 - rsp-server.c, 540
- write
 - channel_ops, 39
- WRITE_BUFFER
 - atacmd.h, 624
- write_dat16
 - mem_ops, 152
- write_dat32
 - mem_ops, 152
- write_dat8
 - mem_ops, 152
- WRITE_DMA
 - atacmd.h, 624
- WRITE_DMA_QUEUED
 - atacmd.h, 624
- WRITE_MULTIPLE
 - atacmd.h, 624
- write_packet
 - vapi.c, 870
- WRITE_SECTOR
 - atacmd.h, 624
- WRITE_SECTORS
 - atacmd.h, 624
- write_to_reg
 - generate.c, 379
- write_up
 - config, 55
- writfunc16
 - mem_ops, 152
- writfunc32
 - mem_ops, 152
- writfunc8
 - mem_ops, 152
- writehit
 - cachestats_entry, 36
- writemiss
 - cachestats_entry, 36
- writeprog32
 - mem_ops, 152
- writeprog32_dat
 - mem_ops, 152
- writeprog8
 - mem_ops, 152
- writeprog8_dat
 - mem_ops, 152
- x_ary
 - COFF_auxent, 43
- x_dimen
 - COFF_auxent, 43
- x_endndx
 - COFF_auxent, 43
- x_fcn

- COFF_auxent, 43
- x_fcary
 - COFF_auxent, 43
- x_file
 - COFF_auxent, 43
- x_fname
 - COFF_auxent, 43
- x_fsize
 - COFF_auxent, 43
- x_inno
 - COFF_auxent, 43
- x_innopt
 - COFF_auxent, 43
- x_insz
 - COFF_auxent, 43
- x_misc
 - COFF_auxent, 43
- x_n
 - COFF_auxent, 43
- x_nlinno
 - COFF_auxent, 43
- x_nreloc
 - COFF_auxent, 43
- x_offset
 - COFF_auxent, 43
- x_scn
 - COFF_auxent, 43
- x_scnlen
 - COFF_auxent, 43
- x_size
 - COFF_auxent, 43
- x_sym
 - COFF_auxent, 43
- x_tagndx
 - COFF_auxent, 43
- x_tv
 - COFF_auxent, 43
- x_tvfill
 - COFF_auxent, 43
- x_tvlen
 - COFF_auxent, 43
- x_tvndx
 - COFF_auxent, 43
- x_tvran
 - COFF_auxent, 43
- x_zeroes
 - COFF_auxent, 43
- xchg_insn
 - load.c, 497
- xglue
 - dyn-rec.h, 353
- xref
 - op_queue, 158
- xresolution
 - INFOHEADER, 131
- xterm.c
 - basename, 676
 - MAX_XTERM_ARGS, 676
 - xterm_channel_ops, 676
 - xterm_close, 676
 - xterm_init, 676
 - xterm_open, 676
- xterm.h
 - xterm_channel_ops, 677
- xterm_channel, 181
 - argv, 181
 - fds, 181
 - pid, 181
- xterm_channel_ops
 - xterm.c, 676
 - xterm.h, 677
- xterm_close
 - xterm.c, 676
- xterm_init
 - xterm.c, 676
- xterm_open
 - xterm.c, 676
- yresolution
 - INFOHEADER, 131