

Or1ksim: The OpenRISC 1000 Architectural Simulator

Generated by Doxygen 1.5.6

Sun Oct 12 09:05:05 2008

Contents

1	Orlksim: the OpenRISC 1000 Architectural Simulator	1
1.1	About	1
1.2	Installation	1
1.3	Documentation	1
1.4	Copying	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	7
3.1	File List	7
4	Data Structure Documentation	11
4.1	_csm_list Struct Reference	11
4.1.1	Field Documentation	12
4.1.1.1	ref	12
4.1.1.2	cnt	12
4.1.1.3	cmovs	12
4.1.1.4	size	12
4.1.1.5	osize	12
4.1.1.6	cmatch	12
4.1.1.7	dead	12
4.1.1.8	ninsn	12
4.1.1.9	from	12
4.1.1.10	next	12
4.2	_cuc_func Struct Reference	13
4.2.1	Field Documentation	14
4.2.1.1	num_bb	14
4.2.1.2	bb	14

4.2.1.3	saved_regs	14
4.2.1.4	lur	14
4.2.1.5	used_regs	14
4.2.1.6	nmsched	14
4.2.1.7	msched	14
4.2.1.8	mtype	14
4.2.1.9	num_init_bb	14
4.2.1.10	init_bb_reloc	14
4.2.1.11	orig_time	14
4.2.1.12	num_runs	14
4.2.1.13	timings	14
4.2.1.14	start_addr	14
4.2.1.15	end_addr	14
4.2.1.16	memory_order	14
4.2.1.17	nfdeps	14
4.2.1.18	fdeps	14
4.2.1.19	tmp	14
4.3	_dep_list_t Struct Reference	15
4.3.1	Field Documentation	15
4.3.1.1	ref	15
4.3.1.2	next	15
4.4	archf Struct Reference	16
4.4.1	Field Documentation	16
4.4.1.1	get_real_func_len	16
4.4.1.2	gen_reloc	16
4.4.1.3	gen_func_reloc	16
4.5	ata_device Struct Reference	17
4.5.1	Field Documentation	20
4.5.1.1	host	20
4.5.1.2	dev	20
4.5.1.3	pio_mode	20
4.5.1.4	dma_mode	20
4.5.1.5	dbuf	20
4.5.1.6	dbuf_ptr	20
4.5.1.7	dbuf_cnt	20
4.5.1.8	state	20

4.5.1.9	heads_per_cylinder	20
4.5.1.10	sectors_per_track	20
4.5.1.11	lba	20
4.5.1.12	nr_sect	20
4.5.1.13	end_t_func	20
4.5.1.14	internals	20
4.5.1.15	command	20
4.5.1.16	cylinder_low	20
4.5.1.17	cylinder_high	20
4.5.1.18	device_control	20
4.5.1.19	device_head	20
4.5.1.20	error	20
4.5.1.21	features	20
4.5.1.22	sector_count	20
4.5.1.23	sector_number	20
4.5.1.24	status	20
4.5.1.25	dataport_i	20
4.5.1.26	regs	20
4.5.1.27	iordy	20
4.5.1.28	intrq	20
4.5.1.29	dmarq	20
4.5.1.30	pdiagi	20
4.5.1.31	pdiago	20
4.5.1.32	daspi	20
4.5.1.33	daspo	20
4.5.1.34	sigs	20
4.5.1.35	file	20
4.5.1.36	stream	20
4.5.1.37	type	20
4.5.1.38	size	20
4.5.1.39	size_sect	20
4.5.1.40	packet	20
4.5.1.41	heads	20
4.5.1.42	sectors	20
4.5.1.43	firmware	20
4.5.1.44	mwdma	20

4.5.1.45	pio	20
4.5.1.46	conf	20
4.6	ata_devices Struct Reference	22
4.6.1	Field Documentation	22
4.6.1.1	device	22
4.7	ata_host Struct Reference	23
4.7.1	Field Documentation	24
4.7.1.1	enabled	24
4.7.1.2	baseaddr	24
4.7.1.3	mem	24
4.7.1.4	irq	24
4.7.1.5	dev_id	24
4.7.1.6	rev	24
4.7.1.7	dev_sel	24
4.7.1.8	pio_mode0_t1	24
4.7.1.9	pio_mode0_t2	24
4.7.1.10	pio_mode0_t4	24
4.7.1.11	pio_mode0_teoc	24
4.7.1.12	dma_mode0_tm	24
4.7.1.13	dma_mode0_td	24
4.7.1.14	dma_mode0_teoc	24
4.7.1.15	ctrl	24
4.7.1.16	stat	24
4.7.1.17	pctr	24
4.7.1.18	pftr0	24
4.7.1.19	pftr1	24
4.7.1.20	dtr0	24
4.7.1.21	dtr1	24
4.7.1.22	txb	24
4.7.1.23	rxb	24
4.7.1.24	regs	24
4.7.1.25	devices	24
4.8	bff Struct Reference	26
4.8.1	Field Documentation	26
4.8.1.1	open_obj	26
4.8.1.2	close_obj	26

4.8.1.3	get_func_name	26
4.8.1.4	get_func_start	26
4.8.1.5	get_func_len	26
4.8.1.6	get_func_reloc	26
4.9	BMP_HEADER Struct Reference	27
4.9.1	Field Documentation	27
4.9.1.1	type	27
4.9.1.2	size	27
4.9.1.3	reserved1	27
4.9.1.4	reserved2	27
4.9.1.5	offset	27
4.10	bpb_entry Struct Reference	28
4.10.1	Field Documentation	28
4.10.1.1	addr	28
4.10.1.2	taken	28
4.10.1.3	lru	28
4.10.1.4	way	28
4.11	bpbstat Struct Reference	29
4.11.1	Field Documentation	29
4.11.1.1	hit	29
4.11.1.2	miss	29
4.11.1.3	correct	29
4.11.1.4	incorrect	29
4.12	branchstat Struct Reference	30
4.12.1	Field Documentation	30
4.12.1.1	taken	30
4.12.1.2	nottaken	30
4.12.1.3	forward	30
4.12.1.4	backward	30
4.13	breakpoint_entry Struct Reference	31
4.13.1	Detailed Description	31
4.13.2	Field Documentation	31
4.13.2.1	addr	31
4.13.2.2	next	31
4.14	btic_entry Struct Reference	32
4.14.1	Field Documentation	32

4.14.1.1	addr	32
4.14.1.2	lru	32
4.14.1.3	insn	32
4.14.1.4	way	32
4.15	bticstat Struct Reference	33
4.15.1	Field Documentation	33
4.15.1.1	hit	33
4.15.1.2	miss	33
4.16	cachestats_entry Struct Reference	34
4.16.1	Field Documentation	34
4.16.1.1	readhit	34
4.16.1.2	readmiss	34
4.16.1.3	writehit	34
4.16.1.4	writemiss	34
4.17	channel Struct Reference	35
4.17.1	Detailed Description	35
4.17.2	Field Documentation	35
4.17.2.1	ops	35
4.17.2.2	data	35
4.18	channel_factory Struct Reference	36
4.18.1	Field Documentation	36
4.18.1.1	name	36
4.18.1.2	ops	36
4.18.1.3	next	36
4.19	channel_ops Struct Reference	37
4.19.1	Detailed Description	37
4.19.2	Field Documentation	37
4.19.2.1	init	37
4.19.2.2	open	37
4.19.2.3	close	37
4.19.2.4	read	37
4.19.2.5	write	37
4.19.2.6	free	37
4.19.2.7	isok	37
4.19.2.8	status	37
4.20	COFF_AOUTHDR Struct Reference	38

4.20.1	Field Documentation	38
4.20.1.1	magic	38
4.20.1.2	vstamp	38
4.20.1.3	tsize	38
4.20.1.4	dsize	38
4.20.1.5	bsize	38
4.20.1.6	entry	38
4.20.1.7	text_start	38
4.20.1.8	data_start	38
4.21	COFF_auxent Union Reference	39
4.21.1	Field Documentation	41
4.21.1.1	x_tagndx	41
4.21.1.2	x_inno	41
4.21.1.3	x_size	41
4.21.1.4	x_lnsz	41
4.21.1.5	x_fsize	41
4.21.1.6	x_misc	41
4.21.1.7	x_innoptr	41
4.21.1.8	x_endndx	41
4.21.1.9	x_fcn	41
4.21.1.10	x_dimen	41
4.21.1.11	x_ary	41
4.21.1.12	x_fcary	41
4.21.1.13	x_tvndx	41
4.21.1.14	x_sym	41
4.21.1.15	x_fname	41
4.21.1.16	x_zeroes	41
4.21.1.17	x_offset	41
4.21.1.18	x_n	41
4.21.1.19	x_file	41
4.21.1.20	x_scrlen	41
4.21.1.21	x_nreloc	41
4.21.1.22	x_nlinno	41
4.21.1.23	x_scn	41
4.21.1.24	x_tvfill	41
4.21.1.25	x_tvlen	41

4.21.1.26	x_tvrn	41
4.21.1.27	x_tv	41
4.22	COFF_filehdr Struct Reference	43
4.22.1	Field Documentation	43
4.22.1.1	f_magic	43
4.22.1.2	f_nsens	43
4.22.1.3	f_timdat	43
4.22.1.4	f_symptr	43
4.22.1.5	f_nsyms	43
4.22.1.6	f_opthdr	43
4.22.1.7	f_flags	43
4.23	COFF_lineno Struct Reference	44
4.23.1	Field Documentation	44
4.23.1.1	l_symndx	44
4.23.1.2	l_paddr	44
4.23.1.3	l_addr	44
4.23.1.4	l_inno	44
4.24	COFF_reloc Struct Reference	45
4.24.1	Field Documentation	45
4.24.1.1	r_vaddr	45
4.24.1.2	r_symndx	45
4.24.1.3	r_type	45
4.25	COFF_scnhdr Struct Reference	46
4.25.1	Field Documentation	46
4.25.1.1	s_name	46
4.25.1.2	s_paddr	46
4.25.1.3	s_vaddr	46
4.25.1.4	s_size	46
4.25.1.5	s_scnptr	46
4.25.1.6	s_relptr	46
4.25.1.7	s_innoptr	46
4.25.1.8	s_nreloc	46
4.25.1.9	s_nlnno	46
4.25.1.10	s_flags	46
4.26	COFF_slib Struct Reference	47
4.26.1	Field Documentation	47

4.26.1.1	sl_entsz	47
4.26.1.2	sl_pathndx	47
4.27	COFF_syment Struct Reference	48
4.27.1	Field Documentation	48
4.27.1.1	e_name	48
4.27.1.2	e_zeroes	48
4.27.1.3	e_offset	48
4.27.1.4	e	48
4.27.1.5	e	48
4.27.1.6	e_value	48
4.27.1.7	e_scnnum	48
4.27.1.8	e_type	48
4.27.1.9	e_sclass	48
4.27.1.10	e_numaux	48
4.28	config Struct Reference	49
4.28.1	Detailed Description	50
4.28.2	Field Documentation	52
4.28.2.1	class_ptr	52
4.28.2.2	read_up	52
4.28.2.3	write_up	52
4.28.2.4	ext	52
4.28.2.5	debug	52
4.28.2.6	verbose	52
4.28.2.7	profile	52
4.28.2.8	prof_fn	52
4.28.2.9	mprofile	52
4.28.2.10	mprof_fn	52
4.28.2.11	history	52
4.28.2.12	exe_log	52
4.28.2.13	exe_log_type	52
4.28.2.14	exe_log_start	52
4.28.2.15	exe_log_end	52
4.28.2.16	exe_log_marker	52
4.28.2.17	exe_log_fn	52
4.28.2.18	clkcycle_ps	52
4.28.2.19	sim	52

4.28.2.20	enabled	52
4.28.2.21	server_port	52
4.28.2.22	log_enabled	52
4.28.2.23	hide_device_id	52
4.28.2.24	vapi_fn	52
4.28.2.25	vapi	52
4.28.2.26	timings_fn	52
4.28.2.27	memory_order	52
4.28.2.28	calling_convention	52
4.28.2.29	enable_bursts	52
4.28.2.30	no_multicycle	52
4.28.2.31	cuc	52
4.28.2.32	superscalar	52
4.28.2.33	hazards	52
4.28.2.34	dependstats	52
4.28.2.35	sbuf_len	52
4.28.2.36	cpu	52
4.28.2.37	nways	52
4.28.2.38	nsets	52
4.28.2.39	blocksize	52
4.28.2.40	ustates	52
4.28.2.41	store_missdelay	52
4.28.2.42	store_hitdelay	52
4.28.2.43	load_missdelay	52
4.28.2.44	load_hitdelay	52
4.28.2.45	dc	52
4.28.2.46	pic	52
4.28.2.47	pm	52
4.28.2.48	sbp_bnf_fwd	52
4.28.2.49	sbp_bf_fwd	52
4.28.2.50	btic	52
4.28.2.51	missdelay	52
4.28.2.52	hitdelay	52
4.28.2.53	bpb	52
4.28.2.54	gdb_enabled	52
4.28.2.55	vapi_id	52

4.28.2.56 debug	52
4.29 config::pic Struct Reference	54
4.29.1 Field Documentation	54
4.29.1.1 enabled	54
4.29.1.2 edge_trigger	54
4.30 config_param Struct Reference	55
4.30.1 Field Documentation	55
4.30.1.1 name	55
4.30.1.2 type	55
4.30.1.3 func	55
4.30.1.4 next	55
4.31 config_section Struct Reference	56
4.31.1 Field Documentation	56
4.31.1.1 name	56
4.31.1.2 sec_start	56
4.31.1.3 sec_end	56
4.31.1.4 dat	56
4.31.1.5 params	56
4.31.1.6 next	56
4.32 cpu_state Struct Reference	57
4.32.1 Detailed Description	57
4.32.2 Field Documentation	57
4.32.2.1 reg	57
4.32.2.2 sprs	57
4.32.2.3 insn_ea	58
4.32.2.4 delay_insn	58
4.32.2.5 pc	58
4.32.2.6 pc_delay	58
4.32.2.7 pic_lines	58
4.32.2.8 iqueue	58
4.32.2.9 icomplet	58
4.33 cuc_bb Struct Reference	59
4.33.1 Field Documentation	60
4.33.1.1 type	60
4.33.1.2 first	60
4.33.1.3 last	60

4.33.1.4	prev	60
4.33.1.5	next	60
4.33.1.6	tmp	60
4.33.1.7	insn	60
4.33.1.8	ninsn	60
4.33.1.9	last_used_reg	60
4.33.1.10	mdep	60
4.33.1.11	nmemory	60
4.33.1.12	cnt	60
4.33.1.13	unrolled	60
4.33.1.14	ntim	60
4.33.1.15	tim	60
4.33.1.16	selected_tim	60
4.34	cuc_conv Struct Reference	61
4.34.1	Field Documentation	61
4.34.1.1	from	61
4.34.1.2	to	61
4.35	cuc_insn Struct Reference	62
4.35.1	Field Documentation	62
4.35.1.1	type	62
4.35.1.2	index	62
4.35.1.3	opt	62
4.35.1.4	op	62
4.35.1.5	dep	62
4.35.1.6	insn	62
4.35.1.7	disasm	62
4.35.1.8	max	62
4.35.1.9	tmp	62
4.36	cuc_known_insn Struct Reference	63
4.36.1	Field Documentation	63
4.36.1.1	name	63
4.36.1.2	comutative	63
4.36.1.3	rtl	63
4.37	cuc_shared_item Struct Reference	64
4.37.1	Field Documentation	64
4.37.1.1	ref	64

4.37.1.2	cmatch	64
4.38	cuc_timing_table Struct Reference	65
4.38.1	Field Documentation	65
4.38.1.1	delay	65
4.38.1.2	size	65
4.38.1.3	delayi	65
4.38.1.4	sizei	65
4.39	cuc_timings Struct Reference	66
4.39.1	Field Documentation	66
4.39.1.1	b	66
4.39.1.2	preroll	66
4.39.1.3	unroll	66
4.39.1.4	nshared	66
4.39.1.5	shared	66
4.39.1.6	new_time	66
4.39.1.7	size	66
4.40	dc_set Struct Reference	67
4.40.1	Field Documentation	67
4.40.1.1	line	67
4.40.1.2	tagaddr	67
4.40.1.3	lru	67
4.40.1.4	way	67
4.41	dev_16450 Struct Reference	68
4.41.1	Field Documentation	71
4.41.1.1	txbuf	71
4.41.1.2	rxbuf	71
4.41.1.3	dll	71
4.41.1.4	dlh	71
4.41.1.5	ier	71
4.41.1.6	iir	71
4.41.1.7	fcr	71
4.41.1.8	lcr	71
4.41.1.9	mcr	71
4.41.1.10	lsr	71
4.41.1.11	msr	71
4.41.1.12	scr	71

4.41.1.13	regs	71
4.41.1.14	txser	71
4.41.1.15	rxser	71
4.41.1.16	loopback	71
4.41.1.17	iregs	71
4.41.1.18	txbuf_head	71
4.41.1.19	txbuf_tail	71
4.41.1.20	rxbuf_head	71
4.41.1.21	rxbuf_tail	71
4.41.1.22	txbuf_full	71
4.41.1.23	rxbuf_full	71
4.41.1.24	receiveing	71
4.41.1.25	recv_break	71
4.41.1.26	ints	71
4.41.1.27	istat	71
4.41.1.28	char_clks	71
4.41.1.29	skew	71
4.41.1.30	vapi	71
4.41.1.31	vapi_buf	71
4.41.1.32	vapi_buf_head_ptr	71
4.41.1.33	vapi_buf_tail_ptr	71
4.41.1.34	fifo_len	71
4.41.1.35	channel	71
4.41.1.36	enabled	71
4.41.1.37	jitter	71
4.41.1.38	baseaddr	71
4.41.1.39	irq	71
4.41.1.40	vapi_id	71
4.41.1.41	uart16550	71
4.41.1.42	channel_str	71
4.42	dev_generic Struct Reference	73
4.42.1	Detailed Description	73
4.42.2	Member Enumeration Documentation	73
4.42.2.1	"@48	73
4.42.2.2	"@49	73
4.42.3	Field Documentation	74

4.42.3.1	trans_direction	74
4.42.3.2	trans_size	74
4.42.3.3	value	74
4.42.3.4	enabled	74
4.42.3.5	byte_enabled	74
4.42.3.6	hw_enabled	74
4.42.3.7	word_enabled	74
4.42.3.8	name	74
4.42.3.9	baseaddr	74
4.42.3.10	size	74
4.43	dev_memarea Struct Reference	75
4.43.1	Detailed Description	75
4.43.2	Field Documentation	76
4.43.2.1	next	76
4.43.2.2	addr_mask	76
4.43.2.3	addr_compare	76
4.43.2.4	size	76
4.43.2.5	size_mask	76
4.43.2.6	valid	76
4.43.2.7	log	76
4.43.2.8	ops	76
4.43.2.9	direct_ops	76
4.44	dma_channel Struct Reference	77
4.44.1	Field Documentation	78
4.44.1.1	controller	78
4.44.1.2	channel_number	78
4.44.1.3	channel_mask	78
4.44.1.4	referenced	78
4.44.1.5	load_next_descriptor_when_done	78
4.44.1.6	current_descriptor	78
4.44.1.7	source	78
4.44.1.8	destination	78
4.44.1.9	source_mask	78
4.44.1.10	destination_mask	78
4.44.1.11	chunk_size	78
4.44.1.12	total_size	78

4.44.1.13	words_transferred	78
4.44.1.14	csr	78
4.44.1.15	sz	78
4.44.1.16	a0	78
4.44.1.17	am0	78
4.44.1.18	a1	78
4.44.1.19	am1	78
4.44.1.20	desc	78
4.44.1.21	swptr	78
4.44.1.22	regs	78
4.44.1.23	dma_req_i	78
4.44.1.24	dma_ack_o	78
4.44.1.25	dma_nd_i	78
4.45	dma_controller Struct Reference	80
4.45.1	Field Documentation	81
4.45.1.1	enabled	81
4.45.1.2	baseaddr	81
4.45.1.3	irq	81
4.45.1.4	vapi_id	81
4.45.1.5	csr	81
4.45.1.6	int_msk_a	81
4.45.1.7	int_msk_b	81
4.45.1.8	int_src_a	81
4.45.1.9	int_src_b	81
4.45.1.10	regs	81
4.45.1.11	ch	81
4.45.1.12	next	81
4.46	dmmu Struct Reference	82
4.46.1	Field Documentation	83
4.46.1.1	enabled	83
4.46.1.2	nways	83
4.46.1.3	nsets	83
4.46.1.4	pagesize	83
4.46.1.5	pagesize_log2	83
4.46.1.6	page_offset_mask	83
4.46.1.7	page_mask	83

4.46.1.8	vpn_mask	83
4.46.1.9	lru_reload	83
4.46.1.10	set_mask	83
4.46.1.11	entrysize	83
4.46.1.12	ustates	83
4.46.1.13	missdelay	83
4.46.1.14	hitdelay	83
4.47	dmmustats_entry Struct Reference	84
4.47.1	Field Documentation	84
4.47.1.1	loads_tlbhit	84
4.47.1.2	loads_tlbmiss	84
4.47.1.3	loads_pagefaults	84
4.47.1.4	stores_tlbhit	84
4.47.1.5	stores_tlbmiss	84
4.47.1.6	stores_pagefaults	84
4.48	dstats_entry Struct Reference	85
4.48.1	Field Documentation	85
4.48.1.1	insn1	85
4.48.1.2	insn2	85
4.48.1.3	cnt_dynamic	85
4.48.1.4	depend	85
4.49	dyn_page Struct Reference	86
4.49.1	Field Documentation	86
4.49.1.1	or_page	86
4.49.1.2	host_page	86
4.49.1.3	host_len	86
4.49.1.4	dirty	86
4.49.1.5	delayr	86
4.49.1.6	ts_bound	86
4.49.1.7	locs	86
4.49.1.8	insns	86
4.49.1.9	insn_indexs	86
4.50	dynamic Struct Reference	87
4.50.1	Field Documentation	87
4.50.1.1	d_tag	87
4.50.1.2	d_val	87

4.50.1.3	d_ptr	87
4.50.1.4	d_un	87
4.51	elf32_hdr Struct Reference	88
4.51.1	Field Documentation	89
4.51.1.1	e_ident	89
4.51.1.2	e_type	89
4.51.1.3	e_machine	89
4.51.1.4	e_version	89
4.51.1.5	e_entry	89
4.51.1.6	e_phoff	89
4.51.1.7	e_shoff	89
4.51.1.8	e_flags	89
4.51.1.9	e_ehsize	89
4.51.1.10	e_phentsize	89
4.51.1.11	e_phnum	89
4.51.1.12	e_shentsize	89
4.51.1.13	e_shnum	89
4.51.1.14	e_shstrndx	89
4.52	elf32_note Struct Reference	90
4.52.1	Field Documentation	90
4.52.1.1	n_namesz	90
4.52.1.2	n_descsz	90
4.52.1.3	n_type	90
4.53	elf32_phdr Struct Reference	91
4.53.1	Field Documentation	91
4.53.1.1	p_type	91
4.53.1.2	p_offset	91
4.53.1.3	p_vaddr	91
4.53.1.4	p_paddr	91
4.53.1.5	p_filesz	91
4.53.1.6	p_memsz	91
4.53.1.7	p_flags	91
4.53.1.8	p_align	91
4.54	elf32_rel Struct Reference	92
4.54.1	Field Documentation	92
4.54.1.1	r_offset	92

4.54.1.2	r_info	92
4.55	elf32_rela Struct Reference	93
4.55.1	Field Documentation	93
4.55.1.1	r_offset	93
4.55.1.2	r_info	93
4.55.1.3	r_addend	93
4.56	elf32_shdr Struct Reference	94
4.56.1	Field Documentation	94
4.56.1.1	sh_name	94
4.56.1.2	sh_type	94
4.56.1.3	sh_flags	94
4.56.1.4	sh_addr	94
4.56.1.5	sh_offset	94
4.56.1.6	sh_size	94
4.56.1.7	sh_link	94
4.56.1.8	sh_info	94
4.56.1.9	sh_addralign	94
4.56.1.10	sh_entsize	94
4.57	elf32_sym Struct Reference	95
4.57.1	Field Documentation	95
4.57.1.1	st_name	95
4.57.1.2	st_value	95
4.57.1.3	st_size	95
4.57.1.4	st_info	95
4.57.1.5	st_other	95
4.57.1.6	st_shndx	95
4.58	Elf64_Dyn Struct Reference	96
4.58.1	Field Documentation	96
4.58.1.1	d_tag	96
4.58.1.2	d_val	96
4.58.1.3	d_ptr	96
4.58.1.4	d_un	96
4.59	elf64_hdr Struct Reference	97
4.59.1	Field Documentation	98
4.59.1.1	e_ident	98
4.59.1.2	e_type	98

4.59.1.3	e_machine	98
4.59.1.4	e_version	98
4.59.1.5	e_entry	98
4.59.1.6	e_phoff	98
4.59.1.7	e_shoff	98
4.59.1.8	e_flags	98
4.59.1.9	e_ehsize	98
4.59.1.10	e_phentsize	98
4.59.1.11	e_phnum	98
4.59.1.12	e_shentsize	98
4.59.1.13	e_shnum	98
4.59.1.14	e_shstrndx	98
4.60	elf64_note Struct Reference	99
4.60.1	Field Documentation	99
4.60.1.1	n_namesz	99
4.60.1.2	n_descsz	99
4.60.1.3	n_type	99
4.61	elf64_phdr Struct Reference	100
4.61.1	Field Documentation	100
4.61.1.1	p_type	100
4.61.1.2	p_flags	100
4.61.1.3	p_offset	100
4.61.1.4	p_vaddr	100
4.61.1.5	p_paddr	100
4.61.1.6	p_filesz	100
4.61.1.7	p_memsz	100
4.61.1.8	p_align	100
4.62	elf64_rel Struct Reference	101
4.62.1	Field Documentation	101
4.62.1.1	r_offset	101
4.62.1.2	r_info	101
4.63	elf64_rela Struct Reference	102
4.63.1	Field Documentation	102
4.63.1.1	r_offset	102
4.63.1.2	r_info	102
4.63.1.3	r_addend	102

4.64 elf64_shdr Struct Reference	103
4.64.1 Field Documentation	103
4.64.1.1 sh_name	103
4.64.1.2 sh_type	103
4.64.1.3 sh_flags	103
4.64.1.4 sh_addr	103
4.64.1.5 sh_offset	103
4.64.1.6 sh_size	103
4.64.1.7 sh_link	103
4.64.1.8 sh_info	103
4.64.1.9 sh_addralign	103
4.64.1.10 sh_entsize	103
4.65 elf64_sym Struct Reference	104
4.65.1 Field Documentation	104
4.65.1.1 st_name	104
4.65.1.2 st_info	104
4.65.1.3 st_other	104
4.65.1.4 st_shndx	104
4.65.1.5 st_value	104
4.65.1.6 st_size	104
4.66 elf_obj Struct Reference	105
4.66.1 Field Documentation	106
4.66.1.1 e_hdr	106
4.66.1.2 e_shdrs	106
4.66.1.3 e_sections	106
4.66.1.4 e_syms	106
4.66.1.5 e_sym_num	106
4.66.1.6 e_sym_str_tab	106
4.66.1.7 e_rels	106
4.66.1.8 e_rel_num	106
4.66.1.9 e_rel_sym	106
4.66.1.10 e_rel_sec	106
4.66.1.11 e_relas	106
4.66.1.12 e_rela_num	106
4.66.1.13 e_rela_sym	106
4.66.1.14 e_rela_sec	106

4.67 eth_device Struct Reference	107
4.67.1 Field Documentation	110
4.67.1.1 enabled	110
4.67.1.2 baseaddr	110
4.67.1.3 dma	110
4.67.1.4 tx_channel	110
4.67.1.5 rx_channel	110
4.67.1.6 mac_address	110
4.67.1.7 mac_int	110
4.67.1.8 base_vapi_id	110
4.67.1.9 rxfile	110
4.67.1.10 txfile	110
4.67.1.11 txfd	110
4.67.1.12 rxfd	110
4.67.1.13 loopback_offset	110
4.67.1.14 sockif	110
4.67.1.15 rtx_sock	110
4.67.1.16 rtx_type	110
4.67.1.17 ifr	110
4.67.1.18 rfd	110
4.67.1.19 wfd	110
4.67.1.20 state	110
4.67.1.21 bd_index	110
4.67.1.22 bd	110
4.67.1.23 bd_addr	110
4.67.1.24 working	110
4.67.1.25 waiting_for_dma	110
4.67.1.26 error	110
4.67.1.27 packet_length	110
4.67.1.28 minimum_length	110
4.67.1.29 maximum_length	110
4.67.1.30 add_crc	110
4.67.1.31 crc_dly	110
4.67.1.32 crc_value	110
4.67.1.33 bytes_left	110
4.67.1.34 bytes_sent	110

4.67.1.35 tx	110
4.67.1.36 fd	110
4.67.1.37 offset	110
4.67.1.38 bytes_read	110
4.67.1.39 rx	110
4.67.1.40 moder	110
4.67.1.41 int_source	110
4.67.1.42 int_mask	110
4.67.1.43 ipgt	110
4.67.1.44 ipgr1	110
4.67.1.45 ipgr2	110
4.67.1.46 packetlen	110
4.67.1.47 collconf	110
4.67.1.48 tx_bd_num	110
4.67.1.49 controlmoder	110
4.67.1.50 miimoder	110
4.67.1.51 miicommand	110
4.67.1.52 miiaddress	110
4.67.1.53 miitx_data	110
4.67.1.54 miirx_data	110
4.67.1.55 miistatus	110
4.67.1.56 hash0	110
4.67.1.57 hash1	110
4.67.1.58 bd_ram	110
4.67.1.59 regs	110
4.67.1.60 rx_buff	110
4.67.1.61 tx_buff	110
4.67.1.62 lo_buff	110
4.68 ether_addr Struct Reference	112
4.68.1 Field Documentation	112
4.68.1.1 ether_addr_octet	112
4.69 ether_header Struct Reference	113
4.69.1 Field Documentation	113
4.69.1.1 ether_dhost	113
4.69.1.2 ether_shost	113
4.69.1.3 ether_type	113

4.70	fb_state Struct Reference	114
4.70.1	Field Documentation	115
4.70.1.1	enabled	115
4.70.1.2	pal	115
4.70.1.3	ctrl	115
4.70.1.4	pic	115
4.70.1.5	in_refresh	115
4.70.1.6	refresh_count	115
4.70.1.7	addr	115
4.70.1.8	cam_addr	115
4.70.1.9	camerax	115
4.70.1.10	cameray	115
4.70.1.11	camera_pos	115
4.70.1.12	baseaddr	115
4.70.1.13	refresh	115
4.70.1.14	refresh_rate	115
4.70.1.15	filename	115
4.71	fd_channel Struct Reference	116
4.71.1	Detailed Description	116
4.71.2	Field Documentation	116
4.71.2.1	fdin	116
4.71.2.2	fdout	116
4.72	file_channel Struct Reference	117
4.72.1	Detailed Description	117
4.72.2	Field Documentation	117
4.72.2.1	fds	117
4.72.2.2	namein	117
4.72.2.3	nameout	117
4.73	fstats_entry Struct Reference	118
4.73.1	Field Documentation	118
4.73.1.1	insn1	118
4.73.1.2	insn2	118
4.73.1.3	cnt_dynamic	118
4.73.1.4	depend	118
4.74	func_struct Struct Reference	119
4.74.1	Detailed Description	119

4.74.2	Field Documentation	119
4.74.2.1	addr	119
4.74.2.2	name	119
4.74.2.3	cum_cycles	119
4.74.2.4	calls	119
4.75	gpio_device Struct Reference	120
4.75.1	Field Documentation	121
4.75.1.1	enabled	121
4.75.1.2	baseaddr	121
4.75.1.3	irq	121
4.75.1.4	gpio_number	121
4.75.1.5	base_vapi_id	121
4.75.1.6	auxiliary_inputs	121
4.75.1.7	in	121
4.75.1.8	out	121
4.75.1.9	oe	121
4.75.1.10	inte	121
4.75.1.11	ptrig	121
4.75.1.12	aux	121
4.75.1.13	ctrl	121
4.75.1.14	ints	121
4.75.1.15	external_clock	121
4.75.1.16	curr	121
4.75.1.17	next	121
4.76	hist_exec Struct Reference	122
4.76.1	Detailed Description	122
4.76.2	Field Documentation	122
4.76.2.1	addr	122
4.76.2.2	prev	122
4.76.2.3	next	122
4.77	ic Struct Reference	123
4.77.1	Field Documentation	124
4.77.1.1	mem	124
4.77.1.2	lrus	124
4.77.1.3	tags	124
4.77.1.4	enabled	124

4.77.1.5	nways	124
4.77.1.6	nsets	124
4.77.1.7	blocksize	124
4.77.1.8	ustates	124
4.77.1.9	missdelay	124
4.77.1.10	hitdelay	124
4.77.1.11	blocksize_log2	124
4.77.1.12	set_mask	124
4.77.1.13	tagaddr_mask	124
4.77.1.14	last_way	124
4.77.1.15	block_offset_mask	124
4.77.1.16	block_mask	124
4.77.1.17	ustates_reload	124
4.78	immu Struct Reference	125
4.78.1	Field Documentation	126
4.78.1.1	enabled	126
4.78.1.2	nways	126
4.78.1.3	nsets	126
4.78.1.4	pagesize	126
4.78.1.5	pagesize_log2	126
4.78.1.6	page_offset_mask	126
4.78.1.7	page_mask	126
4.78.1.8	vpn_mask	126
4.78.1.9	lru_reload	126
4.78.1.10	set_mask	126
4.78.1.11	entrysize	126
4.78.1.12	ustates	126
4.78.1.13	missdelay	126
4.78.1.14	hitdelay	126
4.79	immustats_entry Struct Reference	127
4.79.1	Field Documentation	127
4.79.1.1	fetch_tlbhit	127
4.79.1.2	fetch_tlbmiss	127
4.79.1.3	fetch_pagefaults	127
4.80	INFOHEADER Struct Reference	128
4.80.1	Field Documentation	128

4.80.1.1	size	128
4.80.1.2	width	128
4.80.1.3	height	128
4.80.1.4	planes	128
4.80.1.5	bits	128
4.80.1.6	compression	128
4.80.1.7	imagesize	128
4.80.1.8	xresolution	128
4.80.1.9	yresolution	128
4.80.1.10	ncolours	128
4.80.1.11	importantcolours	128
4.81	iqueue_entry Struct Reference	129
4.81.1	Detailed Description	129
4.81.2	Field Documentation	129
4.81.2.1	insn_index	129
4.81.2.2	insn	129
4.81.2.3	insn_addr	129
4.82	jtr_chain_message Struct Reference	130
4.82.1	Field Documentation	130
4.82.1.1	command	130
4.82.1.2	length	130
4.82.1.3	chain	130
4.83	jtr_chain_response Struct Reference	131
4.83.1	Field Documentation	131
4.83.1.1	status	131
4.84	jtr_failure_response Struct Reference	132
4.84.1	Field Documentation	132
4.84.1.1	status	132
4.85	jtr_read_block_message Struct Reference	133
4.85.1	Field Documentation	133
4.85.1.1	command	133
4.85.1.2	length	133
4.85.1.3	address	133
4.85.1.4	num_regs	133
4.86	jtr_read_block_response Struct Reference	134
4.86.1	Field Documentation	134

4.86.1.1	status	134
4.86.1.2	num_regs	134
4.86.1.3	data	134
4.87	jtr_read_message Struct Reference	135
4.87.1	Field Documentation	135
4.87.1.1	command	135
4.87.1.2	length	135
4.87.1.3	address	135
4.88	jtr_read_response Struct Reference	136
4.88.1	Field Documentation	136
4.88.1.1	status	136
4.88.1.2	data_h	136
4.88.1.3	data_l	136
4.89	jtr_write_block_message Struct Reference	137
4.89.1	Field Documentation	137
4.89.1.1	command	137
4.89.1.2	length	137
4.89.1.3	address	137
4.89.1.4	num_regs	137
4.89.1.5	data	137
4.90	jtr_write_block_response Struct Reference	138
4.90.1	Field Documentation	138
4.90.1.1	status	138
4.91	jtr_write_message Struct Reference	139
4.91.1	Field Documentation	139
4.91.1.1	command	139
4.91.1.2	length	139
4.91.1.3	address	139
4.91.1.4	data_h	139
4.91.1.5	data_l	139
4.92	jtr_write_response Struct Reference	140
4.92.1	Field Documentation	140
4.92.1.1	status	140
4.93	kbd_state Struct Reference	141
4.93.1	Field Documentation	141
4.93.1.1	buf	141

4.93.1.2	buf_count	141
4.93.1.3	buf_head	141
4.93.1.4	buf_tail	141
4.93.1.5	rxfs	141
4.93.1.6	ccmd	141
4.93.1.7	kcmd	141
4.93.1.8	ccmdbyte	141
4.93.1.9	kresp	141
4.93.1.10	slowdown	141
4.93.1.11	enabled	141
4.93.1.12	irq	141
4.93.1.13	baseaddr	141
4.93.1.14	rxfile	141
4.94	label_entry Struct Reference	142
4.94.1	Detailed Description	142
4.94.2	Field Documentation	142
4.94.2.1	name	142
4.94.2.2	addr	142
4.94.2.3	next	142
4.95	mc Struct Reference	143
4.95.1	Field Documentation	144
4.95.1.1	csr	144
4.95.1.2	poc	144
4.95.1.3	ba_mask	144
4.95.1.4	csc	144
4.95.1.5	tms	144
4.95.1.6	baseaddr	144
4.95.1.7	enabled	144
4.95.1.8	index	144
4.95.1.9	mc_areas	144
4.95.1.10	next	144
4.96	mc_area Struct Reference	145
4.96.1	Field Documentation	145
4.96.1.1	mem	145
4.96.1.2	cs	145
4.96.1.3	mc	145

4.96.1.4	next	145
4.97	mem_config Struct Reference	146
4.97.1	Member Enumeration Documentation	146
4.97.1.1	"@52	146
4.97.2	Field Documentation	147
4.97.2.1	ce	147
4.97.2.2	mc	147
4.97.2.3	baseaddr	147
4.97.2.4	size	147
4.97.2.5	name	147
4.97.2.6	log	147
4.97.2.7	delayr	147
4.97.2.8	delayw	147
4.97.2.9	mem	147
4.97.2.10	pattern	147
4.97.2.11	random_seed	147
4.97.2.12	type	147
4.98	mem_ops Struct Reference	148
4.98.1	Detailed Description	148
4.98.2	Field Documentation	149
4.98.2.1	readfunc32	149
4.98.2.2	readfunc16	149
4.98.2.3	readfunc8	149
4.98.2.4	read_dat8	149
4.98.2.5	read_dat16	149
4.98.2.6	read_dat32	149
4.98.2.7	writefunc32	149
4.98.2.8	writefunc16	149
4.98.2.9	writefunc8	149
4.98.2.10	write_dat8	149
4.98.2.11	write_dat16	149
4.98.2.12	write_dat32	149
4.98.2.13	writeprog32	149
4.98.2.14	writeprog8	149
4.98.2.15	writeprog32_dat	149
4.98.2.16	writeprog8_dat	149

4.98.2.17 delayr	149
4.98.2.18 delayw	149
4.98.2.19 log	149
4.99 memory_hash Struct Reference	150
4.99.1 Detailed Description	150
4.99.2 Field Documentation	150
4.99.2.1 next	150
4.99.2.2 addr	150
4.99.2.3 cnt	150
4.100 mprofentry_struct Struct Reference	151
4.100.1 Field Documentation	151
4.100.1.1 addr	151
4.100.1.2 type	151
4.101 mstats_entry Struct Reference	152
4.101.1 Field Documentation	152
4.101.1.1 byteadd	152
4.101.1.2 bf	152
4.101.1.3 bnf	152
4.101.1.4 bpb	152
4.101.1.5 btic	152
4.102 op_queue Struct Reference	153
4.102.1 Field Documentation	154
4.102.1.1 num_ops	154
4.102.1.2 ops_len	154
4.102.1.3 ops	154
4.102.1.4 num_ops_param	154
4.102.1.5 ops_param_len	154
4.102.1.6 ops_param	154
4.102.1.7 jump_local	154
4.102.1.8 jump_local_loc	154
4.102.1.9 not_jump_loc	154
4.102.1.10 xref	154
4.102.1.11 linsn_addr	154
4.102.1.12 reg_t	154
4.102.1.13 flags	154
4.102.1.14 linsn_index	154

4.102.1.15	param_type	154
4.102.1.16	param	154
4.102.1.17	param_num	154
4.102.1.18	nsn	154
4.102.1.19	prev	154
4.102.1.20	next	154
4.103	param_val Union Reference	155
4.103.1	Detailed Description	155
4.103.2	Field Documentation	155
4.103.2.1	str_val	155
4.103.2.2	int_val	155
4.103.2.3	longlong_val	155
4.103.2.4	addr_val	155
4.104	raw_stats Struct Reference	156
4.104.1	Field Documentation	156
4.104.1.1	reg	156
4.104.1.2	range	156
4.105	reloc Struct Reference	157
4.105.1	Field Documentation	157
4.105.1.1	func_offset	157
4.105.1.2	addend	157
4.105.1.3	type	157
4.105.1.4	name	157
4.106	runtime Struct Reference	158
4.106.1	Detailed Description	158
4.106.2	Field Documentation	160
4.106.2.1	fprof	160
4.106.2.2	fmprof	160
4.106.2.3	fexe_log	160
4.106.2.4	fout	160
4.106.2.5	init	160
4.106.2.6	filename	160
4.106.2.7	iprompt	160
4.106.2.8	iprompt_run	160
4.106.2.9	cycles	160
4.106.2.10	end_cycles	160

4.106.2.1	ltime_point	160
4.106.2.2	ext_int	160
4.106.2.3	mem_cycles	160
4.106.2.4	loadcycles	160
4.106.2.5	storecycles	160
4.106.2.6	reset_cycles	160
4.106.2.7	hush	160
4.106.2.8	sim	160
4.106.2.9	instructions	160
4.106.2.20	reset_instructions	160
4.106.2.21	stalled	160
4.106.2.22	hazardwait	160
4.106.2.23	supercycles	160
4.106.2.24	cpu	160
4.106.2.25	enabled	160
4.106.2.26	vapi_file	160
4.106.2.27	server_port	160
4.106.2.28	vapi	160
4.106.2.29	ndelay	160
4.106.2.30	cycle_duration	160
4.106.2.3	lcuc	160
4.107	sched_entry Struct Reference	162
4.107.1	Detailed Description	162
4.107.2	Field Documentation	162
4.107.2.1	time	162
4.107.2.2	param	162
4.107.2.3	func	162
4.107.2.4	next	162
4.108	scheduler_struct Struct Reference	163
4.108.1	Detailed Description	163
4.108.2	Field Documentation	163
4.108.2.1	free_job_queue	163
4.108.2.2	job_queue	163
4.109	sim_command Struct Reference	164
4.109.1	Field Documentation	164
4.109.1.1	name	164

4.109.1.2 cmd_handle	164
4.110sim_reset_hook Struct Reference	165
4.110.1 Detailed Description	165
4.110.2 Field Documentation	165
4.110.2.1 dat	165
4.110.2.2 reset_hook	165
4.110.2.3 next	165
4.111sim_stat Struct Reference	166
4.111.1 Field Documentation	166
4.111.1.1 stat_func	166
4.111.1.2 dat	166
4.111.1.3 next	166
4.112spr_bit_def Struct Reference	167
4.112.1 Field Documentation	167
4.112.1.1 name	167
4.112.1.2 mask	167
4.113spr_def Struct Reference	168
4.113.1 Field Documentation	168
4.113.1.1 from_spr	168
4.113.1.2 to_spr	168
4.113.1.3 name	168
4.113.1.4 bits	168
4.114sstats_entry Struct Reference	169
4.114.1 Field Documentation	169
4.114.1.1 insn	169
4.114.1.2 cnt_dynamic	169
4.115stack_struct Struct Reference	170
4.115.1 Detailed Description	170
4.115.2 Field Documentation	170
4.115.2.1 addr	170
4.115.2.2 cycles	170
4.115.2.3 raddr	170
4.115.2.4 name	170
4.116tcp_channel Struct Reference	171
4.116.1 Detailed Description	171
4.116.2 Field Documentation	171

4.116.2.1	fds	171
4.116.2.2	socket_fd	171
4.116.2.3	port_number	171
4.116.2.4	connected	171
4.116.2.5	nonblocking	171
4.117	tty_channel Struct Reference	172
4.117.1	Detailed Description	172
4.117.2	Field Documentation	172
4.117.2.1	fds	172
4.118	vapi_handler Struct Reference	173
4.118.1	Field Documentation	173
4.118.1.1	fd	173
4.118.1.2	base_id	173
4.118.1.3	num_ids	173
4.118.1.4	read_func	173
4.118.1.5	priv_dat	173
4.118.1.6	next	173
4.118.1.7	temp	173
4.119	vga_state Struct Reference	174
4.119.1	Field Documentation	175
4.119.1.1	enabled	175
4.119.1.2	pics	175
4.119.1.3	ctrl	175
4.119.1.4	stat	175
4.119.1.5	htim	175
4.119.1.6	vtim	175
4.119.1.7	vbindx	175
4.119.1.8	vbar	175
4.119.1.9	hlen	175
4.119.1.10	vlen	175
4.119.1.11	lpindex	175
4.119.1.12	palette	175
4.119.1.13	baseaddr	175
4.119.1.14	refresh_rate	175
4.119.1.15	rq	175
4.119.1.16	filename	175

4.120xterm_channel Struct Reference	176
4.120.1 Detailed Description	176
4.120.2 Field Documentation	176
4.120.2.1 fds	176
4.120.2.2 pid	176
4.120.2.3 argv	176
5 File Documentation	177
5.1 bpb/branch-predict.c File Reference	177
5.1.1 Define Documentation	180
5.1.1.1 BPB_LEN	180
5.1.1.2 BPB_PSTATES	180
5.1.1.3 BPB_USTATES	180
5.1.1.4 BPB_WAYS	180
5.1.1.5 BTIC_BLOCKSIZE	180
5.1.1.6 BTIC_LEN	180
5.1.1.7 BTIC_USTATES	180
5.1.1.8 BTIC_WAYS	180
5.1.2 Function Documentation	180
5.1.2.1 bpb_btic	180
5.1.2.2 bpb_enabled	180
5.1.2.3 bpb_hitdelay	180
5.1.2.4 bpb_info	180
5.1.2.5 bpb_missdelay	180
5.1.2.6 bpb_sbp_bf_fwd	180
5.1.2.7 bpb_sbp_bnf_fwd	180
5.1.2.8 bpb_update	180
5.1.2.9 btic_info	180
5.1.2.10 btic_update	180
5.1.2.11 reg_bpb_sec	180
5.1.3 Variable Documentation	181
5.1.3.1 bpb	181
5.1.3.2 btic	181
5.2 bpb/branch-predict.h File Reference	182
5.2.1 Function Documentation	183
5.2.1.1 bpb_info	183
5.2.1.2 bpb_update	183

5.2.1.3	btic_info	183
5.2.1.4	btic_update	183
5.2.1.5	reg_bpb_sec	183
5.3	cache/dcache-model.c File Reference	184
5.3.1	Function Documentation	185
5.3.1.1	dc_blocksize	185
5.3.1.2	dc_enabled	185
5.3.1.3	dc_info	185
5.3.1.4	dc_inv	185
5.3.1.5	dc_load_hitdelay	185
5.3.1.6	dc_load_missdelay	185
5.3.1.7	dc_nsets	185
5.3.1.8	dc_nways	186
5.3.1.9	dc_simulate_read	186
5.3.1.10	dc_simulate_write	187
5.3.1.11	dc_store_hitdelay	187
5.3.1.12	dc_store_missdelay	187
5.3.1.13	dc_ustates	187
5.3.1.14	reg_dc_sec	188
5.3.2	Variable Documentation	188
5.3.2.1	dc	188
5.4	cache/dcache-model.h File Reference	189
5.4.1	Define Documentation	190
5.4.1.1	MAX_DC_BLOCK_SIZE	190
5.4.1.2	MAX_DC_SETS	190
5.4.1.3	MAX_DC_WAYS	190
5.4.1.4	MIN_DC_BLOCK_SIZE	190
5.4.2	Function Documentation	190
5.4.2.1	dc_info	190
5.4.2.2	dc_inv	190
5.4.2.3	dc_simulate_read	190
5.4.2.4	dc_simulate_write	191
5.4.2.5	reg_dc_sec	191
5.5	cache/icache-model.c File Reference	192
5.5.1	Define Documentation	193
5.5.1.1	MAX_IC_BLOCK_SIZE	193

5.5.1.2	MAX_IC_SETS	193
5.5.1.3	MAX_IC_WAYS	193
5.5.1.4	MIN_IC_BLOCK_SIZE	193
5.5.2	Function Documentation	193
5.5.2.1	ic_blocksize	193
5.5.2.2	ic_enabled	193
5.5.2.3	ic_end_sec	194
5.5.2.4	ic_hitdelay	194
5.5.2.5	ic_info	194
5.5.2.6	ic_inv	194
5.5.2.7	ic_missdelay	194
5.5.2.8	ic_nsets	194
5.5.2.9	ic_nways	194
5.5.2.10	ic_simulate_fetch	195
5.5.2.11	ic_start_sec	195
5.5.2.12	ic_ustates	195
5.5.2.13	reg_ic_sec	196
5.5.3	Variable Documentation	196
5.5.3.1	ic_state	196
5.6	cache/icache-model.h File Reference	197
5.6.1	Function Documentation	198
5.6.1.1	ic_inv	198
5.6.1.2	ic_simulate_fetch	198
5.6.1.3	reg_ic_sec	198
5.6.2	Variable Documentation	199
5.6.2.1	ic_state	199
5.7	cpu-config.c File Reference	200
5.7.1	Define Documentation	201
5.7.1.1	WARNING	201
5.7.2	Function Documentation	201
5.7.2.1	cpu_cfg	201
5.7.2.2	cpu_cfgr	201
5.7.2.3	cpu_dependstats	201
5.7.2.4	cpu_hazards	201
5.7.2.5	cpu_rev	201
5.7.2.6	cpu_sbuf_len	201

5.7.2.7	cpu_sr	201
5.7.2.8	cpu_superscalar	202
5.7.2.9	cpu_upr	202
5.7.2.10	cpu_ver	202
5.7.2.11	reg_cpu_sec	202
5.8	cpu-config.h File Reference	204
5.8.1	Function Documentation	204
5.8.1.1	reg_cpu_sec	204
5.9	cpu/common/abstract.c File Reference	205
5.9.1	Function Documentation	207
5.9.1.1	adjust_rw_delay	207
5.9.1.2	bit_mask	207
5.9.1.3	disassemble_memory	207
5.9.1.4	done_memory_table	207
5.9.1.5	dump_memory	207
5.9.1.6	eval_direct16	208
5.9.1.7	eval_direct32	208
5.9.1.8	eval_direct8	209
5.9.1.9	eval_insn	209
5.9.1.10	eval_mem16	209
5.9.1.11	eval_mem32	210
5.9.1.12	eval_mem8	210
5.9.1.13	eval_mem_16_inv	211
5.9.1.14	eval_mem_16_inv_direct	211
5.9.1.15	eval_mem_32_inv	211
5.9.1.16	eval_mem_32_inv_direct	212
5.9.1.17	eval_mem_8_inv	212
5.9.1.18	eval_mem_8_inv_direct	212
5.9.1.19	evalsim_mem16	212
5.9.1.20	evalsim_mem32	213
5.9.1.21	evalsim_mem8	213
5.9.1.22	generate_time_pretty	214
5.9.1.23	memory_table_status	214
5.9.1.24	reg_mem_area	214
5.9.1.25	register_memoryarea_mask	214
5.9.1.26	set_direct16	215

5.9.1.27	set_direct32	215
5.9.1.28	set_direct8	216
5.9.1.29	set_mem16	216
5.9.1.30	set_mem32	217
5.9.1.31	set_mem8	217
5.9.1.32	set_mem_16_inv	218
5.9.1.33	set_mem_16_inv_direct	218
5.9.1.34	set_mem_32_inv	218
5.9.1.35	set_mem_32_inv_direct	219
5.9.1.36	set_mem_8_inv	219
5.9.1.37	set_mem_8_inv_direct	219
5.9.1.38	set_mem_valid	219
5.9.1.39	set_program32	219
5.9.1.40	set_program8	219
5.9.1.41	setsim_mem16	220
5.9.1.42	setsim_mem32	220
5.9.1.43	setsim_mem8	221
5.9.1.44	verify_memoryarea	221
5.9.2	Variable Documentation	221
5.9.2.1	cur_area	221
5.9.2.2	cur_vadd	221
5.9.2.3	data_ci	221
5.9.2.4	dev_list	221
5.9.2.5	insn_ci	221
5.9.2.6	mc_area	221
5.10	cpu/common/abstract.h File Reference	222
5.10.1	Define Documentation	224
5.10.1.1	CT_NONE	224
5.10.1.2	CT_PHYSICAL	224
5.10.1.3	CT_VIRTUAL	224
5.10.1.4	DEFAULT_MEMORY_LEN	224
5.10.1.5	DEFAULT_MEMORY_START	224
5.10.1.6	HISTEXEC_LEN	224
5.10.1.7	INSNAME_LEN	224
5.10.1.8	LABELNAME_LEN	224
5.10.1.9	LE16	224

5.10.1.10	LONGEST	224
5.10.1.11	MAX_OPERANDS	224
5.10.1.12	OP_MEM_ACCESS	224
5.10.1.13	OPERANDNAME_LEN	224
5.10.1.14	STACK_SIZE	224
5.10.1.15	ULONGEST	224
5.10.2	Function Documentation	224
5.10.2.1	adjust_rw_delay	224
5.10.2.2	disassemble_memory	224
5.10.2.3	done_memory_table	225
5.10.2.4	dump_memory	225
5.10.2.5	eval_direct16	226
5.10.2.6	eval_direct32	226
5.10.2.7	eval_direct8	227
5.10.2.8	eval_insn	227
5.10.2.9	eval_mem16	227
5.10.2.10	eval_mem32	228
5.10.2.11	eval_mem8	228
5.10.2.12	evalsim_mem16	229
5.10.2.13	evalsim_mem32	229
5.10.2.14	evalsim_mem8	230
5.10.2.15	generate_time_pretty	230
5.10.2.16	memory_table_status	230
5.10.2.17	reg_mem_area	231
5.10.2.18	set_direct16	231
5.10.2.19	set_direct32	232
5.10.2.20	set_direct8	232
5.10.2.21	set_mem16	233
5.10.2.22	set_mem32	233
5.10.2.23	set_mem8	234
5.10.2.24	set_mem_valid	234
5.10.2.25	set_program32	234
5.10.2.26	set_program8	234
5.10.2.27	setsim_mem16	235
5.10.2.28	setsim_mem32	235
5.10.2.29	setsim_mem8	236

5.10.2.30	verify_memoryarea	236
5.10.3	Variable Documentation	236
5.10.3.1	cur_area	236
5.10.3.2	data_ci	236
5.10.3.3	hist_exec_tail	236
5.10.3.4	insn_ci	236
5.11	cpu/common/coff.h File Reference	237
5.11.1	Define Documentation	242
5.11.1.1	C_ALIAS	242
5.11.1.2	C_ARG	242
5.11.1.3	C_AUTO	242
5.11.1.4	C_AUTOARG	242
5.11.1.5	C_BCOMM	242
5.11.1.6	C_BINCL	242
5.11.1.7	C_BLOCK	242
5.11.1.8	C_BSTAT	242
5.11.1.9	C_DECL	242
5.11.1.10	C_DEFINE	242
5.11.1.11	C_ECOML	242
5.11.1.12	C_ECOMM	242
5.11.1.13	C_EFCN	242
5.11.1.14	C_EINCL	242
5.11.1.15	C_ENTAG	242
5.11.1.16	C_ENTRY	242
5.11.1.17	C_EOS	242
5.11.1.18	C_ESTAT	242
5.11.1.19	C_EXT	242
5.11.1.20	C_EXTDEF	242
5.11.1.21	C_EXTLAB	242
5.11.1.22	C_FCEN	242
5.11.1.23	C_FIELD	242
5.11.1.24	C_FILE	242
5.11.1.25	C_FUN	242
5.11.1.26	C_GSYM	242
5.11.1.27	C_HIDDEN	242
5.11.1.28	C_HIDEXT	242

5.11.1.29 C_LABEL	242
5.11.1.30 C_LASTENT	242
5.11.1.31 C_LEAFEXT	242
5.11.1.32 C_LEAFPROC	242
5.11.1.33 C_LEAFSTAT	242
5.11.1.34 C_LINE	242
5.11.1.35 C_LSYM	242
5.11.1.36 C_MOE	242
5.11.1.37 C_MOS	242
5.11.1.38 C_MOU	242
5.11.1.39 C_NT_WEAK	242
5.11.1.40 C_NULL	242
5.11.1.41 C_OPTVAR	242
5.11.1.42 C_PRAGMA	242
5.11.1.43 C_PSYM	242
5.11.1.44 C_REG	242
5.11.1.45 C_REGPARM	242
5.11.1.46 C_RPSYM	242
5.11.1.47 C_RSYM	242
5.11.1.48 C_SCALL	242
5.11.1.49 C_SECTION	242
5.11.1.50 C_SEGMENT	242
5.11.1.51 C_SHADOW	242
5.11.1.52 C_STAT	242
5.11.1.53 C_STATLAB	242
5.11.1.54 C_STRTAG	242
5.11.1.55 C_STSYM	242
5.11.1.56 C_SYSTEM	242
5.11.1.57 C_TCSYM	242
5.11.1.58 C_THUMBEXT	242
5.11.1.59 C_THUMBEXTFUNC	242
5.11.1.60 C_THUMBLABEL	242
5.11.1.61 C_THUMBSTAT	242
5.11.1.62 C_THUMBSTATFUNC	242
5.11.1.63 C_TPDEF	242
5.11.1.64 C_UEXT	242

5.11.1.65 C_ULABEL	242
5.11.1.66 C_UNTAG	242
5.11.1.67 C_USTATIC	242
5.11.1.68 C_VERSION	242
5.11.1.69 C_WEAKEXT	242
5.11.1.70 COFF_AOUTSZ	242
5.11.1.71 COFF_AUXENT	242
5.11.1.72 COFF_AUXESZ	242
5.11.1.73 COFF_BSS	242
5.11.1.74 COFF_COMMENT	242
5.11.1.75 COFF_DATA	242
5.11.1.76 COFF_DEF_BSS_SECTION_ALIGNMENT	242
5.11.1.77 COFF_DEF_DATA_SECTION_ALIGNMENT	242
5.11.1.78 COFF_DEF_SECTION_ALIGNMENT	242
5.11.1.79 COFF_DEF_TEXT_SECTION_ALIGNMENT	242
5.11.1.80 COFF_DMAGIC	242
5.11.1.81 COFF_E_DIMNUM	242
5.11.1.82 COFF_E_FILNMLEN	242
5.11.1.83 COFF_E_SYMNMLEN	242
5.11.1.84 COFF_ETEXT	242
5.11.1.85 COFF_F_AR16WR	242
5.11.1.86 COFF_F_AR32W	242
5.11.1.87 COFF_F_AR32WR	242
5.11.1.88 COFF_F_EXEC	242
5.11.1.89 COFF_F_LNNO	242
5.11.1.90 COFF_F_LSYMS	242
5.11.1.91 COFF_F_MINMAL	242
5.11.1.92 COFF_F_NODF	242
5.11.1.93 COFF_F_PATCH	242
5.11.1.94 COFF_F_RELFLG	242
5.11.1.95 COFF_F_SWABD	242
5.11.1.96 COFF_F_UPDATE	242
5.11.1.97 COFF_FILHDR	242
5.11.1.98 COFF_FILHSZ	242
5.11.1.99 COFF_I386BADMAG	242
5.11.1.100COFF_I386MAGIC	242

5.11.1.101	COFF_JMAGIC	242
5.11.1.102	COFF_LIB	242
5.11.1.103	COFF_LINENO	242
5.11.1.104	COFF_LINESZ	242
5.11.1.105	COFF_LONG	242
5.11.1.106	COFF_LONG_H	242
5.11.1.107	COFF_LONG_L	242
5.11.1.108	COFF_N_BTMASK	242
5.11.1.109	COFF_N_BTSHIFT	242
5.11.1.110	COFF_N_TMASK	242
5.11.1.111	COFF_N_TSHIFT	242
5.11.1.112	COFF_OMAGIC	242
5.11.1.113	COFF_RELOC	242
5.11.1.114	COFF_RELSZ	242
5.11.1.115	COFF_SCNHDR	242
5.11.1.116	COFF_SCNHSZ	242
5.11.1.117	COFF_SECT_BSS	242
5.11.1.118	COFF_SECT_DATA	242
5.11.1.119	COFF_SECT_REQD	242
5.11.1.120	COFF_SECT_TEXT	242
5.11.1.121	COFF_SHMAGIC	242
5.11.1.122	COFF_SHORT	242
5.11.1.123	COFF_SHORT_H	242
5.11.1.124	COFF_SHORT_L	242
5.11.1.125	COFF_SLIBHD	242
5.11.1.126	COFF_SLIBSZ	242
5.11.1.127	COFF_STMAGIC	242
5.11.1.128	COFF_STYP_BSS	242
5.11.1.129	COFF_STYP_COPY	242
5.11.1.130	COFF_STYP_DATA	242
5.11.1.131	COFF_STYP_DSECT	242
5.11.1.132	COFF_STYP_GROUP	242
5.11.1.133	COFF_STYP_INFO	242
5.11.1.134	COFF_STYP_LIB	242
5.11.1.135	COFF_STYP_NOLOAD	242
5.11.1.136	COFF_STYP_OVER	242

5.11.1.137	COFF_STYP_PAD	242
5.11.1.138	COFF_STYP_REG	242
5.11.1.139	COFF_STYP_TEXT	242
5.11.1.140	COFF_SYMENT	242
5.11.1.141	COFF_SYMESZ	242
5.11.1.142	COFF_TEXT	242
5.11.1.143	COFF_ZMAGIC	242
5.11.1.144	E_DIMNUM	242
5.11.1.145	E_FILNMLEN	242
5.11.1.146	E_SYMNMLEN	242
5.11.1.147	KEEP_ENDIAN_LONG	242
5.11.1.148	KEEP_ENDIAN_SHORT	243
5.11.1.149	SWAP_ENDIAN_LONG	243
5.11.1.150	SWAP_ENDIAN_SHORT	243
5.12	cpu/common/elf.h File Reference	244
5.12.1	Define Documentation	250
5.12.1.1	AT_BASE	250
5.12.1.2	AT_EGID	250
5.12.1.3	AT_ENTRY	250
5.12.1.4	AT_EUID	250
5.12.1.5	AT_EXECFD	250
5.12.1.6	AT_FLAGS	250
5.12.1.7	AT_GID	250
5.12.1.8	AT_IGNORE	250
5.12.1.9	AT_NOTELF	250
5.12.1.10	AT_NULL	250
5.12.1.11	AT_PAGESZ	250
5.12.1.12	AT_PHDR	250
5.12.1.13	AT_PHERENT	250
5.12.1.14	AT_PHNUM	250
5.12.1.15	AT_UID	250
5.12.1.16	DT_DEBUG	250
5.12.1.17	DT_FINI	250
5.12.1.18	DT_HASH	250
5.12.1.19	DT_HIPROC	250
5.12.1.20	DT_INIT	250

5.12.1.21 DT_JMPREL	250
5.12.1.22 DT_LOPROC	250
5.12.1.23 DT_NEEDED	250
5.12.1.24 DT_NULL	250
5.12.1.25 DT_PLTGOT	250
5.12.1.26 DT_PLTREL	250
5.12.1.27 DT_PLTRELSZ	250
5.12.1.28 DT_REL	250
5.12.1.29 DT_RELA	250
5.12.1.30 DT_RELAENT	250
5.12.1.31 DT_RELASZ	250
5.12.1.32 DT_RELENT	250
5.12.1.33 DT_RELSZ	250
5.12.1.34 DT_RPATH	250
5.12.1.35 DT_SONAME	250
5.12.1.36 DT_STRSZ	250
5.12.1.37 DT_STRTAB	250
5.12.1.38 DT_SYMBOLIC	250
5.12.1.39 DT_SYMENT	250
5.12.1.40 DT_SYMTAB	250
5.12.1.41 DT_TEXTREL	250
5.12.1.42 EI_CLASS	250
5.12.1.43 EI_DATA	250
5.12.1.44 EI_MAG0	250
5.12.1.45 EI_MAG1	250
5.12.1.46 EI_MAG2	250
5.12.1.47 EI_MAG3	250
5.12.1.48 EI_NIDENT	250
5.12.1.49 EI_PAD	250
5.12.1.50 EI_VERSION	250
5.12.1.51 ELF32_R_SYM	250
5.12.1.52 ELF32_R_TYPE	250
5.12.1.53 ELF32_ST_BIND	250
5.12.1.54 ELF32_ST_TYPE	250
5.12.1.55 ELF_LONG_H	250
5.12.1.56 elf_note	251

5.12.1.57 elf_phdr	251
5.12.1.58 ELF_SHORT_H	251
5.12.1.59 ELFCLASS32	253
5.12.1.60 ELFCLASS64	253
5.12.1.61 ELFCLASSNONE	253
5.12.1.62 ELFCLASSNUM	253
5.12.1.63 ELFDATA2LSB	253
5.12.1.64 ELFDATA2MSB	253
5.12.1.65 ELFDATANONE	253
5.12.1.66 elfhdr	253
5.12.1.67 ELFMAG	253
5.12.1.68 ELFMAG0	253
5.12.1.69 ELFMAG1	253
5.12.1.70 ELFMAG2	253
5.12.1.71 ELFMAG3	253
5.12.1.72 EM_386	253
5.12.1.73 EM_486	253
5.12.1.74 EM_68K	253
5.12.1.75 EM_860	253
5.12.1.76 EM_88K	253
5.12.1.77 EM_ALPHA	253
5.12.1.78 EM_M32	253
5.12.1.79 EM_MIPS	253
5.12.1.80 EM_MIPS_RS4_BE	253
5.12.1.81 EM_NONE	253
5.12.1.82 EM_PARISC	253
5.12.1.83 EM_PPC	253
5.12.1.84 EM_SPARC	253
5.12.1.85 EM_SPARC32PLUS	253
5.12.1.86 EM_SPARC64	253
5.12.1.87 ET_CORE	253
5.12.1.88 ET_DYN	253
5.12.1.89 ET_EXEC	253
5.12.1.90 ET_HIPROC	253
5.12.1.91 ET_LOPROC	253
5.12.1.92 ET_NONE	253

5.12.1.93 ET_REL	253
5.12.1.94 EV_CURRENT	253
5.12.1.95 EV_NONE	253
5.12.1.96 EV_NUM	253
5.12.1.97 NT_PRFPREG	253
5.12.1.98 NT_PRPSINFO	253
5.12.1.99 NT_PRSTATUS	253
5.12.1.100NT_TASKSTRUCT	253
5.12.1.101PF_R	253
5.12.1.102PF_W	253
5.12.1.103PF_X	253
5.12.1.104PT_DYNAMIC	253
5.12.1.105PT_HIPROC	253
5.12.1.106PT_INTERP	253
5.12.1.107PT_LOAD	253
5.12.1.108PT_LOPROC	253
5.12.1.109PT_NOTE	253
5.12.1.110PT_NULL	253
5.12.1.111PT_PHDR	253
5.12.1.112PT_SHLIB	253
5.12.1.113R_386_32	253
5.12.1.114R_386_COPY	253
5.12.1.115R_386_GLOB_DAT	253
5.12.1.116R_386_GOT32	253
5.12.1.117R_386_GOTOFF	253
5.12.1.118R_386_GOTPC	253
5.12.1.119R_386_JMP_SLOT	253
5.12.1.120R_386_NONE	253
5.12.1.121R_386_NUM	253
5.12.1.122R_386_PC32	253
5.12.1.123R_386_PLT32	253
5.12.1.124R_386_RELATIVE	253
5.12.1.125R_68K_16	253
5.12.1.126R_68K_32	253
5.12.1.127R_68K_8	253
5.12.1.128R_68K_COPY	253

5.12.1.129R_68K_GLOB_DAT	253
5.12.1.130R_68K_GOT16	253
5.12.1.131R_68K_GOT16O	253
5.12.1.132R_68K_GOT32	253
5.12.1.133R_68K_GOT32O	253
5.12.1.134R_68K_GOT8	253
5.12.1.135R_68K_GOT8O	253
5.12.1.136R_68K_JMP_SLOT	253
5.12.1.137R_68K_NONE	253
5.12.1.138R_68K_PC16	253
5.12.1.139R_68K_PC32	253
5.12.1.140R_68K_PC8	253
5.12.1.141R_68K_PLT16	253
5.12.1.142R_68K_PLT16O	253
5.12.1.143R_68K_PLT32	253
5.12.1.144R_68K_PLT32O	253
5.12.1.145R_68K_PLT8	253
5.12.1.146R_68K_PLT8O	253
5.12.1.147R_68K_RELATIVE	253
5.12.1.148SELMAG	253
5.12.1.149SHF_ALLOC	253
5.12.1.150SHF_EXECINSTR	253
5.12.1.151SHF_MASKPROC	253
5.12.1.152SHF_WRITE	253
5.12.1.153SHN_ABS	253
5.12.1.154SHN_COMMON	253
5.12.1.155SHN_HIPROC	253
5.12.1.156SHN_HIRESERVE	253
5.12.1.157SHN_LOPROC	253
5.12.1.158SHN_LORESERVE	253
5.12.1.159SHN_UNDEF	253
5.12.1.160SHT_DYNAMIC	253
5.12.1.161SHT_DYNSYM	253
5.12.1.162SHT_HASH	253
5.12.1.163SHT_HIPROC	253
5.12.1.164SHT_HIUSER	253

5.12.1.165	SHT_LOPROC	253
5.12.1.166	SHT_LOUSER	253
5.12.1.167	SHT_NOBITS	253
5.12.1.168	SHT_NOTE	253
5.12.1.169	SHT_NULL	253
5.12.1.170	SHT_NUM	253
5.12.1.171	SHT_PROGBITS	253
5.12.1.172	SHT_REL	253
5.12.1.173	SHT_RELA	253
5.12.1.174	SHT_SHLIB	253
5.12.1.175	SHT_STRTAB	253
5.12.1.176	SHT_SYMTAB	253
5.12.1.177	STB_GLOBAL	253
5.12.1.178	STB_LOCAL	253
5.12.1.179	STB_WEAK	253
5.12.1.180	STT_FILE	253
5.12.1.181	STT_FUNC	253
5.12.1.182	STT_NOTYPE	253
5.12.1.183	STT_OBJECT	253
5.12.1.184	STT_SECTION	253
5.12.2	Typedef Documentation	253
5.12.2.1	Elf32_Addr	253
5.12.2.2	Elf32_Dyn	253
5.12.2.3	Elf32_Ehdr	253
5.12.2.4	Elf32_Half	253
5.12.2.5	Elf32_Nhdr	253
5.12.2.6	Elf32_Off	253
5.12.2.7	Elf32_Phdr	253
5.12.2.8	Elf32_Rel	253
5.12.2.9	Elf32_Rela	253
5.12.2.10	Elf32_Shdr	253
5.12.2.11	Elf32_Sword	253
5.12.2.12	Elf32_Sym	253
5.12.2.13	Elf32_Word	253
5.12.2.14	Elf64_Ehdr	253
5.12.2.15	Elf64_Nhdr	253

5.12.2.16	Elf64_Phdr	253
5.12.2.17	Elf64_Rel	253
5.12.2.18	Elf64_Rela	253
5.12.2.19	Elf64_Shdr	253
5.12.2.20	Elf64_Sym	253
5.12.3	Variable Documentation	253
5.12.3.1	_DYNAMIC	253
5.13	cpu/common/execute.h File Reference	254
5.13.1	Define Documentation	255
5.13.1.1	CURINSN	255
5.13.2	Function Documentation	255
5.13.2.1	analysis	255
5.13.2.2	cpu_clock	255
5.13.2.3	cpu_reset	256
5.13.2.4	depend_operands	256
5.13.2.5	dump_exe_log	256
5.13.2.6	dumpreg	257
5.13.2.7	eval_operand_val	257
5.13.2.8	evalsim_reg	258
5.13.2.9	exec_main	258
5.13.2.10	setsim_reg	259
5.13.3	Variable Documentation	260
5.13.3.1	cpu_state	260
5.13.3.2	do_stats	260
5.13.3.3	hist_exec_tail	260
5.13.3.4	pcnext	260
5.13.3.5	sbuf_total_cyc	260
5.13.3.6	sbuf_wait_cyc	260
5.14	cpu/common/labels.c File Reference	261
5.14.1	Define Documentation	262
5.14.1.1	LABELS_HASH_SIZE	262
5.14.2	Function Documentation	262
5.14.2.1	add_breakpoint	262
5.14.2.2	add_label	262
5.14.2.3	eval_label	262
5.14.2.4	find_label	262

5.14.2.5	get_label	262
5.14.2.6	has_breakpoint	262
5.14.2.7	init_breakpoints	262
5.14.2.8	init_labels	262
5.14.2.9	print_breakpoints	262
5.14.2.10	remove_breakpoint	262
5.14.3	Variable Documentation	262
5.14.3.1	breakpoints	262
5.14.3.2	label_hash	262
5.15	cpu/common/labels.h File Reference	263
5.15.1	Function Documentation	264
5.15.1.1	add_breakpoint	264
5.15.1.2	add_label	264
5.15.1.3	eval_label	264
5.15.1.4	find_label	264
5.15.1.5	get_label	264
5.15.1.6	has_breakpoint	264
5.15.1.7	init_breakpoints	264
5.15.1.8	init_labels	264
5.15.1.9	print_breakpoints	264
5.15.1.10	remove_breakpoint	264
5.15.2	Variable Documentation	264
5.15.2.1	breakpoints	264
5.16	cpu/common/parse.c File Reference	265
5.16.1	Define Documentation	266
5.16.1.1	IMM_STATS	266
5.16.1.2	MEMORY_LEN	266
5.16.2	Function Documentation	266
5.16.2.1	addprogram	266
5.16.2.2	identifyfile	267
5.16.2.3	loadcode	267
5.16.2.4	readfile_coff	268
5.16.2.5	readfile_elf	269
5.16.2.6	readsyms_coff	269
5.16.2.7	strstrip	269
5.16.2.8	translate	269

5.16.3	Variable Documentation	270
5.16.3.1	freemem	270
5.16.3.2	transl_error	270
5.16.3.3	transl_table	270
5.17	cpu/common/parse.h File Reference	271
5.17.1	Function Documentation	271
5.17.1.1	loadcode	271
5.17.1.2	rstrip	272
5.18	cpu/common/stats.c File Reference	273
5.18.1	Define Documentation	275
5.18.1.1	DSTATS_LEN	275
5.18.1.2	FSTATS_LEN	275
5.18.1.3	RAW_RANGE	275
5.18.1.4	SD	275
5.18.1.5	SSTATS_LEN	275
5.18.2	Function Documentation	275
5.18.2.1	addstats	275
5.18.2.2	addfstats	275
5.18.2.3	addsstats	275
5.18.2.4	initstats	275
5.18.2.5	printotherstats	275
5.18.2.6	printstats	275
5.18.3	Variable Documentation	275
5.18.3.1	dc_stats	275
5.18.3.2	dmmu_stats	275
5.18.3.3	dstats	275
5.18.3.4	fstats	276
5.18.3.5	func_unit_str	276
5.18.3.6	ic_stats	276
5.18.3.7	immu_stats	276
5.18.3.8	or1k_mstats	276
5.18.3.9	raw_stats	276
5.18.3.10	sstats	276
5.19	cpu/common/stats.h File Reference	277
5.19.1	Function Documentation	278
5.19.1.1	addstats	278

5.19.1.2	addfstats	278
5.19.1.3	addsstats	278
5.19.1.4	initstats	278
5.19.1.5	printstats	278
5.19.2	Variable Documentation	278
5.19.2.1	dc_stats	278
5.19.2.2	dmmu_stats	278
5.19.2.3	ic_stats	278
5.19.2.4	immu_stats	278
5.19.2.5	or1k_mstats	278
5.20	cpu/common/trace.c File Reference	279
5.20.1	Function Documentation	279
5.20.1.1	set_insnbrkpoint	279
5.21	cpu/common/trace.h File Reference	281
5.21.1	Function Documentation	281
5.21.1.1	set_insnbrkpoint	281
5.22	cpu/or1k/arch.h File Reference	283
5.22.1	Define Documentation	283
5.22.1.1	ADDR_C	283
5.22.1.2	LINK_REGNO	283
5.22.1.3	PRIdREG	283
5.22.1.4	PRIxADDR	284
5.22.1.5	PRIxREG	284
5.22.1.6	REG_C	284
5.22.2	Typedef Documentation	284
5.22.2.1	oraddr_t	284
5.22.2.2	orreg_t	284
5.22.2.3	uorreg_t	284
5.23	cpu/or1k/except.c File Reference	285
5.23.1	Function Documentation	286
5.23.1.1	DEFAULT_DEBUG_CHANNEL	286
5.23.1.2	except_handle	286
5.23.1.3	except_name	286
5.23.2	Variable Documentation	286
5.23.2.1	except_names	286
5.23.2.2	except_pending	286

5.24	cpu/or1k/except.h File Reference	287
5.24.1	Define Documentation	289
5.24.1.1	EXCEPT_ALIGN	289
5.24.1.2	EXCEPT_BUSERR	289
5.24.1.3	EXCEPT_DPF	289
5.24.1.4	EXCEPT_DTLBMISS	289
5.24.1.5	EXCEPT_ILLEGAL	289
5.24.1.6	EXCEPT_INT	289
5.24.1.7	EXCEPT_IPF	289
5.24.1.8	EXCEPT_ITLBMISS	289
5.24.1.9	EXCEPT_RANGE	289
5.24.1.10	EXCEPT_RESET	289
5.24.1.11	EXCEPT_SYSCALL	289
5.24.1.12	EXCEPT_TICK	289
5.24.1.13	EXCEPT_TRAP	289
5.24.2	Function Documentation	289
5.24.2.1	except_handle	289
5.24.3	Variable Documentation	290
5.24.3.1	except_pending	290
5.25	cpu/or1k/spr-defs.h File Reference	291
5.25.1	Define Documentation	298
5.25.1.1	MATCHPOINTS_TO_NDP	298
5.25.1.2	MAX_GRPS	300
5.25.1.3	MAX_MATCHPOINTS	300
5.25.1.4	MAX_SPRS	300
5.25.1.5	MAX_SPRS_PER_GRP	300
5.25.1.6	MAX_SPRS_PER_GRP_BITS	300
5.25.1.7	MAX_WATCHPOINTS	300
5.25.1.8	NOP_CNT_RESET	300
5.25.1.9	NOP_EXIT	300
5.25.1.10	NOP_NOP	300
5.25.1.11	NOP_PRINTF	300
5.25.1.12	NOP_PUTC	300
5.25.1.13	NOP_REPORT	300
5.25.1.14	NOP_REPORT_FIRST	300
5.25.1.15	NOP_REPORT_LAST	300

5.25.1.16 SPR_CPUCFGR	300
5.25.1.17 SPR_CPUCFGR_CGF	300
5.25.1.18 SPR_CPUCFGR_NS GF	300
5.25.1.19 SPR_CPUCFGR_OB32S	300
5.25.1.20 SPR_CPUCFGR_OB64S	300
5.25.1.21 SPR_CPUCFGR_OF32S	300
5.25.1.22 SPR_CPUCFGR_OF64S	300
5.25.1.23 SPR_CPUCFGR_OV64S	300
5.25.1.24 SPR_CPUCFGR_RES	300
5.25.1.25 SPR_DCBFR	300
5.25.1.26 SPR_DCBIR	300
5.25.1.27 SPR_DCBLR	300
5.25.1.28 SPR_DCBPR	300
5.25.1.29 SPR_DCBWR	300
5.25.1.30 SPR_DCCFGR	300
5.25.1.31 SPR_DCCFGR_CBFRI	300
5.25.1.32 SPR_DCCFGR_CBIRI	300
5.25.1.33 SPR_DCCFGR_CBLRI	300
5.25.1.34 SPR_DCCFGR_CBPRI	300
5.25.1.35 SPR_DCCFGR_CBS	300
5.25.1.36 SPR_DCCFGR_CBS_OFF	300
5.25.1.37 SPR_DCCFGR_CBWBRI	300
5.25.1.38 SPR_DCCFGR_CCRI	300
5.25.1.39 SPR_DCCFGR_CWS	300
5.25.1.40 SPR_DCCFGR_NCS	300
5.25.1.41 SPR_DCCFGR_NCS_OFF	300
5.25.1.42 SPR_DCCFGR_NCW	300
5.25.1.43 SPR_DCCFGR_NCW_OFF	300
5.25.1.44 SPR_DCCR	300
5.25.1.45 SPR_DCCR_EW	300
5.25.1.46 SPR_DCFGR	300
5.25.1.47 SPR_DCFGR_NDP	300
5.25.1.48 SPR_DCFGR_NDP1	300
5.25.1.49 SPR_DCFGR_NDP2	300
5.25.1.50 SPR_DCFGR_NDP3	300
5.25.1.51 SPR_DCFGR_NDP4	300

5.25.1.52 SPR_DCFGR_NDP5	300
5.25.1.53 SPR_DCFGR_NDP6	300
5.25.1.54 SPR_DCFGR_NDP7	300
5.25.1.55 SPR_DCFGR_NDP8	300
5.25.1.56 SPR_DCFGR_WPCI	300
5.25.1.57 SPR_DCR	300
5.25.1.58 SPR_DCR_BASE	300
5.25.1.59 SPR_DCR_CC	300
5.25.1.60 SPR_DCR_CC_EQUAL	300
5.25.1.61 SPR_DCR_CC_GREAT	300
5.25.1.62 SPR_DCR_CC_GREATER	300
5.25.1.63 SPR_DCR_CC_LESS	300
5.25.1.64 SPR_DCR_CC_LESSE	300
5.25.1.65 SPR_DCR_CC_MASKED	300
5.25.1.66 SPR_DCR_CC_NEQUAL	300
5.25.1.67 SPR_DCR_CT	300
5.25.1.68 SPR_DCR_CT_DISABLED	300
5.25.1.69 SPR_DCR_CT_IFEA	300
5.25.1.70 SPR_DCR_CT_LD	300
5.25.1.71 SPR_DCR_CT_LEA	300
5.25.1.72 SPR_DCR_CT_LSD	300
5.25.1.73 SPR_DCR_CT_LSEA	300
5.25.1.74 SPR_DCR_CT_SD	300
5.25.1.75 SPR_DCR_CT_SEA	300
5.25.1.76 SPR_DCR_DP	300
5.25.1.77 SPR_DCR_LAST	300
5.25.1.78 SPR_DCR_SC	300
5.25.1.79 SPR_DMMUCFGR	300
5.25.1.80 SPR_DMMUCFGR_CRI	300
5.25.1.81 SPR_DMMUCFGR_HTR	300
5.25.1.82 SPR_DMMUCFGR_NAE	300
5.25.1.83 SPR_DMMUCFGR_NTS	300
5.25.1.84 SPR_DMMUCFGR_NTS_OFF	300
5.25.1.85 SPR_DMMUCFGR_NTW	300
5.25.1.86 SPR_DMMUCFGR_NTW_OFF	300
5.25.1.87 SPR_DMMUCFGR_PRI	300

5.25.1.88	SPR_DMMUCFGR_TEIRI	300
5.25.1.89	SPR_DMMUCR	300
5.25.1.90	SPR_DMMUCR_P1S	300
5.25.1.91	SPR_DMMUCR_P2S	300
5.25.1.92	SPR_DMMUCR_PADDR_WIDTH	300
5.25.1.93	SPR_DMMUCR_VADDR_WIDTH	300
5.25.1.94	SPR_DMR1	300
5.25.1.95	SPR_DMR1_BT	300
5.25.1.96	SPR_DMR1_CW	300
5.25.1.97	SPR_DMR1_CW0	300
5.25.1.98	SPR_DMR1_CW0_AND	300
5.25.1.99	SPR_DMR1_CW0_OR	300
5.25.1.100	SPR_DMR1_CW1	300
5.25.1.101	SPR_DMR1_CW1_AND	300
5.25.1.102	SPR_DMR1_CW1_OR	300
5.25.1.103	SPR_DMR1_CW2	300
5.25.1.104	SPR_DMR1_CW2_AND	300
5.25.1.105	SPR_DMR1_CW2_OR	300
5.25.1.106	SPR_DMR1_CW3	300
5.25.1.107	SPR_DMR1_CW3_AND	300
5.25.1.108	SPR_DMR1_CW3_OR	300
5.25.1.109	SPR_DMR1_CW4	300
5.25.1.110	SPR_DMR1_CW4_AND	300
5.25.1.111	SPR_DMR1_CW4_OR	300
5.25.1.112	SPR_DMR1_CW5	300
5.25.1.113	SPR_DMR1_CW5_AND	300
5.25.1.114	SPR_DMR1_CW5_OR	300
5.25.1.115	SPR_DMR1_CW6	300
5.25.1.116	SPR_DMR1_CW6_AND	300
5.25.1.117	SPR_DMR1_CW6_OR	300
5.25.1.118	SPR_DMR1_CW7	300
5.25.1.119	SPR_DMR1_CW7_AND	300
5.25.1.120	SPR_DMR1_CW7_OR	300
5.25.1.121	SPR_DMR1_CW8	300
5.25.1.122	SPR_DMR1_CW8_AND	300
5.25.1.123	SPR_DMR1_CW8_OR	300

5.25.1.124	SPR_DMR1_CW9	300
5.25.1.125	SPR_DMR1_CW9_AND	300
5.25.1.126	SPR_DMR1_CW9_OR	300
5.25.1.127	SPR_DMR1_RES1	300
5.25.1.128	SPR_DMR1_RES2	300
5.25.1.129	SPR_DMR1_ST	300
5.25.1.130	SPR_DMR2	300
5.25.1.131	SPR_DMR2_AWTC	300
5.25.1.132	SPR_DMR2_AWTC_OFF	300
5.25.1.133	SPR_DMR2_WBS	300
5.25.1.134	SPR_DMR2_WBS_OFF	300
5.25.1.135	SPR_DMR2_WCE0	300
5.25.1.136	SPR_DMR2_WCE1	300
5.25.1.137	SPR_DMR2_WGB	300
5.25.1.138	SPR_DMR2_WGB_OFF	300
5.25.1.139	SPR_DRR	300
5.25.1.140	SPR_DRR_AE	300
5.25.1.141	SPR_DRR_BUSEE	300
5.25.1.142	SPR_DRR_DME	300
5.25.1.143	SPR_DRR_DPFE	300
5.25.1.144	SPR_DRR_IE	300
5.25.1.145	SPR_DRR_IIE	300
5.25.1.146	SPR_DRR_IME	300
5.25.1.147	SPR_DRR_IPFE	300
5.25.1.148	SPR_DRR_RE	300
5.25.1.149	SPR_DRR_RST	300
5.25.1.150	SPR_DRR_SCE	300
5.25.1.151	SPR_DRR_TE	300
5.25.1.152	SPR_DRR_TTE	300
5.25.1.153	SPR_DSR	300
5.25.1.154	SPR_DSR_AE	300
5.25.1.155	SPR_DSR_BUSEE	300
5.25.1.156	SPR_DSR_DME	300
5.25.1.157	SPR_DSR_DPFE	300
5.25.1.158	SPR_DSR_IE	300
5.25.1.159	SPR_DSR_IIE	300

5.25.1.160	SPR_DSR_IME	300
5.25.1.161	SPR_DSR_IPFE	300
5.25.1.162	SPR_DSR_RE	300
5.25.1.163	SPR_DSR_RSTE	300
5.25.1.164	SPR_DSR_SCE	300
5.25.1.165	SPR_DSR_SSE	300
5.25.1.166	SPR_DSR_TE	300
5.25.1.167	SPR_DSR_TTE	300
5.25.1.168	SPR_DTLBMR_BASE	300
5.25.1.169	SPR_DTLBMR_CID	300
5.25.1.170	SPR_DTLBMR_LAST	300
5.25.1.171	SPR_DTLBMR_LRU	300
5.25.1.172	SPR_DTLBMR_PL1	300
5.25.1.173	SPR_DTLBMR_V	300
5.25.1.174	SPR_DTLBMR_VPN	300
5.25.1.175	SPR_DTLBTR_A	300
5.25.1.176	SPR_DTLBTR_BASE	300
5.25.1.177	SPR_DTLBTR_CC	300
5.25.1.178	SPR_DTLBTR_CI	300
5.25.1.179	SPR_DTLBTR_D	300
5.25.1.180	SPR_DTLBTR_LAST	300
5.25.1.181	SPR_DTLBTR_PPN	300
5.25.1.182	SPR_DTLBTR_SRE	300
5.25.1.183	SPR_DTLBTR_SWE	300
5.25.1.184	SPR_DTLBTR_URE	300
5.25.1.185	SPR_DTLBTR_UWE	300
5.25.1.186	SPR_DTLBTR_WBC	300
5.25.1.187	SPR_DTLBTR_WOM	300
5.25.1.188	SPR_DVR	300
5.25.1.189	SPR_DWCR0	300
5.25.1.190	SPR_DWCR1	300
5.25.1.191	SPR_DWCR_COUNT	300
5.25.1.192	SPR_DWCR_MATCH	300
5.25.1.193	SPR_DWCR_MATCH_OFF	300
5.25.1.194	SPR_EEAR_BASE	300
5.25.1.195	SPR_EEAR_LAST	300

5.25.1.196	SPR_EPCR_BASE	300
5.25.1.197	SPR_EPCR_LAST	300
5.25.1.198	SPR_ESR_BASE	300
5.25.1.199	SPR_ESR_LAST	300
5.25.1.200	SPR_ICBIR	300
5.25.1.201	SPR_ICBLR	300
5.25.1.202	SPR_ICBPR	300
5.25.1.203	SPR_ICCFGR	300
5.25.1.204	SPR_ICCFGR_CBIRI	300
5.25.1.205	SPR_ICCFGR_CBLRI	300
5.25.1.206	SPR_ICCFGR_CBPRI	300
5.25.1.207	SPR_ICCFGR_CBS	300
5.25.1.208	SPR_ICCFGR_CBS_OFF	300
5.25.1.209	SPR_ICCFGR_CCRI	300
5.25.1.210	SPR_ICCFGR_NCS	300
5.25.1.211	SPR_ICCFGR_NCS_OFF	300
5.25.1.212	SPR_ICCFGR_NCW	300
5.25.1.213	SPR_ICCFGR_NCW_OFF	300
5.25.1.214	SPR_ICCR	300
5.25.1.215	SPR_ICCR_EW	300
5.25.1.216	SPR_ICR_BASE	300
5.25.1.217	SPR_ICR_LAST	300
5.25.1.218	SPR_IMMUCFGR	300
5.25.1.219	SPR_IMMUCFGR_CRI	300
5.25.1.220	SPR_IMMUCFGR_HTR	300
5.25.1.221	SPR_IMMUCFGR_NAE	300
5.25.1.222	SPR_IMMUCFGR_NTS	300
5.25.1.223	SPR_IMMUCFGR_NTS_OFF	300
5.25.1.224	SPR_IMMUCFGR_NTW	300
5.25.1.225	SPR_IMMUCFGR_NTW_OFF	300
5.25.1.226	SPR_IMMUCFGR_PRI	300
5.25.1.227	SPR_IMMUCFGR_TEIRI	300
5.25.1.228	SPR_IMMUCR	300
5.25.1.229	SPR_IMMUCR_P1S	300
5.25.1.230	SPR_IMMUCR_P2S	300
5.25.1.231	SPR_IMMUCR_PADDR_WIDTH	300

5.25.1.232	SPR_IMMUCR_VADDR_WIDTH	300
5.25.1.233	SPR_ITLBMR_BASE	300
5.25.1.234	SPR_ITLBMR_CID	300
5.25.1.235	SPR_ITLBMR_LAST	300
5.25.1.236	SPR_ITLBMR_LRU	300
5.25.1.237	SPR_ITLBMR_PL1	300
5.25.1.238	SPR_ITLBMR_V	300
5.25.1.239	SPR_ITLBMR_VPN	300
5.25.1.240	SPR_ITLBTR_A	300
5.25.1.241	SPR_ITLBTR_BASE	300
5.25.1.242	SPR_ITLBTR_CC	300
5.25.1.243	SPR_ITLBTR_CI	300
5.25.1.244	SPR_ITLBTR_D	300
5.25.1.245	SPR_ITLBTR_LAST	300
5.25.1.246	SPR_ITLBTR_PPN	300
5.25.1.247	SPR_ITLBTR_SXE	300
5.25.1.248	SPR_ITLBTR_UXE	300
5.25.1.249	SPR_ITLBTR_WBC	300
5.25.1.250	SPR_ITLBTR_WOM	300
5.25.1.251	SPR_MACHI	300
5.25.1.252	SPR_MACLO	300
5.25.1.253	SPR_NPC	300
5.25.1.254	SPR_PCCFGR	300
5.25.1.255	SPR_PCCR	300
5.25.1.256	SPR_PCMR	300
5.25.1.257	SPR_PCMR_BS	300
5.25.1.258	SPR_PCMR_CISM	300
5.25.1.259	SPR_PCMR_CIUM	300
5.25.1.260	SPR_PCMR_CP	300
5.25.1.261	SPR_PCMR_DCM	300
5.25.1.262	SPR_PCMR_DDS	300
5.25.1.263	SPR_PCMR_DTLBM	300
5.25.1.264	SPR_PCMR_ICM	300
5.25.1.265	SPR_PCMR_IF	300
5.25.1.266	SPR_PCMR_IFS	300
5.25.1.267	SPR_PCMR_ITLBM	300

5.25.1.268	SPR_PCMR_LA	300
5.25.1.269	SPR_PCMR_LSUS	300
5.25.1.270	SPR_PCMR_SA	300
5.25.1.271	SPR_PCMR_UMRA	300
5.25.1.272	SPR_PCMR_WPE	300
5.25.1.273	SPR_PICMR	300
5.25.1.274	SPR_PICMR_IUM	300
5.25.1.275	SPR_PICPR	300
5.25.1.276	SPR_PICPR_IPRIO	300
5.25.1.277	SPR_PICSR	300
5.25.1.278	SPR_PICSR_IS	300
5.25.1.279	SPR_PMR	300
5.25.1.280	SPR_PMR_DCGE	300
5.25.1.281	SPR_PMR_DME	300
5.25.1.282	SPR_PMR_SDF	300
5.25.1.283	SPR_PMR_SME	300
5.25.1.284	SPR_PMR_SUME	300
5.25.1.285	SPR_PPC	300
5.25.1.286	SPR_SR	300
5.25.1.287	SPR_SR_CE	300
5.25.1.288	SPR_SR_CID	300
5.25.1.289	SPR_SR_CY	300
5.25.1.290	SPR_SR_DCE	300
5.25.1.291	SPR_SR_DME	300
5.25.1.292	SPR_SR_DSX	300
5.25.1.293	SPR_SR_EPH	300
5.25.1.294	SPR_SR_F	300
5.25.1.295	SPR_SR_FO	300
5.25.1.296	SPR_SR_ICE	300
5.25.1.297	SPR_SR_IEE	300
5.25.1.298	SPR_SR_IME	300
5.25.1.299	SPR_SR_LEE	300
5.25.1.300	SPR_SR_OV	300
5.25.1.301	SPR_SR_OVE	300
5.25.1.302	SPR_SR_RES	300
5.25.1.303	SPR_SR_SM	300

5.25.1.304	SPR_SR_SUMRA	300
5.25.1.305	SPR_SR_TEE	300
5.25.1.306	SPR_TTCR	300
5.25.1.307	SPR_TTCR_PERIOD	300
5.25.1.308	SPR_TTMR	300
5.25.1.309	SPR_TTMR_CR	300
5.25.1.310	SPR_TTMR_IE	300
5.25.1.311	SPR_TTMR_IP	300
5.25.1.312	SPR_TTMR_M	300
5.25.1.313	SPR_TTMR_PERIOD	300
5.25.1.314	SPR_TTMR_RT	300
5.25.1.315	SPR_TTMR_SR	300
5.25.1.316	SPR_UPR	300
5.25.1.317	SPR_UPR_CUP	300
5.25.1.318	SPR_UPR_DCP	300
5.25.1.319	SPR_UPR_DMP	300
5.25.1.320	SPR_UPR_DUP	300
5.25.1.321	SPR_UPR_ICP	300
5.25.1.322	SPR_UPR_IMP	300
5.25.1.323	SPR_UPR_MP	300
5.25.1.324	SPR_UPR_PCUP	300
5.25.1.325	SPR_UPR_PICP	300
5.25.1.326	SPR_UPR_PMP	300
5.25.1.327	SPR_UPR_RES	300
5.25.1.328	SPR_UPR_TTP	300
5.25.1.329	SPR_UPR_UP	300
5.25.1.330	SPR_VR	300
5.25.1.331	SPR_VR_CFG	300
5.25.1.332	SPR_VR_CFG_OFF	300
5.25.1.333	SPR_VR_RES	300
5.25.1.334	SPR_VR_REV	300
5.25.1.335	SPR_VR_REV_OFF	300
5.25.1.336	SPR_VR_VER	300
5.25.1.337	SPR_VR_VER_OFF	300
5.25.1.338	SPRGROUP_D	300
5.25.1.339	SPRGROUP_DC	300

5.25.1.340	SPRGROUP_DMMU	300
5.25.1.341	SPRGROUP_IC	300
5.25.1.342	SPRGROUP_IMMU	300
5.25.1.343	SPRGROUP_MAC	300
5.25.1.344	SPRGROUP_PC	300
5.25.1.345	SPRGROUP_PIC	300
5.25.1.346	SPRGROUP_PM	300
5.25.1.347	SPRGROUP_SYS	300
5.25.1.348	SPRGROUP_TT	300
5.26	cpu/or1k/spr-dump.c File Reference	301
5.26.1	Function Documentation	302
5.26.1.1	DEFAULT_DEBUG_CHANNEL	302
5.26.1.2	dump_spr	302
5.26.2	Variable Documentation	302
5.26.2.1	ret_spr	302
5.26.2.2	spr_cpufgr	302
5.26.2.3	spr_d_group	303
5.26.2.4	spr_dc_group	303
5.26.2.5	spr_dccfgr	303
5.26.2.6	spr_dccr	304
5.26.2.7	spr_dcfgr	304
5.26.2.8	spr_dmmu_group	304
5.26.2.9	spr_dmmucfgr	304
5.26.2.10	spr_dmmucr	304
5.26.2.11	spr_dmr1	305
5.26.2.12	spr_dmr2	305
5.26.2.13	spr_drr	305
5.26.2.14	spr_dsr	306
5.26.2.15	spr_dtlbmr	306
5.26.2.16	spr_dtlbtr	306
5.26.2.17	spr_dwcr	306
5.26.2.18	spr_groups	307
5.26.2.19	spr_ic_group	307
5.26.2.20	spr_iccfgr	307
5.26.2.21	spr_iccr	307
5.26.2.22	spr_immu_group	308

5.26.2.23	spr_immucfgr	308
5.26.2.24	spr_immucr	308
5.26.2.25	spr_itlbmr	308
5.26.2.26	spr_itlbtr	309
5.26.2.27	spr_mac_group	309
5.26.2.28	spr_one_val	309
5.26.2.29	spr_pc_group	309
5.26.2.30	spr_pcmr	309
5.26.2.31	spr_pic_group	310
5.26.2.32	spr_picmr	310
5.26.2.33	spr_pm_group	310
5.26.2.34	spr_pmr	310
5.26.2.35	spr_sr	311
5.26.2.36	spr_sys_group	311
5.26.2.37	spr_tt_group	311
5.26.2.38	spr_ttmr	312
5.26.2.39	spr_upr	312
5.26.2.40	spr_vr	312
5.27	cpu/or1k/spr-dump.h File Reference	313
5.27.1	Function Documentation	313
5.27.1.1	dump_spr	313
5.28	cpu/or1k/sprs.c File Reference	314
5.28.1	Function Documentation	315
5.28.1.1	DECLARE_DEBUG_CHANNEL	315
5.28.1.2	DEFAULT_DEBUG_CHANNEL	315
5.28.1.3	mfspr	315
5.28.1.4	mtspr	315
5.28.1.5	sprs_status	316
5.28.2	Variable Documentation	316
5.28.2.1	audio_cnt	316
5.28.2.2	fo	316
5.29	cpu/or1k/sprs.h File Reference	317
5.29.1	Function Documentation	317
5.29.1.1	dump_spr	317
5.29.1.2	mfspr	318
5.29.1.3	mtspr	318

5.29.1.4	sprs_status	318
5.30	cpu/or32/common_i386.h File Reference	319
5.30.1	Function Documentation	319
5.30.1.1	get_pc	319
5.30.1.2	set_pc	319
5.30.1.3	upd_sim_cycles	319
5.30.2	Variable Documentation	319
5.30.2.1	high32	319
5.30.2.2	low32	319
5.30.2.3	useless_x86	319
5.30.2.4	val3232	319
5.30.2.5	val64	319
5.31	cpu/or32/def_op_t.h File Reference	320
5.31.1	Define Documentation	320
5.31.1.1	DEF_1T_OP	320
5.31.1.2	DEF_2T_OP	320
5.31.1.3	DEF_2T_OP_NEQ	320
5.31.1.4	DEF_3T_OP	321
5.31.1.5	DEF_3T_OP_NEQ	321
5.31.1.6	DEF_GPR_OP	321
5.31.1.7	GPR_T	321
5.32	cpu/or32/dyn32_defs.h File Reference	322
5.32.1	Function Documentation	322
5.32.1.1	l_none	322
5.32.1.2	PARAMS	322
5.33	cpu/or32/dyn_rec.c File Reference	323
5.33.1	Define Documentation	330
5.33.1.1	OPS_ENLARGE_BY	330
5.33.1.2	RECED_PAGE_ENLARGE_BY	330
5.33.1.3	T_NONE	330
5.33.1.4	TFLAG_DST	330
5.33.1.5	TFLAG_SAVED	330
5.33.1.6	TFLAG_SOURCED	330
5.33.1.7	TFLAG_SRC	330
5.33.2	Typedef Documentation	330
5.33.2.1	generic_gen_op	330

5.33.2.2	imm_gen_op	330
5.33.3	Function Documentation	330
5.33.3.1	add_to_op_params	330
5.33.3.2	add_to_opq	330
5.33.3.3	DEF_1T_OP	330
5.33.3.4	DEF_1T_OP	330
5.33.3.5	DEF_1T_OP	330
5.33.3.6	DEF_1T_OP	330
5.33.3.7	DEF_1T_OP	330
5.33.3.8	DEF_1T_OP	330
5.33.3.9	DEF_1T_OP	330
5.33.3.10	DEF_1T_OP	330
5.33.3.11	DEF_1T_OP	330
5.33.3.12	DEF_1T_OP	330
5.33.3.13	DEF_1T_OP	330
5.33.3.14	DEF_1T_OP	330
5.33.3.15	DEF_1T_OP	330
5.33.3.16	DEF_1T_OP	330
5.33.3.17	DEF_1T_OP	330
5.33.3.18	DEF_1T_OP	330
5.33.3.19	DEF_1T_OP	330
5.33.3.20	DEF_1T_OP	330
5.33.3.21	DEF_1T_OP	330
5.33.3.22	DEF_1T_OP	330
5.33.3.23	DEF_1T_OP	330
5.33.3.24	DEF_1T_OP	330
5.33.3.25	DEF_1T_OP	330
5.33.3.26	DEF_1T_OP	330
5.33.3.27	DEF_1T_OP	330
5.33.3.28	DEF_1T_OP	330
5.33.3.29	DEF_1T_OP	330
5.33.3.30	DEF_1T_OP	330
5.33.3.31	DEF_1T_OP	330
5.33.3.32	DEF_1T_OP	330
5.33.3.33	DEF_1T_OP	330
5.33.3.34	DEF_1T_OP	330

5.33.3.35 DEF_1T_OP	330
5.33.3.36 DEF_1T_OP	330
5.33.3.37 DEF_1T_OP	330
5.33.3.38 DEF_1T_OP	330
5.33.3.39 DEF_1T_OP	330
5.33.3.40 DEF_1T_OP	330
5.33.3.41 DEF_1T_OP	330
5.33.3.42 DEF_1T_OP	330
5.33.3.43 DEF_1T_OP	330
5.33.3.44 DEF_1T_OP	330
5.33.3.45 DEF_2T_OP	330
5.33.3.46 DEF_2T_OP	330
5.33.3.47 DEF_2T_OP	330
5.33.3.48 DEF_2T_OP	330
5.33.3.49 DEF_2T_OP	330
5.33.3.50 DEF_2T_OP	330
5.33.3.51 DEF_2T_OP	330
5.33.3.52 DEF_2T_OP	330
5.33.3.53 DEF_2T_OP	330
5.33.3.54 DEF_2T_OP	330
5.33.3.55 DEF_2T_OP	330
5.33.3.56 DEF_2T_OP	330
5.33.3.57 DEF_2T_OP	330
5.33.3.58 DEF_2T_OP	330
5.33.3.59 DEF_2T_OP	330
5.33.3.60 DEF_2T_OP	330
5.33.3.61 DEF_2T_OP	330
5.33.3.62 DEF_2T_OP	330
5.33.3.63 DEF_2T_OP	330
5.33.3.64 DEF_2T_OP	330
5.33.3.65 DEF_2T_OP	330
5.33.3.66 DEF_2T_OP	330
5.33.3.67 DEF_2T_OP	330
5.33.3.68 DEF_2T_OP	330
5.33.3.69 DEF_2T_OP	330
5.33.3.70 DEF_2T_OP	330

5.33.3.71 DEF_2T_OP	330
5.33.3.72 DEF_2T_OP	330
5.33.3.73 DEF_2T_OP	330
5.33.3.74 DEF_2T_OP	330
5.33.3.75 DEF_2T_OP	330
5.33.3.76 DEF_2T_OP	330
5.33.3.77 DEF_2T_OP	330
5.33.3.78 DEF_2T_OP	330
5.33.3.79 DEF_2T_OP	330
5.33.3.80 DEF_2T_OP	330
5.33.3.81 DEF_2T_OP	330
5.33.3.82 DEF_2T_OP_NEQ	330
5.33.3.83 DEF_3T_OP	330
5.33.3.84 DEF_3T_OP	330
5.33.3.85 DEF_3T_OP	330
5.33.3.86 DEF_3T_OP	330
5.33.3.87 DEF_3T_OP	330
5.33.3.88 DEF_3T_OP	330
5.33.3.89 DEF_3T_OP	330
5.33.3.90 DEF_3T_OP	330
5.33.3.91 DEF_3T_OP	330
5.33.3.92 DEF_3T_OP	330
5.33.3.93 DEF_3T_OP_NEQ	330
5.33.3.94 DEF_3T_OP_NEQ	330
5.33.3.95 DEF_3T_OP_NEQ	330
5.33.3.96 DEF_3T_OP_NEQ	330
5.33.3.97 DEF_GPR_OP	330
5.33.3.98 DEF_GPR_OP	330
5.33.3.99 dirtyfy_page	330
5.33.3.100dyn_checkwrite	331
5.33.3.101dyn_main	331
5.33.3.102dyn_ret_stack_prot	332
5.33.3.103dyn_sigsegv_debug	332
5.33.3.104enough_host_page	332
5.33.3.105eval_insn_ops	332
5.33.3.106find_jump_loc	332

5.33.3.107	find_t	333
5.33.3.108	gen_j_imm	333
5.33.3.109	gen_j_reg	333
5.33.3.110	gen_l_add	333
5.33.3.111	gen_l_addc	333
5.33.3.112	gen_l_and	333
5.33.3.113	gen_l_bf	333
5.33.3.114	gen_l_bnf	334
5.33.3.115	gen_l_cmov	335
5.33.3.116	gen_l_cust1	335
5.33.3.117	gen_l_cust2	335
5.33.3.118	gen_l_cust3	335
5.33.3.119	gen_l_cust4	335
5.33.3.120	gen_l_cust5	335
5.33.3.121	gen_l_cust6	335
5.33.3.122	gen_l_cust7	335
5.33.3.123	gen_l_cust8	335
5.33.3.124	gen_l_div	335
5.33.3.125	gen_l_divu	335
5.33.3.126	gen_l_extbs	335
5.33.3.127	gen_l_extbz	335
5.33.3.128	gen_l_exths	335
5.33.3.129	gen_l_exthz	335
5.33.3.130	gen_l_extws	335
5.33.3.131	gen_l_extwz	335
5.33.3.132	gen_l_ff1	335
5.33.3.133	gen_l_invalid	335
5.33.3.134	gen_l_j	335
5.33.3.135	gen_l_jal	336
5.33.3.136	gen_l_jalr	336
5.33.3.137	gen_l_jr	336
5.33.3.138	gen_l_lbs	336
5.33.3.139	gen_l_lbz	337
5.33.3.140	gen_l_lhs	337
5.33.3.141	gen_l_lhz	337
5.33.3.142	gen_l_lws	337

5.33.3.143	gen_l_lwz	337
5.33.3.144	gen_l_mac	338
5.33.3.145	gen_l_macrc	338
5.33.3.146	gen_l_mfspr	338
5.33.3.147	gen_l_movhi	338
5.33.3.148	gen_l_msb	338
5.33.3.149	gen_l_mtspr	338
5.33.3.150	gen_l_mul	338
5.33.3.151	gen_l_mulu	338
5.33.3.152	gen_l_nop	338
5.33.3.153	gen_l_or	338
5.33.3.154	gen_l_rfe	338
5.33.3.155	gen_l_sb	338
5.33.3.156	gen_l_sfeq	339
5.33.3.157	gen_l_sfges	339
5.33.3.158	gen_l_sfgeu	339
5.33.3.159	gen_l_sfgts	339
5.33.3.160	gen_l_sfgtu	339
5.33.3.161	gen_l_sfles	339
5.33.3.162	gen_l_sfleu	339
5.33.3.163	gen_l_sftts	339
5.33.3.164	gen_l_sfttu	339
5.33.3.165	gen_l_sfne	339
5.33.3.166	gen_l_sh	339
5.33.3.167	gen_l_sll	339
5.33.3.168	gen_l_sra	339
5.33.3.169	gen_l_srl	339
5.33.3.170	gen_l_sub	339
5.33.3.171	gen_l_sw	339
5.33.3.172	gen_l_sys	340
5.33.3.173	gen_l_trap	340
5.33.3.174	gen_l_xor	340
5.33.3.175	gen_lf_add_s	340
5.33.3.176	gen_lf_div_s	340
5.33.3.177	gen_lf_ftoi_s	340
5.33.3.178	gen_lf_itof_s	340

5.33.3.179	gen_lf_madd_s	340
5.33.3.180	gen_lf_mul_s	341
5.33.3.181	gen_lf_rem_s	341
5.33.3.182	gen_lf_sfreq_s	341
5.33.3.183	gen_lf_sfge_s	341
5.33.3.184	gen_lf_sfgt_s	341
5.33.3.185	gen_lf_sfle_s	341
5.33.3.186	gen_lf_sflt_s	342
5.33.3.187	gen_lf_sfne_s	342
5.33.3.188	gen_lf_sub_s	342
5.33.3.189	gen_op_mark_loc	342
5.33.3.190	immu_retranslate	343
5.33.3.191	linit_dyn_recomp	343
5.33.3.192	new_dp	344
5.33.3.193	recheck_immu	344
5.33.3.194	recompile_delay_insn	344
5.33.3.195	recompile_insn	344
5.33.3.196	recompile_page	345
5.33.3.197	run_sched_out_of_line	345
5.33.3.198	ship_gprs_out_t	345
5.33.3.199	ship_t_out	346
5.33.4	Variable Documentation	346
5.33.4.1	__op_param1	346
5.33.4.2	__op_param2	346
5.33.4.3	__op_param3	346
5.33.4.4	do_stats	346
5.33.4.5	set_pc_delay_gpr	346
5.33.4.6	sigsegv_addr	346
5.33.4.7	sigsegv_state	346
5.34	cpu/or32/dyn_rec.h File Reference	347
5.34.1	Define Documentation	348
5.34.1.1	glue	348
5.34.1.2	IMMU_GOT_DISABLED	348
5.34.1.3	IMMU_GOT_ENABLED	348
5.34.1.4	xglue	348
5.34.2	Typedef Documentation	348

5.34.2.1	gen_code_ent	348
5.34.3	Function Documentation	348
5.34.3.1	add_to_op_params	348
5.34.3.2	add_to_opq	348
5.34.3.3	dyn_checkwrite	348
5.34.3.4	dyn_main	349
5.34.3.5	enough_host_page	349
5.34.3.6	enter_dyn_code	349
5.34.3.7	init_dyn_recomp	349
5.34.3.8	new_dp	350
5.34.3.9	recheck_immu	350
5.34.3.10	recompile_page	350
5.34.3.11	run_sched_out_of_line	351
5.34.4	Variable Documentation	351
5.34.4.1	rec_stack_base	351
5.35	cpu/or32/dyngen.c File Reference	352
5.35.1	Define Documentation	353
5.35.1.1	MAX_PARAMS	353
5.35.1.2	OP_FUNC_PARAM_PREFIX	353
5.35.1.3	OP_FUNC_PREFIX	353
5.35.2	Function Documentation	353
5.35.2.1	gen_func_proto	353
5.35.2.2	main	353
5.35.3	Variable Documentation	353
5.35.3.1	c_file_head	353
5.35.3.2	c_rel_file_head	353
5.35.3.3	c_rel_file_tail	353
5.35.3.4	c_sw_file_head	354
5.35.3.5	c_sw_file_tail	354
5.35.3.6	gen_code_proto	354
5.36	cpu/or32/dyngen.h File Reference	355
5.36.1	Variable Documentation	355
5.36.1.1	archfs	355
5.36.1.2	bffs	355
5.37	cpu/or32/dyngen_elf.c File Reference	356
5.37.1	Function Documentation	356

5.37.1.1	<code>elf_close_obj</code>	356
5.37.1.2	<code>elf_find_func</code>	356
5.37.1.3	<code>elf_get_func_len</code>	356
5.37.1.4	<code>elf_get_func_name</code>	357
5.37.1.5	<code>elf_get_func_reloc</code>	357
5.37.1.6	<code>elf_get_func_start</code>	357
5.37.1.7	<code>elf_get_sym_name</code>	357
5.37.1.8	<code>elf_open_obj</code>	357
5.37.2	Variable Documentation	357
5.37.2.1	<code>bffs</code>	357
5.38	<code>cpu/or32/dyngen_i386.c</code> File Reference	358
5.38.1	Define Documentation	358
5.38.1.1	<code>RET_OPCODE</code>	358
5.38.2	Function Documentation	358
5.38.2.1	<code>i386_gen_func_reloc</code>	358
5.38.2.2	<code>i386_gen_reloc</code>	358
5.38.2.3	<code>i386_get_real_func_len</code>	358
5.38.3	Variable Documentation	358
5.38.3.1	<code>archfs</code>	358
5.39	<code>cpu/or32/execute.c</code> File Reference	360
5.39.1	Define Documentation	361
5.39.1.1	<code>SET_OV_FLAG_FN</code>	361
5.39.2	Function Documentation	361
5.39.2.1	<code>analysis</code>	361
5.39.2.2	<code>check_depend</code>	362
5.39.2.3	<code>cpu_clock</code>	362
5.39.2.4	<code>cpu_reset</code>	363
5.39.2.5	<code>decode_execute</code>	363
5.39.2.6	<code>decode_execute_wrapper</code>	363
5.39.2.7	<code>dump_exe_log</code>	364
5.39.2.8	<code>dumpreg</code>	364
5.39.2.9	<code>eval_operand_val</code>	365
5.39.2.10	<code>evalsim_reg</code>	365
5.39.2.11	<code>exec_main</code>	366
5.39.2.12	<code>fetch</code>	366
5.39.2.13	<code>l_invalid</code>	367

5.39.2.14	sbuf_load	367
5.39.2.15	sbuf_store	368
5.39.2.16	setsim_reg	368
5.39.2.17	update_pc	368
5.39.3	Variable Documentation	368
5.39.3.1	breakpoint	368
5.39.3.2	cpu_state	368
5.39.3.3	do_stats	368
5.39.3.4	hist_exec_tail	368
5.39.3.5	issued_per_cycle	369
5.39.3.6	multissue	369
5.39.3.7	next_delay_insn	369
5.39.3.8	pcnext	369
5.39.3.9	sbuf_buf	369
5.39.3.10	sbuf_count	369
5.39.3.11	sbuf_head	369
5.39.3.12	sbuf_prev_cycles	369
5.39.3.13	sbuf_tail	369
5.39.3.14	sbuf_total_cyc	369
5.39.3.15	sbuf_wait_cyc	369
5.40	cpu/or32/generate.c File Reference	370
5.40.1	Function Documentation	370
5.40.1.1	gen_eval_operands	370
5.40.1.2	generate_body	371
5.40.1.3	generate_footer	371
5.40.1.4	generate_header	371
5.40.1.5	main	371
5.40.1.6	output_call	372
5.40.1.7	output_function	372
5.40.1.8	shift_fprintf	372
5.40.2	Variable Documentation	372
5.40.2.1	in_file	372
5.40.2.2	out_file	372
5.40.2.3	out_lines	372
5.40.2.4	write_to_reg	372
5.41	cpu/or32/i386_regs.h File Reference	373

5.41.1	Define Documentation	373
5.41.1.1	CPU_STATE_REG	373
5.41.1.2	NUM_T_REGS	373
5.41.1.3	T0_REG	373
5.41.1.4	T1_REG	373
5.41.1.5	T2_REG	373
5.42	cpu/or32/insnset.c File Reference	374
5.42.1	Function Documentation	377
5.42.1.1	INSTRUCTION	377
5.42.1.2	INSTRUCTION	377
5.42.1.3	INSTRUCTION	377
5.42.1.4	INSTRUCTION	377
5.42.1.5	INSTRUCTION	377
5.42.1.6	INSTRUCTION	377
5.42.1.7	INSTRUCTION	377
5.42.1.8	INSTRUCTION	377
5.42.1.9	INSTRUCTION	377
5.42.1.10	INSTRUCTION	377
5.42.1.11	INSTRUCTION	377
5.42.1.12	INSTRUCTION	377
5.42.1.13	INSTRUCTION	377
5.42.1.14	INSTRUCTION	377
5.42.1.15	INSTRUCTION	377
5.42.1.16	INSTRUCTION	377
5.42.1.17	INSTRUCTION	377
5.42.1.18	INSTRUCTION	377
5.42.1.19	INSTRUCTION	377
5.42.1.20	INSTRUCTION	377
5.42.1.21	INSTRUCTION	377
5.42.1.22	INSTRUCTION	377
5.42.1.23	INSTRUCTION	377
5.42.1.24	INSTRUCTION	377
5.42.1.25	INSTRUCTION	378
5.42.1.26	INSTRUCTION	378
5.42.1.27	INSTRUCTION	379
5.42.1.28	INSTRUCTION	380

5.42.1.29 INSTRUCTION	380
5.42.1.30 INSTRUCTION	380
5.42.1.31 INSTRUCTION	380
5.42.1.32 INSTRUCTION	380
5.42.1.33 INSTRUCTION	380
5.42.1.34 INSTRUCTION	380
5.42.1.35 INSTRUCTION	380
5.42.1.36 INSTRUCTION	380
5.42.1.37 INSTRUCTION	380
5.42.1.38 INSTRUCTION	380
5.42.1.39 INSTRUCTION	380
5.42.1.40 INSTRUCTION	380
5.42.1.41 INSTRUCTION	380
5.42.1.42 INSTRUCTION	380
5.42.1.43 INSTRUCTION	380
5.42.1.44 INSTRUCTION	380
5.42.1.45 INSTRUCTION	381
5.42.1.46 INSTRUCTION	381
5.42.1.47 INSTRUCTION	381
5.42.1.48 INSTRUCTION	381
5.42.1.49 INSTRUCTION	382
5.42.1.50 INSTRUCTION	382
5.42.1.51 INSTRUCTION	382
5.42.1.52 INSTRUCTION	382
5.42.1.53 INSTRUCTION	382
5.42.1.54 INSTRUCTION	382
5.42.1.55 INSTRUCTION	382
5.42.1.56 INSTRUCTION	383
5.42.1.57 INSTRUCTION	383
5.42.1.58 INSTRUCTION	383
5.42.1.59 INSTRUCTION	383
5.42.1.60 INSTRUCTION	383
5.42.1.61 INSTRUCTION	383
5.42.1.62 INSTRUCTION	383
5.42.1.63 INSTRUCTION	383
5.42.1.64 INSTRUCTION	384

5.42.1.65 INSTRUCTION	384
5.42.1.66 INSTRUCTION	384
5.42.1.67 INSTRUCTION	385
5.42.1.68 INSTRUCTION	385
5.42.1.69 INSTRUCTION	386
5.42.1.70 INSTRUCTION	386
5.42.1.71 INSTRUCTION	386
5.42.1.72 INSTRUCTION	386
5.43 cpu/or32/op.c File Reference	387
5.43.1 Define Documentation	394
5.43.1.1 __or_dynop	394
5.43.1.2 COMP	394
5.43.1.3 COMP	394
5.43.1.4 COMP	394
5.43.1.5 COMP	394
5.43.1.6 COMP	394
5.43.1.7 COMP	394
5.43.1.8 COMP	394
5.43.1.9 COMP	394
5.43.1.10 COMP	394
5.43.1.11 COMP	394
5.43.1.12 COMP_CAST	394
5.43.1.13 COMP_CAST	394
5.43.1.14 COMP_CAST	394
5.43.1.15 COMP_CAST	394
5.43.1.16 COMP_CAST	394
5.43.1.17 COMP_CAST	394
5.43.1.18 COMP_CAST	394
5.43.1.19 COMP_CAST	394
5.43.1.20 COMP_CAST	394
5.43.1.21 COMP_CAST	394
5.43.1.22 COMP_NAME	394
5.43.1.23 COMP_NAME	394
5.43.1.24 COMP_NAME	394
5.43.1.25 COMP_NAME	394
5.43.1.26 COMP_NAME	394

5.43.1.27 COMP_NAME	394
5.43.1.28 COMP_NAME	394
5.43.1.29 COMP_NAME	394
5.43.1.30 COMP_NAME	394
5.43.1.31 COMP_NAME	394
5.43.1.32 EXT_CAST	394
5.43.1.33 EXT_CAST	394
5.43.1.34 EXT_CAST	394
5.43.1.35 EXT_CAST	394
5.43.1.36 EXT_NAME	394
5.43.1.37 EXT_NAME	394
5.43.1.38 EXT_NAME	394
5.43.1.39 EXT_NAME	394
5.43.1.40 EXT_TYPE	394
5.43.1.41 EXT_TYPE	394
5.43.1.42 EXT_TYPE	394
5.43.1.43 EXT_TYPE	394
5.43.1.44 LS_OP_CAST	394
5.43.1.45 LS_OP_CAST	394
5.43.1.46 LS_OP_CAST	394
5.43.1.47 LS_OP_CAST	394
5.43.1.48 LS_OP_CAST	394
5.43.1.49 LS_OP_CAST	394
5.43.1.50 LS_OP_FUNC	394
5.43.1.51 LS_OP_FUNC	394
5.43.1.52 LS_OP_FUNC	394
5.43.1.53 LS_OP_FUNC	394
5.43.1.54 LS_OP_FUNC	394
5.43.1.55 LS_OP_FUNC	394
5.43.1.56 LS_OP_NAME	394
5.43.1.57 LS_OP_NAME	394
5.43.1.58 LS_OP_NAME	394
5.43.1.59 LS_OP_NAME	394
5.43.1.60 LS_OP_NAME	394
5.43.1.61 LS_OP_NAME	394
5.43.1.62 OP	394

5.43.1.63 OP	394
5.43.1.64 OP	394
5.43.1.65 OP	394
5.43.1.66 OP	394
5.43.1.67 OP	394
5.43.1.68 OP	394
5.43.1.69 OP	394
5.43.1.70 OP	394
5.43.1.71 OP	394
5.43.1.72 OP	394
5.43.1.73 OP	394
5.43.1.74 OP	394
5.43.1.75 OP	394
5.43.1.76 OP	394
5.43.1.77 OP_CAST	394
5.43.1.78 OP_CAST	394
5.43.1.79 OP_CAST	394
5.43.1.80 OP_CAST	394
5.43.1.81 OP_CAST	394
5.43.1.82 OP_CAST	394
5.43.1.83 OP_CAST	394
5.43.1.84 OP_CAST	394
5.43.1.85 OP_CAST	394
5.43.1.86 OP_CAST	394
5.43.1.87 OP_CAST	394
5.43.1.88 OP_CAST	394
5.43.1.89 OP_CAST	394
5.43.1.90 OP_EXTRA	394
5.43.1.91 OP_EXTRA	394
5.43.1.92 OP_EXTRA	394
5.43.1.93 OP_FILE	394
5.43.1.94 OP_FILE	394
5.43.1.95 OP_FILE	394
5.43.1.96 OP_FILE	394
5.43.1.97 OP_FILE	394
5.43.1.98 OP_FILE	394

5.43.1.99 OP_FILE	394
5.43.1.100OP_FILE	394
5.43.1.101OP_FILE	394
5.43.1.102OP_FILE	394
5.43.1.103OP_FILE	394
5.43.1.104OP_NAME	394
5.43.1.105OP_NAME	394
5.43.1.106OP_NAME	394
5.43.1.107OP_NAME	394
5.43.1.108OP_NAME	394
5.43.1.109OP_NAME	394
5.43.1.110OP_NAME	394
5.43.1.111OP_NAME	394
5.43.1.112OP_NAME	394
5.43.1.113OP_NAME	394
5.43.1.114OP_NAME	394
5.43.1.115OP_NAME	394
5.43.1.116OP_NAME	394
5.43.1.117OP_NAME	394
5.43.1.118OP_NAME	394
5.43.1.119OP_PARAM1	394
5.43.1.120OP_PARAM2	394
5.43.1.121OP_PARAM3	394
5.43.1.122S_FUNC	394
5.43.1.123S_FUNC	394
5.43.1.124S_FUNC	394
5.43.1.125S_OP_NAME	394
5.43.1.126S_OP_NAME	394
5.43.1.127S_OP_NAME	394
5.43.2 Function Documentation	394
5.43.2.1 asm	394
5.43.2.2 asm	394
5.43.2.3 asm	394
5.43.2.4 asm	394
5.43.2.5 do_sched_wrap	394
5.43.2.6 do_sched_wrap_delay	395

5.43.2.7	enter_dyn_code	395
5.43.2.8	op_add_pc	395
5.43.2.9	op_analysis	395
5.43.2.10	op_check_delay_slot	395
5.43.2.11	op_check_flag	395
5.43.2.12	op_check_flag_delay	396
5.43.2.13	op_check_not_flag	396
5.43.2.14	op_check_not_flag_delay	396
5.43.2.15	op_clear_delay_insn	396
5.43.2.16	op_clear_flag	396
5.43.2.17	op_clear_pc_delay	396
5.43.2.18	op_do_jump_delay	396
5.43.2.19	op_do_sched	396
5.43.2.20	op_do_sched_delay	396
5.43.2.21	op_illegal	398
5.43.2.22	op_illegal_delay	398
5.43.2.23	op_jump_imm	398
5.43.2.24	op_join_mem_cycles	398
5.43.2.25	op_macc	398
5.43.2.26	op_move_gpr10_pc_delay	398
5.43.2.27	op_move_gpr11_pc_delay	398
5.43.2.28	op_move_gpr12_pc_delay	398
5.43.2.29	op_move_gpr13_pc_delay	398
5.43.2.30	op_move_gpr14_pc_delay	398
5.43.2.31	op_move_gpr15_pc_delay	398
5.43.2.32	op_move_gpr16_pc_delay	398
5.43.2.33	op_move_gpr17_pc_delay	398
5.43.2.34	op_move_gpr18_pc_delay	398
5.43.2.35	op_move_gpr19_pc_delay	398
5.43.2.36	op_move_gpr1_pc_delay	398
5.43.2.37	op_move_gpr20_pc_delay	398
5.43.2.38	op_move_gpr21_pc_delay	398
5.43.2.39	op_move_gpr22_pc_delay	398
5.43.2.40	op_move_gpr23_pc_delay	398
5.43.2.41	op_move_gpr24_pc_delay	398
5.43.2.42	op_move_gpr25_pc_delay	398

5.43.2.43	op_move_gpr26_pc_delay	398
5.43.2.44	op_move_gpr27_pc_delay	398
5.43.2.45	op_move_gpr28_pc_delay	398
5.43.2.46	op_move_gpr29_pc_delay	398
5.43.2.47	op_move_gpr2_pc_delay	398
5.43.2.48	op_move_gpr30_pc_delay	398
5.43.2.49	op_move_gpr31_pc_delay	398
5.43.2.50	op_move_gpr3_pc_delay	398
5.43.2.51	op_move_gpr4_pc_delay	398
5.43.2.52	op_move_gpr5_pc_delay	398
5.43.2.53	op_move_gpr6_pc_delay	398
5.43.2.54	op_move_gpr7_pc_delay	398
5.43.2.55	op_move_gpr8_pc_delay	398
5.43.2.56	op_move_gpr9_pc_delay	398
5.43.2.57	op_nop_exit	398
5.43.2.58	op_nop_printf	399
5.43.2.59	op_nop_report	399
5.43.2.60	op_nop_report_imm	399
5.43.2.61	op_nop_reset	399
5.43.2.62	op_prep_rfe	400
5.43.2.63	op_prep_sys	400
5.43.2.64	op_prep_sys_delay	400
5.43.2.65	op_prep_trap	400
5.43.2.66	op_prep_trap_delay	400
5.43.2.67	op_set_delay_insn	401
5.43.2.68	op_set_flag	401
5.43.2.69	op_set_pc_delay_imm	401
5.43.2.70	op_set_pc_delay_pc	401
5.43.2.71	op_set_pc_pc_delay	401
5.43.2.72	op_store_insn_ea	401
5.43.2.73	op_store_link_addr_gpr	401
5.43.2.74	prep_except	401
5.43.2.75	save_t_bound	401
5.43.3	Variable Documentation	401
5.43.3.1	__op_param1	401
5.43.3.2	__op_param2	401

5.43.3.3	__op_param3	401
5.44	cpu/or32/op_1t.h File Reference	402
5.44.1	Define Documentation	402
5.44.1.1	OP_1T	402
5.44.1.2	T	402
5.45	cpu/or32/op_1t_op.h File Reference	403
5.45.1	Function Documentation	403
5.45.1.1	glue	403
5.45.1.2	glue	403
5.45.1.3	glue	403
5.45.1.4	glue	403
5.45.1.5	glue	403
5.45.1.6	glue	403
5.45.1.7	glue	403
5.46	cpu/or32/op_2t.h File Reference	404
5.46.1	Define Documentation	404
5.46.1.1	OP_2T	404
5.46.1.2	T	404
5.47	cpu/or32/op_2t_op.h File Reference	405
5.47.1	Function Documentation	405
5.47.1.1	glue	405
5.47.1.2	glue	405
5.47.1.3	glue	405
5.48	cpu/or32/op_3t.h File Reference	406
5.48.1	Define Documentation	406
5.48.1.1	OP_3T	406
5.48.1.2	T	406
5.49	cpu/or32/op_3t_op.h File Reference	407
5.49.1	Function Documentation	407
5.49.1.1	glue	407
5.50	cpu/or32/op_arith_op.h File Reference	408
5.51	cpu/or32/op_comp_op.h File Reference	409
5.52	cpu/or32/op_extend_op.h File Reference	410
5.52.1	Function Documentation	410
5.52.1.1	glue	410
5.53	cpu/or32/op_ff1_op.h File Reference	411

5.53.1	Function Documentation	411
5.53.1.1	glue	411
5.54	cpu/or32/op_i386.h File Reference	412
5.54.1	Define Documentation	412
5.54.1.1	FORCE_RET	412
5.54.1.2	OP_JUMP	412
5.54.1.3	SPEEDY_CALL	412
5.54.2	Function Documentation	412
5.54.2.1	asm	412
5.55	cpu/or32/op_lwhb_op.h File Reference	413
5.56	cpu/or32/op_mac_op.h File Reference	414
5.56.1	Function Documentation	414
5.56.1.1	glue	414
5.57	cpu/or32/op_mftspr_op.h File Reference	415
5.57.1	Function Documentation	415
5.57.1.1	op_mtspr_imm_clear	415
5.58	cpu/or32/op_support.c File Reference	416
5.58.1	Function Documentation	417
5.58.1.1	do_jump	417
5.58.1.2	op_support_analysis	417
5.58.1.3	op_support_nop_exit	417
5.58.1.4	op_support_nop_printf	417
5.58.1.5	op_support_nop_report	418
5.58.1.6	op_support_nop_report_imm	418
5.58.1.7	op_support_nop_reset	418
5.59	cpu/or32/op_support.h File Reference	419
5.59.1	Function Documentation	419
5.59.1.1	do_jump	419
5.59.1.2	op_support_analysis	419
5.59.1.3	op_support_nop_exit	420
5.59.1.4	op_support_nop_printf	420
5.59.1.5	op_support_nop_report	420
5.59.1.6	op_support_nop_report_imm	420
5.59.1.7	op_support_nop_reset	420
5.59.1.8	upd_reg_from_t	420
5.60	cpu/or32/op_swhb_op.h File Reference	421

5.60.1	Function Documentation	421
5.60.1.1	glue	421
5.61	cpu/or32/op_t_reg_mov_op.h File Reference	422
5.61.1	Function Documentation	425
5.61.1.1	glue	425
5.61.1.2	glue	425
5.61.1.3	glue	425
5.61.1.4	glue	425
5.61.1.5	glue	425
5.61.1.6	glue	425
5.61.1.7	glue	425
5.61.1.8	glue	425
5.61.1.9	glue	425
5.61.1.10	glue	425
5.61.1.11	glue	425
5.61.1.12	glue	425
5.61.1.13	glue	425
5.61.1.14	glue	425
5.61.1.15	glue	425
5.61.1.16	glue	425
5.61.1.17	glue	425
5.61.1.18	glue	425
5.61.1.19	glue	425
5.61.1.20	glue	425
5.61.1.21	glue	425
5.61.1.22	glue	425
5.61.1.23	glue	425
5.61.1.24	glue	425
5.61.1.25	glue	425
5.61.1.26	glue	425
5.61.1.27	glue	425
5.61.1.28	glue	425
5.61.1.29	glue	425
5.61.1.30	glue	425
5.61.1.31	glue	425
5.61.1.32	glue	425

5.61.1.33 glue	425
5.61.1.34 glue	425
5.61.1.35 glue	425
5.61.1.36 glue	425
5.61.1.37 glue	425
5.61.1.38 glue	425
5.61.1.39 glue	425
5.61.1.40 glue	425
5.61.1.41 glue	425
5.61.1.42 glue	425
5.61.1.43 glue	425
5.61.1.44 glue	425
5.61.1.45 glue	425
5.61.1.46 glue	425
5.61.1.47 glue	425
5.61.1.48 glue	425
5.61.1.49 glue	425
5.61.1.50 glue	425
5.61.1.51 glue	425
5.61.1.52 glue	425
5.61.1.53 glue	425
5.61.1.54 glue	425
5.61.1.55 glue	425
5.61.1.56 glue	425
5.61.1.57 glue	425
5.61.1.58 glue	425
5.61.1.59 glue	425
5.61.1.60 glue	425
5.61.1.61 glue	425
5.61.1.62 glue	425
5.62 cpu/or32/or32.c File Reference	426
5.62.1 Define Documentation	428
5.62.1.1 EF	428
5.62.1.2 EFI	428
5.62.1.3 EFN	428
5.62.1.4 MAX_AUTOMATA_SIZE	428

5.62.1.5	MAX_LEN	428
5.62.1.6	MAX_OP_TABLE_SIZE	428
5.62.1.7	MIN	428
5.62.2	Function Documentation	428
5.62.2.1	build_automata	428
5.62.2.2	cover_insn	428
5.62.2.3	destruct_automata	429
5.62.2.4	disassemble_index	429
5.62.2.5	disassemble_insn	429
5.62.2.6	extend_imm	429
5.62.2.7	insn_decode	430
5.62.2.8	insn_extract	430
5.62.2.9	insn_index	430
5.62.2.10	insn_len	430
5.62.2.11	insn_name	430
5.62.2.12	l_none	430
5.62.2.13	letter_range	430
5.62.2.14	letter_signed	430
5.62.2.15	num_ones	430
5.62.2.16	or32_debug	430
5.62.2.17	or32_extract	430
5.62.2.18	or32_print_immediate	430
5.62.2.19	or32_print_register	430
5.62.2.20	parse_params	431
5.62.3	Variable Documentation	431
5.62.3.1	automata	431
5.62.3.2	curpass	431
5.62.3.3	disassembled	431
5.62.3.4	disassembled_str	431
5.62.3.5	num_opcodes	431
5.62.3.6	nuncovered	431
5.62.3.7	op_data	431
5.62.3.8	op_start	431
5.62.3.9	or32_letters	431
5.62.3.10	or32_opcodes	432
5.62.3.11	range_cache	432

5.62.3.12	ti	432
5.63	cpu/or32/rec_i386.h File Reference	433
5.63.1	Function Documentation	433
5.63.1.1	get_sp	433
5.64	cpu/or32/sched_i386.h File Reference	434
5.64.1	Function Documentation	434
5.64.1.1	set_sched_cycle	434
5.65	cpu/or32/simpl32_defs.h File Reference	435
5.65.1	Function Documentation	435
5.65.1.1	PARAMS	435
5.65.1.2	PARAMS	435
5.66	cuc/adv.c File Reference	436
5.66.1	Function Documentation	437
5.66.1.1	calc_max	437
5.66.1.2	detect_max_values	437
5.66.1.3	insert_conditional_facts	437
5.66.1.4	mark_successors	437
5.66.1.5	mask	437
5.66.1.6	max_op	437
5.67	cuc/bb.c File Reference	438
5.67.1	Function Documentation	439
5.67.1.1	build_bb	439
5.67.1.2	count_bb_seq	439
5.67.1.3	cpy_bb	439
5.67.1.4	cuc_check	439
5.67.1.5	detect_bb	440
5.67.1.6	dup_func	440
5.67.1.7	expand_bb	440
5.67.1.8	free_func	440
5.67.1.9	generate_bb_seq	440
5.67.1.10	join_bb	441
5.67.1.11	optimize_bb	441
5.67.1.12	preunroll_loop	441
5.67.1.13	print_bb_num	442
5.67.1.14	print_cuc_bb	442
5.67.1.15	recalc_last_used_reg	442

5.67.1.16	reg_dep	442
5.67.1.17	reg_dep_rec	442
5.67.1.18	relocate_bb	442
5.67.1.19	remove_dead_bb	442
5.67.1.20	roll_loop	442
5.67.1.21	simplify_bb	443
5.68	cuc/cuc.c File Reference	444
5.68.1	Function Documentation	447
5.68.1.1	analyse_function	447
5.68.1.2	calc_cycles	448
5.68.1.3	calc_size	448
5.68.1.4	cuc_calling_conv	448
5.68.1.5	cuc_enable_bursts	448
5.68.1.6	cuc_memory_order	448
5.68.1.7	cuc_no_multicycle	449
5.68.1.8	cuc_optimize	449
5.68.1.9	cuc_timings_fn	450
5.68.1.10	extract_function	450
5.68.1.11	format_func_options	450
5.68.1.12	gen_option	452
5.68.1.13	generate_function	452
5.68.1.14	main_cuc	454
5.68.1.15	options_cmd	455
5.68.1.16	preunroll_bb	457
5.68.1.17	print_option	458
5.68.1.18	reg_cuc_sec	458
5.68.1.19	set_func_deps	458
5.68.1.20	tim_comp	458
5.68.2	Variable Documentation	458
5.68.2.1	caller_saved	458
5.68.2.2	cuc_debug	459
5.68.2.3	flog	459
5.68.2.4	func	459
5.68.2.5	func_v	459
5.68.2.6	option_char	459
5.69	cuc/cuc.h File Reference	460

5.69.1	Define Documentation	465
5.69.1.1	BB_DEAD	465
5.69.1.2	BB_INLOOP	465
5.69.1.3	BB_OPTIONAL	465
5.69.1.4	BBID_END	465
5.69.1.5	BBID_START	465
5.69.1.6	CUC_MAX_STACK	465
5.69.1.7	CUC_WIDTH_ITERATIONS	465
5.69.1.8	cucdebug	465
5.69.1.9	FLAG_REG	465
5.69.1.10	INSN	465
5.69.1.11	IT_BBEND	465
5.69.1.12	IT_BBSTART	465
5.69.1.13	IT_BRANCH	465
5.69.1.14	IT_COND	465
5.69.1.15	IT_CUT	465
5.69.1.16	IT_FLAG1	465
5.69.1.17	IT_FLAG2	465
5.69.1.18	IT_INDELAY	465
5.69.1.19	IT_LATCHED	465
5.69.1.20	IT_MEMADD	465
5.69.1.21	IT_MEMORY	465
5.69.1.22	IT_OUTPUT	465
5.69.1.23	IT_SIGNED	465
5.69.1.24	IT_UNUSED	465
5.69.1.25	IT_VOLATILE	465
5.69.1.26	log	465
5.69.1.27	LRBB_REG	465
5.69.1.28	MAX	465
5.69.1.29	MAX_BB	465
5.69.1.30	MAX_INSNS	465
5.69.1.31	MAX_PREROLL	465
5.69.1.32	MAX_REGS	465
5.69.1.33	MAX_UNROLL	465
5.69.1.34	MIN	465
5.69.1.35	MO_EXACT	465

5.69.1.36	MO_NONE	465
5.69.1.37	MO_STRONG	465
5.69.1.38	MO_WEAK	465
5.69.1.39	MT_BURST	465
5.69.1.40	MT_BURSTE	465
5.69.1.41	MT_CALL	465
5.69.1.42	MT_LOAD	465
5.69.1.43	MT_SIGNED	465
5.69.1.44	MT_STORE	465
5.69.1.45	MT_WIDTH	465
5.69.1.46	OPT_BB	465
5.69.1.47	OPT_CONST	465
5.69.1.48	OPT_DEST	465
5.69.1.49	OPT_JUMP	465
5.69.1.50	OPT_LRBB	465
5.69.1.51	OPT_NONE	465
5.69.1.52	OPT_REF	465
5.69.1.53	OPT_REGISTER	465
5.69.1.54	REF	465
5.69.1.55	REF_BB	465
5.69.1.56	REF_I	465
5.69.2	Typedef Documentation	465
5.69.2.1	cuc_func	465
5.69.2.2	cuc_shared_list	465
5.69.2.3	dep_list	465
5.69.3	Function Documentation	465
5.69.3.1	add_data_dep	465
5.69.3.2	add_dep	466
5.69.3.3	add_latches	466
5.69.3.4	add_memory_dep	466
5.69.3.5	analyse_timings	466
5.69.3.6	build_bb	466
5.69.3.7	clean_deps	467
5.69.3.8	cse	467
5.69.3.9	csm	467
5.69.3.10	csm_gen	467

5.69.3.11	cuc_check	467
5.69.3.12	cuc_load	468
5.69.3.13	detect_bb	468
5.69.3.14	detect_max_values	468
5.69.3.15	dispose_list	469
5.69.3.16	dup_func	469
5.69.3.17	expand_bb	469
5.69.3.18	free_func	469
5.69.3.19	generate_bb_seq	469
5.69.3.20	insert_conditional_facts	470
5.69.3.21	insert_insns	470
5.69.3.22	main_cuc	472
5.69.3.23	mark_cut	473
5.69.3.24	negate_conditional	473
5.69.3.25	optimize_bb	473
5.69.3.26	optimize_cmovs	473
5.69.3.27	optimize_tree	474
5.69.3.28	preunroll_loop	474
5.69.3.29	print_bb_num	474
5.69.3.30	print_cuc_bb	474
5.69.3.31	print_cuc_insns	475
5.69.3.32	print_insns	475
5.69.3.33	recalc_cnts	475
5.69.3.34	reg_cuc_sec	475
5.69.3.35	reg_dep	476
5.69.3.36	remove_dead	476
5.69.3.37	remove_dead_bb	476
5.69.3.38	remove_nops	476
5.69.3.39	remove_trivial_regs	476
5.69.3.40	schedule_memory	477
5.69.3.41	set_io	477
5.69.4	Variable Documentation	477
5.69.4.1	caller_saved	477
5.69.4.2	cuc_debug	477
5.69.4.3	flog	477
5.69.4.4	insn	477

5.69.4.5	num_insn	477
5.69.4.6	reloc	477
5.70	cuc/insn.c File Reference	478
5.70.1	Function Documentation	479
5.70.1.1	add_data_dep	479
5.70.1.2	add_dep	479
5.70.1.3	add_latches	479
5.70.1.4	apply_edge_condition	480
5.70.1.5	change_insn_type	480
5.70.1.6	cmov_needed	480
5.70.1.7	count_cmovs	480
5.70.1.8	cse	480
5.70.1.9	csm	480
5.70.1.10	csm_gen	480
5.70.1.11	cuc_insn_name	481
5.70.1.12	dispose_list	481
5.70.1.13	insert_insns	481
5.70.1.14	insn_uses	481
5.70.1.15	optimize_cmov_more	481
5.70.1.16	optimize_cmovs	481
5.70.1.17	optimize_tree	482
5.70.1.18	print_insns	482
5.70.1.19	print_shared	482
5.70.1.20	remove_dead	482
5.70.1.21	remove_nops	483
5.70.1.22	remove_trivial_regs	483
5.70.1.23	search_csm	483
5.70.1.24	set_io	483
5.70.1.25	unmark_tree	483
5.70.2	Variable Documentation	483
5.70.2.1	iteration	483
5.70.2.2	known	483
5.70.2.3	main_list	483
5.70.2.4	tmp_op	483
5.70.2.5	tmp_opt	483
5.71	cuc/insn.h File Reference	484

5.71.1	Define Documentation	486
5.71.1.1	II_ADD	486
5.71.1.2	II_AND	486
5.71.1.3	II_BF	486
5.71.1.4	II_CALL	486
5.71.1.5	II_CMOV	486
5.71.1.6	II_IS_LOAD	486
5.71.1.7	II_IS_STORE	486
5.71.1.8	II_LAST	486
5.71.1.9	II_LB	486
5.71.1.10	II_LH	486
5.71.1.11	II_LRBB	486
5.71.1.12	II_LW	486
5.71.1.13	II_MASK	486
5.71.1.14	II_MEM	486
5.71.1.15	II_MEM_WIDTH	486
5.71.1.16	II_MUL	488
5.71.1.17	II_NOP	488
5.71.1.18	II_OR	488
5.71.1.19	II_REG	488
5.71.1.20	II_SB	488
5.71.1.21	II_SFEQ	488
5.71.1.22	II_SFGE	488
5.71.1.23	II_SFGT	488
5.71.1.24	II_SFLE	488
5.71.1.25	II_SFLT	488
5.71.1.26	II_SFNE	488
5.71.1.27	II_SH	488
5.71.1.28	II_SIGNED	488
5.71.1.29	II_SLL	488
5.71.1.30	II_SRA	488
5.71.1.31	II_SRL	488
5.71.1.32	II_SUB	488
5.71.1.33	II_SW	488
5.71.1.34	II_XOR	488
5.71.2	Function Documentation	488

5.71.2.1	change_insn_type	488
5.71.2.2	cuc_insn_name	488
5.71.2.3	ii_size	488
5.71.2.4	insn_size	488
5.71.2.5	insn_time	488
5.71.2.6	load_timing_table	488
5.71.2.7	print_shared	488
5.71.3	Variable Documentation	489
5.71.3.1	known	489
5.72	cuc/load.c File Reference	490
5.72.1	Function Documentation	491
5.72.1.1	build_insn	491
5.72.1.2	cuc_load	492
5.72.1.3	detect_locals	492
5.72.1.4	expand_branch	492
5.72.1.5	expand_calls	492
5.72.1.6	expand_memory	493
5.72.1.7	expand_signed	493
5.72.1.8	negate_conditional	493
5.72.1.9	print_cuc_insns	493
5.72.1.10	remove_dslots	493
5.72.1.11	xchg_insn	494
5.72.2	Variable Documentation	494
5.72.2.1	conv	494
5.72.2.2	insn	494
5.72.2.3	num_insn	494
5.72.2.4	reloc	494
5.73	cuc/memory.c File Reference	495
5.73.1	Function Documentation	496
5.73.1.1	add_memory_dep	496
5.73.1.2	check_memory_conflict	496
5.73.1.3	clean_deps	496
5.73.1.4	join_transfers	496
5.73.1.5	mem_ordering_cmp	496
5.73.1.6	same_transfers	496
5.73.1.7	schedule_memory	497

5.74 peripheral/memory.c File Reference	498
5.74.1 Function Documentation	499
5.74.1.1 mem_reset	499
5.74.1.2 memory_baseaddr	499
5.74.1.3 memory_ce	499
5.74.1.4 memory_delayr	499
5.74.1.5 memory_delayw	499
5.74.1.6 memory_log	499
5.74.1.7 memory_mc	499
5.74.1.8 memory_name	499
5.74.1.9 memory_pattern	499
5.74.1.10 memory_random_seed	500
5.74.1.11 memory_sec_end	500
5.74.1.12 memory_sec_start	500
5.74.1.13 memory_size	500
5.74.1.14 memory_type	500
5.74.1.15 reg_memory_sec	501
5.74.1.16 simmem_read16	502
5.74.1.17 simmem_read32	502
5.74.1.18 simmem_read8	502
5.74.1.19 simmem_read_zero16	502
5.74.1.20 simmem_read_zero32	502
5.74.1.21 simmem_read_zero8	502
5.74.1.22 simmem_write16	502
5.74.1.23 simmem_write32	502
5.74.1.24 simmem_write8	502
5.74.1.25 simmem_write_null16	502
5.74.1.26 simmem_write_null32	502
5.74.1.27 simmem_write_null8	502
5.75 cuc/timings.c File Reference	503
5.75.1 Function Documentation	504
5.75.1.1 analyse_timings	504
5.75.1.2 bb_size	504
5.75.1.3 cut_tree	504
5.75.1.4 ii_size	505
5.75.1.5 insn_size	505

5.75.1.6	<code>insn_time</code>	505
5.75.1.7	<code>load_timing_table</code>	505
5.75.1.8	<code>mark_cut</code>	505
5.75.1.9	<code>max_delay</code>	505
5.75.1.10	<code>memory_delay</code>	505
5.75.1.11	<code>new_bb_cycles</code>	505
5.75.1.12	<code>recalc_cnts</code>	505
5.75.2	Variable Documentation	505
5.75.2.1	<code>max_bb_delay</code>	505
5.75.2.2	<code>timing_table</code>	505
5.76	<code>cuc/verilog.c</code> File Reference	506
5.76.1	Define Documentation	507
5.76.1.1	<code>GEN</code>	507
5.76.2	Function Documentation	507
5.76.2.1	<code>branch_index</code>	507
5.76.2.2	<code>find_lsc_index</code>	507
5.76.2.3	<code>func_index</code>	507
5.76.2.4	<code>generate_main</code>	507
5.76.2.5	<code>output_verilog</code>	507
5.76.2.6	<code>print_deps</code>	508
5.76.2.7	<code>print_insn_v</code>	508
5.76.2.8	<code>print_op_v</code>	508
5.76.2.9	<code>print_turn_off_dep</code>	508
5.77	<code>cuc/verilog.h</code> File Reference	509
5.77.1	Function Documentation	509
5.77.1.1	<code>generate_main</code>	509
5.77.1.2	<code>output_verilog</code>	509
5.78	<code>debug/debug-unit.c</code> File Reference	510
5.78.1	Define Documentation	511
5.78.1.1	<code>RISCOP_RESET</code>	511
5.78.1.2	<code>RISCOP_STALL</code>	511
5.78.2	Enumeration Type Documentation	512
5.78.2.1	<code>development_interface_address_space</code>	512
5.78.3	Function Documentation	512
5.78.3.1	<code>calculate_watchpoints</code>	512
5.78.3.2	<code>check_debug_unit</code>	512

5.78.3.3	debug_enabled	513
5.78.3.4	debug_gdb_enabled	513
5.78.3.5	debug_get_mem	513
5.78.3.6	debug_get_register	514
5.78.3.7	debug_ignore_exception	514
5.78.3.8	debug_server_port	515
5.78.3.9	debug_set_chain	515
5.78.3.10	debug_set_mem	515
5.78.3.11	debug_set_register	515
5.78.3.12	debug_vapi_id	516
5.78.3.13	DECLARE_DEBUG_CHANNEL	516
5.78.3.14	du_reset	516
5.78.3.15	get_devint_reg	517
5.78.3.16	get_devint_reg	517
5.78.3.17	reg_debug_sec	517
5.78.3.18	set_devint_reg	517
5.78.3.19	set_devint_reg	518
5.78.3.20	set_stall_state	518
5.78.4	Variable Documentation	518
5.78.4.1	current_scan_chain	518
5.78.4.2	development	518
5.78.4.3	in_reset	518
5.79	debug/debug-unit.h File Reference	519
5.79.1	Enumeration Type Documentation	519
5.79.1.1	debug_scan_chain_ids	519
5.79.1.2	debug_unit_action	520
5.79.2	Function Documentation	520
5.79.2.1	check_debug_unit	520
5.79.2.2	debug_get_register	520
5.79.2.3	debug_ignore_exception	521
5.79.2.4	debug_set_chain	521
5.79.2.5	debug_set_register	522
5.79.2.6	du_reset	522
5.79.2.7	reg_debug_sec	522
5.79.2.8	set_stall_state	523
5.80	debug/gdb.h File Reference	524

5.80.1	Enumeration Type Documentation	524
5.80.1.1	or1k_jtag_errors	524
5.80.1.2	or1k_jtag_proxy_protocol_commands	525
5.81	debug/gdbcomm.c File Reference	526
5.81.1	Function Documentation	527
5.81.1.1	block_jtag	527
5.81.1.2	gdb_read	527
5.81.1.3	gdb_request	527
5.81.1.4	gdb_write	528
5.81.1.5	gdbcomm_init	528
5.81.1.6	get_server_socket	528
5.81.1.7	handle_server_socket	528
5.81.1.8	jtag_request	529
5.81.1.9	protocol_clean	529
5.81.2	Variable Documentation	529
5.81.2.1	gdb_fd	529
5.81.2.2	server_fd	529
5.81.2.3	server_ip	529
5.81.2.4	server_port	529
5.81.2.5	tcp_level	529
5.82	debug/gdbcomm.h File Reference	530
5.82.1	Enumeration Type Documentation	530
5.82.1.1	boolean	530
5.82.2	Function Documentation	530
5.82.2.1	block_jtag	530
5.82.2.2	gdbcomm_init	530
5.82.2.3	handle_server_socket	531
5.83	libtoplevel.c File Reference	532
5.83.1	Function Documentation	533
5.83.1.1	internal_or1ksim_time	533
5.83.1.2	or1ksim_clock_rate	533
5.83.1.3	or1ksim_get_time_period	533
5.83.1.4	or1ksim_init	533
5.83.1.5	or1ksim_interrupt	536
5.83.1.6	or1ksim_is_le	536
5.83.1.7	or1ksim_reset_duration	536

5.83.1.8	or1ksim_run	536
5.83.1.9	or1ksim_set_time_point	537
5.84	mainpage File Reference	538
5.84.1	Detailed Description	538
5.85	mmu/dmmu.c File Reference	539
5.85.1	Function Documentation	540
5.85.1.1	DEFAULT_DEBUG_CHANNEL	540
5.85.1.2	dmmu_enabled	540
5.85.1.3	dmmu_end_sec	540
5.85.1.4	dmmu_entsize	540
5.85.1.5	dmmu_find_tlbmr	541
5.85.1.6	dmmu_hitdelay	541
5.85.1.7	dmmu_missdelay	541
5.85.1.8	dmmu_nsets	541
5.85.1.9	dmmu_nways	541
5.85.1.10	dmmu_pagesize	541
5.85.1.11	dmmu_start_sec	542
5.85.1.12	dmmu_translate	542
5.85.1.13	dmmu_ustates	543
5.85.1.14	dtlb_status	543
5.85.1.15	peek_into_dtlb	543
5.85.1.16	reg_dmmu_sec	544
5.85.2	Variable Documentation	544
5.85.2.1	dmmu_state	544
5.86	mmu/dmmu.h File Reference	545
5.86.1	Define Documentation	546
5.86.1.1	DADDR_PAGE	546
5.86.2	Function Documentation	546
5.86.2.1	dmmu_simulate_tlb	546
5.86.2.2	dmmu_translate	546
5.86.2.3	peek_into_dtlb	546
5.86.2.4	reg_dmmu_sec	547
5.86.3	Variable Documentation	547
5.86.3.1	dmmu_state	547
5.87	mmu/immu.c File Reference	548
5.87.1	Function Documentation	549

5.87.1.1	DEFAULT_DEBUG_CHANNEL	549
5.87.1.2	immu_enabled	549
5.87.1.3	immu_end_sec	549
5.87.1.4	immu_entrysize	549
5.87.1.5	immu_find_tlbmr	550
5.87.1.6	immu_hitdelay	550
5.87.1.7	immu_missdelay	550
5.87.1.8	immu_nsets	550
5.87.1.9	immu_nways	550
5.87.1.10	immu_pagesize	550
5.87.1.11	immu_start_sec	551
5.87.1.12	immu_translate	551
5.87.1.13	immu_ustates	551
5.87.1.14	itlb_status	552
5.87.1.15	peek_into_itlb	552
5.87.1.16	reg_immu_sec	552
5.87.2	Variable Documentation	552
5.87.2.1	immu_state	552
5.88	mmu/immu.h File Reference	553
5.88.1	Define Documentation	554
5.88.1.1	IADDR_PAGE	554
5.88.2	Function Documentation	554
5.88.2.1	immu_simulate_tlb	554
5.88.2.2	immu_translate	554
5.88.2.3	peek_into_itlb	554
5.88.2.4	reg_immu_sec	555
5.88.3	Variable Documentation	555
5.88.3.1	immu_state	555
5.89	mprofiler.c File Reference	556
5.89.1	Define Documentation	557
5.89.1.1	BUF_SIZE	557
5.89.1.2	HASH_FUNC	557
5.89.1.3	HASH_SIZE	557
5.89.1.4	MODE_ACCESS	557
5.89.1.5	MODE_DETAIL	557
5.89.1.6	MODE_PRETTY	557

5.89.1.7	MODE_WIDTH	557
5.89.2	Function Documentation	557
5.89.2.1	hash_add	557
5.89.2.2	hash_get	557
5.89.2.3	init	557
5.89.2.4	main_mprofiler	557
5.89.2.5	nbits	558
5.89.2.6	printout	558
5.89.2.7	read_file	558
5.89.3	Variable Documentation	558
5.89.3.1	end_addr	558
5.89.3.2	fprof	558
5.89.3.3	group_bits	558
5.89.3.4	hash	558
5.89.3.5	start_addr	559
5.90	mprofiler.h File Reference	560
5.90.1	Function Documentation	560
5.90.1.1	main_mprofiler	560
5.91	or1ksim.h File Reference	561
5.91.1	Enumeration Type Documentation	561
5.91.1.1	or1ksim_rc	561
5.91.2	Function Documentation	561
5.91.2.1	or1ksim_clock_rate	561
5.91.2.2	or1ksim_get_time_period	562
5.91.2.3	or1ksim_init	562
5.91.2.4	or1ksim_interrupt	562
5.91.2.5	or1ksim_is_le	562
5.91.2.6	or1ksim_reset_duration	562
5.91.2.7	or1ksim_run	563
5.91.2.8	or1ksim_set_time_point	563
5.92	peripheral/16450.c File Reference	564
5.92.1	Define Documentation	567
5.92.1.1	MAX_SKEW	567
5.92.1.2	MIN	567
5.92.1.3	UART_ADDR_SPACE	567
5.92.1.4	UART_BREAK_COUNT	567

5.92.1.5	UART_CHAR_TIMEOUT	567
5.92.1.6	UART_CLOCK_DIVIDER	567
5.92.1.7	UART_DLH	567
5.92.1.8	UART_DLL	567
5.92.1.9	UART_FCR	567
5.92.1.10	UART_FCR_FIE	567
5.92.1.11	UART_FCR_RRXFI	567
5.92.1.12	UART_FCR_RTXFI	567
5.92.1.13	UART_FGETC_SLOWDOWN	567
5.92.1.14	UART_FIFO_TRIGGER	568
5.92.1.15	UART_IER	570
5.92.1.16	UART_IER_MSI	570
5.92.1.17	UART_IER_RDI	570
5.92.1.18	UART_IER_RLSI	570
5.92.1.19	UART_IER_THRI	570
5.92.1.20	UART_IIR	570
5.92.1.21	UART_IIR_CTI	570
5.92.1.22	UART_IIR_ID	570
5.92.1.23	UART_IIR_MSI	570
5.92.1.24	UART_IIR_NO_INT	570
5.92.1.25	UART_IIR_RDI	570
5.92.1.26	UART_IIR_RLSI	570
5.92.1.27	UART_IIR_THRI	570
5.92.1.28	UART_LCR	570
5.92.1.29	UART_LCR_DLAB	570
5.92.1.30	UART_LCR_EPAR	570
5.92.1.31	UART_LCR_PARITY	570
5.92.1.32	UART_LCR_RESET	570
5.92.1.33	UART_LCR_SBC	570
5.92.1.34	UART_LCR_SPAR	570
5.92.1.35	UART_LCR_STOP	570
5.92.1.36	UART_LCR_WLEN5	570
5.92.1.37	UART_LCR_WLEN6	570
5.92.1.38	UART_LCR_WLEN7	570
5.92.1.39	UART_LCR_WLEN8	570
5.92.1.40	UART_LSR	570

5.92.1.41	UART_LSR_BREAK	570
5.92.1.42	UART_LSR_FRAME	570
5.92.1.43	UART_LSR_OVRRUN	570
5.92.1.44	UART_LSR_PARITY	570
5.92.1.45	UART_LSR_RDRDY	570
5.92.1.46	UART_LSR_RXERR	570
5.92.1.47	UART_LSR_TXBUFE	570
5.92.1.48	UART_LSR_TXSERE	570
5.92.1.49	UART_MAX_FIFO_LEN	570
5.92.1.50	UART_MCR	571
5.92.1.51	UART_MCR_AUX1	571
5.92.1.52	UART_MCR_AUX2	571
5.92.1.53	UART_MCR_DTR	571
5.92.1.54	UART_MCR_LOOP	571
5.92.1.55	UART_MCR_RTS	571
5.92.1.56	UART_MSR	571
5.92.1.57	UART_MSR_CTS	571
5.92.1.58	UART_MSR_DCD	571
5.92.1.59	UART_MSR_DCTS	571
5.92.1.60	UART_MSR_DDCD	571
5.92.1.61	UART_MSR_DDSR	571
5.92.1.62	UART_MSR_DSR	571
5.92.1.63	UART_MSR_RI	571
5.92.1.64	UART_MSR_TERI	571
5.92.1.65	UART_RXBUF	571
5.92.1.66	UART_SCR	571
5.92.1.67	UART_TXBUF	571
5.92.1.68	UART_VALID_FCR	571
5.92.1.69	UART_VALID_IER	571
5.92.1.70	UART_VALID_IIR	571
5.92.1.71	UART_VALID_LCR	571
5.92.1.72	UART_VALID_LSR	571
5.92.1.73	UART_VALID_MCR	571
5.92.1.74	UART_VALID_MSR	571
5.92.1.75	UART_VAPI_BUF_LEN	571
5.92.2	Function Documentation	573

5.92.2.1	char_clks	573
5.92.2.2	DEFAULT_DEBUG_CHANNEL	573
5.92.2.3	reg_uart_sec	573
5.92.2.4	send_char	574
5.92.2.5	uart_16550	574
5.92.2.6	uart_add_char	574
5.92.2.7	uart_baseaddr	574
5.92.2.8	uart_channel	574
5.92.2.9	uart_char_clock	575
5.92.2.10	uart_check_char	575
5.92.2.11	uart_check_rdi	575
5.92.2.12	uart_check_rlsi	576
5.92.2.13	uart_clear_int	576
5.92.2.14	uart_enabled	577
5.92.2.15	uart_int_cti	577
5.92.2.16	uart_int_msi	577
5.92.2.17	uart_int_rdi	578
5.92.2.18	uart_int_rlsi	578
5.92.2.19	uart_int_thri	579
5.92.2.20	uart_irq	579
5.92.2.21	uart_jitter	579
5.92.2.22	uart_loopback	579
5.92.2.23	uart_newway	580
5.92.2.24	uart_next_int	580
5.92.2.25	uart_read_byte	580
5.92.2.26	uart_recv_break	581
5.92.2.27	uart_recv_break_start	581
5.92.2.28	uart_recv_break_stop	581
5.92.2.29	uart_recv_char	582
5.92.2.30	uart_reset	582
5.92.2.31	uart_sched_recv_check	582
5.92.2.32	uart_sec_end	583
5.92.2.33	uart_sec_start	583
5.92.2.34	uart_send_break	584
5.92.2.35	uart_status	584
5.92.2.36	uart_tx_send	584

5.92.2.37	uart_vapi_cmd	585
5.92.2.38	uart_vapi_id	585
5.92.2.39	uart_vapi_read	585
5.92.2.40	uart_write_byte	585
5.93	peripheral/16450.h File Reference	586
5.93.1	Function Documentation	588
5.93.1.1	reg_uart_sec	588
5.93.1.2	uart_reset	589
5.93.1.3	uart_status	589
5.94	peripheral/atacmd.h File Reference	590
5.94.1	Define Documentation	593
5.94.1.1	CFA_DISABLE_8BIT_PIO_TRANSFER_MODE	593
5.94.1.2	CFA_DISABLE_POWER_MODE1	593
5.94.1.3	CFA_ENABLE_8BIT_PIO_TRANSFER_MODE	593
5.94.1.4	CFA_ENABLE_POWER_MODE1	593
5.94.1.5	CFA_ERASE_SECTORS	593
5.94.1.6	CFA_REQUEST_EXTENDED_ERROR_CODE	593
5.94.1.7	CFA_TRANSLATE_SECTOR	593
5.94.1.8	CFA_WRITE_MULTIPLE_WITHOUT_ERASE	593
5.94.1.9	CFA_WRITE_SECTORS_WITHOUT_ERASE	593
5.94.1.10	CHECK_POWER_MODE	593
5.94.1.11	DEVICE_RESET	593
5.94.1.12	DISABLE_ADVANCED_POWER_MANAGEMENT	593
5.94.1.13	DISABLE_MEDIA_STATUS_NOTIFICATION	593
5.94.1.14	DISABLE_POWERUP_IN_STANDBY_FEATURE_SET	593
5.94.1.15	DISABLE_READ_LOOKAHEAD	593
5.94.1.16	DISABLE_RELEASE_INTERRUPT	593
5.94.1.17	DISABLE_REVERTING_TO_POWERON_DEFAULTS	593
5.94.1.18	DISABLE_SERVICE_INTERRUPT	593
5.94.1.19	DISABLE_WRITE_CACHE	593
5.94.1.20	DOWNLOAD_MICROCODE	593
5.94.1.21	ENABLE_ADVANCED_POWER_MANAGEMENT	593
5.94.1.22	ENABLE_MEDIA_STATUS_NOTIFICATION	593
5.94.1.23	ENABLE_POWERUP_IN_STANDBY_FEATURE_SET	593
5.94.1.24	ENABLE_READ_LOOKAHEAD_FEATURE	593
5.94.1.25	ENABLE_RELEASE_INTERRUPT	593

5.94.1.26	ENABLE_REVERTING_TO_POWERON_DEFAULTS	593
5.94.1.27	ENABLE_SERVICE_INTERRUPT	593
5.94.1.28	ENABLE_WRITE_CACHE	593
5.94.1.29	EXECUTE_DEVICE_DIAGNOSTICS	593
5.94.1.30	FLUSH_CACHE	593
5.94.1.31	GET_MEDIA_STATUS	593
5.94.1.32	IDENTIFY_DEVICE	593
5.94.1.33	IDENTIFY_PACKET_DEVICE	593
5.94.1.34	IDLE	593
5.94.1.35	IDLE_IMMEDIATE	593
5.94.1.36	INITIALIZE_DEVICE_PARAMETERS	593
5.94.1.37	MEDIA_EJECT	593
5.94.1.38	MEDIA_LOCK	593
5.94.1.39	MEDIA_UNLOCK	593
5.94.1.40	NOP	593
5.94.1.41	PACKET	593
5.94.1.42	POWERUP_IN_STANDBY_FEATURE_SET_SPINUP	593
5.94.1.43	READ_BUFFER	593
5.94.1.44	READ_DMA	593
5.94.1.45	READ_DMA_QUEUED	593
5.94.1.46	READ_MULTIPLE	593
5.94.1.47	READ_NATIVE_MAX_ADDRESS	593
5.94.1.48	READ_SECTOR	593
5.94.1.49	READ_SECTORS	593
5.94.1.50	READ_VERIFY_SECTOR	593
5.94.1.51	READ_VERIFY_SECTORS	593
5.94.1.52	SECURITY_DISABLE_PASSWORD	593
5.94.1.53	SECURITY_ERASE_PREPARE	593
5.94.1.54	SECURITY_ERASE_UNIT	593
5.94.1.55	SECURITY_FREEZE_LOCK	593
5.94.1.56	SECURITY_SET_PASSWORD	593
5.94.1.57	SECURITY_UNLOCK	593
5.94.1.58	SEEK	593
5.94.1.59	SERVICE	593
5.94.1.60	SET_FEATURES	593
5.94.1.61	SET_MAX	593

5.94.1.62	SET_MAX_ADDRESS	593
5.94.1.63	SET_MAX_FREEZE_LOCK	593
5.94.1.64	SET_MAX_LOCK	593
5.94.1.65	SET_MAX_SET_PASSWORD	593
5.94.1.66	SET_MAX_UNLOCK	593
5.94.1.67	SET_MULTIPLE_MODE	593
5.94.1.68	SET_TRANSFER_MODE_SECTOR_COUNT_REG	593
5.94.1.69	SLEEP	593
5.94.1.70	SMART	593
5.94.1.71	SMART_ATTRIBUTE_AUTOSAVE	593
5.94.1.72	SMART_DISABLE_OPERATIONS	593
5.94.1.73	SMART_ENABLE_OPERATIONS	593
5.94.1.74	SMART_EXECUTE_OFFLINE_IMMEDIATE	593
5.94.1.75	SMART_READ_DATA	593
5.94.1.76	SMART_READ_LOG	593
5.94.1.77	SMART_RETURN_STATUS	593
5.94.1.78	SMART_SAVE_ATTRIBUTE_VALUES	593
5.94.1.79	SMART_WRITE_LOG	593
5.94.1.80	STANDBY	593
5.94.1.81	STANDBY_IMMEDIATE	593
5.94.1.82	WRITE_BUFFER	593
5.94.1.83	WRITE_DMA	593
5.94.1.84	WRITE_DMA_QUEUED	593
5.94.1.85	WRITE_MULTIPLE	593
5.94.1.86	WRITE_SECTOR	593
5.94.1.87	WRITE_SECTORS	593
5.95	peripheral/atadevice.c File Reference	594
5.95.1	Function Documentation	595
5.95.1.1	ata_device_do_command_register	595
5.95.1.2	ata_device_do_control_register	595
5.95.1.3	ata_device_hw_reset	595
5.95.1.4	ata_device_init	595
5.95.1.5	ata_device_write	596
5.95.1.6	ata_devices_hw_reset	596
5.95.1.7	ata_devices_init	596
5.95.1.8	ata_devices_read	596

5.95.1.9	ata_devices_write	597
5.95.1.10	ata_pretty_status	597
5.95.1.11	DEFAULT_DEBUG_CHANNEL	597
5.95.1.12	open_file	597
5.95.1.13	open_local	597
5.96	peripheral/atadevice.h File Reference	598
5.96.1	Define Documentation	601
5.96.1.1	ATA_ASR	601
5.96.1.2	ATA_CHR	601
5.96.1.3	ATA_CLR	601
5.96.1.4	ATA_CR	601
5.96.1.5	ATA_DA	601
5.96.1.6	ATA_DAR_DS0	601
5.96.1.7	ATA_DAR_DS1	601
5.96.1.8	ATA_DAR_H	601
5.96.1.9	ATA_DAR_WTG	601
5.96.1.10	ATA_DCR	601
5.96.1.11	ATA_DCR_IEN	601
5.96.1.12	ATA_DCR_RST	601
5.96.1.13	ATA_DHR	601
5.96.1.14	ATA_DHR_DEV	601
5.96.1.15	ATA_DHR_H	601
5.96.1.16	ATA_DHR_LBA	601
5.96.1.17	ATA_DR	601
5.96.1.18	ATA_ERR	601
5.96.1.19	ATA_ERR_ABT	601
5.96.1.20	ATA_ERR_AMN	601
5.96.1.21	ATA_ERR_BBK	601
5.96.1.22	ATA_ERR_IDNF	601
5.96.1.23	ATA_ERR_TON	601
5.96.1.24	ATA_ERR_UNC	601
5.96.1.25	ATA_FR	601
5.96.1.26	ATA_SCR	601
5.96.1.27	ATA_SNR	601
5.96.1.28	ATA_SR	601
5.96.1.29	ATA_SR_BSY	601

5.96.1.30	ATA_SR_COR	601
5.96.1.31	ATA_SR_DF	601
5.96.1.32	ATA_SR_DRDY	601
5.96.1.33	ATA_SR_DRQ	601
5.96.1.34	ATA_SR_DSC	601
5.96.1.35	ATA_SR_ERR	601
5.96.1.36	ATA_SR_IDX	601
5.96.1.37	ATA_STATE_HW_RST	601
5.96.1.38	ATA_STATE_IDLE	601
5.96.1.39	ATA_STATE_SW_RST	601
5.96.1.40	TYPE_FILE	601
5.96.1.41	TYPE_LOCAL	601
5.96.1.42	TYPE_NO_CONNECT	601
5.96.2	Function Documentation	601
5.96.2.1	ata_devices_hw_reset	601
5.96.2.2	ata_devices_init	602
5.96.2.3	ata_devices_read	602
5.96.2.4	ata_devices_write	602
5.97	peripheral/atadevice_cmdi.c File Reference	603
5.97.1	Function Documentation	604
5.97.1.1	ata_calc_lba	604
5.97.1.2	ata_cmd_complete	604
5.97.1.3	ata_device_execute_cmd	604
5.97.1.4	ata_device_reset_cmd	604
5.97.1.5	ata_execute_device_diagnostics_cmd	605
5.97.1.6	ata_identify_device_cmd	605
5.97.1.7	ata_initialize_device_parameters_cmd	605
5.97.1.8	ata_read_native_max_addr	605
5.97.1.9	ata_read_sect	605
5.97.1.10	ata_read_sectors_cmd	606
5.97.1.11	ata_set_device_signature	606
5.97.1.12	ata_set_features	606
5.97.1.13	ata_set_sect	606
5.97.1.14	ata_write_sect	606
5.97.1.15	ata_write_sectors	607
5.97.1.16	DEFAULT_DEBUG_CHANNEL	607

5.98 peripheral/atadevice_cmdi.h File Reference	608
5.98.1 Define Documentation	611
5.98.1.1 BYTES_PER_SECTOR	611
5.98.1.2 MIN_MWDMA_CYCLE_TIME	611
5.98.1.3 MIN_PIO_CYCLE_TIME_IORDY	611
5.98.1.4 MIN_PIO_CYCLE_TIME_NO_IORDY	611
5.98.1.5 QUEUE_DEPTH	611
5.98.1.6 RECOMMENDED_MWDMA_CYCLE_TIME	611
5.98.1.7 SET_FEATURES_REQUIRED_AFTER_POWER_UP	611
5.98.1.8 SUPPORT_APM	611
5.98.1.9 SUPPORT_CFA	611
5.98.1.10 SUPPORT_DEVICE_RESET_CMD	611
5.98.1.11 SUPPORT_DOWNLOAD_MICROCODE	611
5.98.1.12 SUPPORT_HOST_PROTECTED_AREA	611
5.98.1.13 SUPPORT_LOOKAHEAD	611
5.98.1.14 SUPPORT_NOP_CMD	611
5.98.1.15 SUPPORT_POWER_MANAGEMENT	611
5.98.1.16 SUPPORT_POWER_UP_IN_STANDBY_MODE	611
5.98.1.17 SUPPORT_READ_BUFFER_CMD	611
5.98.1.18 SUPPORT_READ_WRITE_DMA_QUEUED	611
5.98.1.19 SUPPORT_RELEASE_INTERRUPT	611
5.98.1.20 SUPPORT_REMOVABLE_MEDIA	611
5.98.1.21 SUPPORT_REMOVABLE_MEDIA_NOTIFICATION	611
5.98.1.22 SUPPORT_SECURITY_MODE	611
5.98.1.23 SUPPORT_SERVICE_INTERRUPT	611
5.98.1.24 SUPPORT_SET_MAX	611
5.98.1.25 SUPPORT_SMART	611
5.98.1.26 SUPPORT_WRITE_BUFFER_CMD	611
5.98.1.27 SUPPORT_WRITE_CACHE	611
5.98.2 Function Documentation	611
5.98.2.1 ata_device_execute_cmd	611
5.98.2.2 ata_execute_device_diagnostics_cmd	612
5.99 peripheral/atahost.c File Reference	613
5.99.1 Define Documentation	615
5.99.1.1 DMA_MODE0_TD	615
5.99.1.2 DMA_MODE0_TEOC	615

5.99.1.3	DMA_MODE0_TM	615
5.99.1.4	PIO_MODE0_T1	615
5.99.1.5	PIO_MODE0_T2	615
5.99.1.6	PIO_MODE0_T4	615
5.99.1.7	PIO_MODE0_TEOC	615
5.99.2	Function Documentation	615
5.99.2.1	ata_baseaddr	615
5.99.2.2	ata_dev_id	615
5.99.2.3	ata_dma_mode0_td	615
5.99.2.4	ata_dma_mode0_teoc	615
5.99.2.5	ata_dma_mode0_tm	615
5.99.2.6	ata_enabled	615
5.99.2.7	ata_enddevice	615
5.99.2.8	ata_file	615
5.99.2.9	ata_firmware	616
5.99.2.10	ata_heads	616
5.99.2.11	ata_int	616
5.99.2.12	ata_irq	616
5.99.2.13	ata_mwdma	616
5.99.2.14	ata_packet	616
5.99.2.15	ata_pio	616
5.99.2.16	ata_pio_mode0_t1	617
5.99.2.17	ata_pio_mode0_t2	617
5.99.2.18	ata_pio_mode0_t4	617
5.99.2.19	ata_pio_mode0_teoc	617
5.99.2.20	ata_read32	617
5.99.2.21	ata_reset	617
5.99.2.22	ata_rev	617
5.99.2.23	ata_sec_end	618
5.99.2.24	ata_sec_start	618
5.99.2.25	ata_sectors	619
5.99.2.26	ata_size	619
5.99.2.27	ata_start_device	619
5.99.2.28	ata_status	619
5.99.2.29	ata_type	619
5.99.2.30	ata_write32	619

5.99.2.31	DEFAULT_DEBUG_CHANNEL	621
5.99.2.32	reg_ata_sec	621
5.99.3	Variable Documentation	622
5.99.3.1	conf_dev	622
5.100	peripheral/atahost.h File Reference	623
5.100.1	Define Documentation	626
5.100.1.1	ATA_ADDR_SPACE	626
5.100.1.2	ATA_BELECO	626
5.100.1.3	ATA_BELEC1	626
5.100.1.4	ATA_CTRL	626
5.100.1.5	ATA_DEVID	626
5.100.1.6	ata_dma_delay	626
5.100.1.7	ATA_DMA_EN	626
5.100.1.8	ATA_DMA_RD	626
5.100.1.9	ATA_DMA_TIP	626
5.100.1.10	ATA_DMA_WR	626
5.100.1.11	ATA_DMARQ	626
5.100.1.12	ATA_DRBE	626
5.100.1.13	ATA_DTBF	626
5.100.1.14	ATA_DTR0	626
5.100.1.15	ATA_DTR1	626
5.100.1.16	ATA_FTE0	626
5.100.1.17	ATA_FTE1	626
5.100.1.18	ATA_IDE_EN	626
5.100.1.19	ATA_IDEIS	626
5.100.1.20	ATA_IORDY	626
5.100.1.21	ATA_IORDY_FTE0	626
5.100.1.22	ATA_IORDY_FTE1	626
5.100.1.23	ATA_PCTR	626
5.100.1.24	ATA_PFTR0	626
5.100.1.25	ATA_PFTR1	626
5.100.1.26	ata_pio_delay	626
5.100.1.27	ATA_PIO_TIP	626
5.100.1.28	ATA_PWPP	626
5.100.1.29	ATA_PWPPF	626
5.100.1.30	ATA_REVNO	626

5.100.1.31	ATA_RST	626
5.100.1.32	ATA_RXB	626
5.100.1.33	ATA_STAT	626
5.100.1.34	ATA_T1	626
5.100.1.35	ATA_T2	626
5.100.1.36	ATA_T4	626
5.100.1.37	ATA_TD	626
5.100.1.38	ATA_TEOC	626
5.100.1.39	ATA_TM	626
5.100.1.40	ATA_TXB	626
5.100.1.4	lis_ata_hostadr	626
5.100.2	Function Documentation	626
5.100.2.1	ata_int	626
5.100.2.2	reg_ata_sec	628
5.101	peripheral/atahost_define.h File Reference	629
5.101.1	Define Documentation	629
5.101.1.1	DMA_MODE0_TD	629
5.101.1.2	DMA_MODE0_TEOC	629
5.101.1.3	DMA_MODE0_TM	629
5.101.1.4	PIO_MODE0_T1	629
5.101.1.5	PIO_MODE0_T2	629
5.101.1.6	PIO_MODE0_T4	629
5.101.1.7	PIO_MODE0_TEOC	629
5.102	peripheral/channels/channel.c File Reference	630
5.102.1	Function Documentation	631
5.102.1.1	channel_close	631
5.102.1.2	channel_init	631
5.102.1.3	channel_open	631
5.102.1.4	channel_read	631
5.102.1.5	channel_write	631
5.102.1.6	find_channel_factory	631
5.102.2	Variable Documentation	631
5.102.2.1	head	631
5.102.2.2	preloaded	631
5.103	peripheral/channels/channel.h File Reference	632
5.103.1	Function Documentation	632

5.103.1.1	channel_close	632
5.103.1.2	channel_init	632
5.103.1.3	channel_open	632
5.103.1.4	channel_read	632
5.103.1.5	channel_write	632
5.104	peripheral/channels/fd.c File Reference	633
5.104.1	Function Documentation	634
5.104.1.1	fd_init	634
5.104.1.2	fd_isok	634
5.104.1.3	fd_read	634
5.104.1.4	fd_status	634
5.104.1.5	fd_status_fd	634
5.104.1.6	fd_write	634
5.104.2	Variable Documentation	634
5.104.2.1	fd_channel_ops	634
5.105	peripheral/channels/fd.h File Reference	635
5.105.1	Function Documentation	635
5.105.1.1	fd_read	635
5.105.1.2	fd_write	635
5.105.2	Variable Documentation	635
5.105.2.1	fd_channel_ops	635
5.106	peripheral/channels/file.c File Reference	636
5.106.1	Function Documentation	637
5.106.1.1	file_close	637
5.106.1.2	file_free	637
5.106.1.3	file_init	637
5.106.1.4	file_open	637
5.106.2	Variable Documentation	637
5.106.2.1	file_channel_ops	637
5.107	peripheral/channels/file.h File Reference	638
5.107.1	Variable Documentation	638
5.107.1.1	file_channel_ops	638
5.108	peripheral/channels/tcp.c File Reference	639
5.108.1	Function Documentation	640
5.108.1.1	tcp_init	640
5.108.1.2	tcp_open	640

5.108.1.3 tcp_read	640
5.108.1.4 tcp_write	640
5.108.1.5 wait_for_tcp_connect	640
5.108.2 Variable Documentation	640
5.108.2.1 tcp_channel_ops	640
5.109peripheral/channels/tcp.h File Reference	641
5.109.1 Variable Documentation	641
5.109.1.1 tcp_channel_ops	641
5.110peripheral/channels/tty.c File Reference	642
5.110.1 Define Documentation	643
5.110.1.1 DEFAULT_BAUD	643
5.110.1.2 DEFAULT_TTY_DEVICE	643
5.110.2 Function Documentation	643
5.110.2.1 parse_baud	643
5.110.2.2 tty_init	643
5.110.2.3 tty_open	643
5.110.3 Variable Documentation	643
5.110.3.1 baud_table	643
5.110.3.2 name	643
5.110.3.3 tty_channel_ops	643
5.110.3.4 value	643
5.111peripheral/channels/tty.h File Reference	644
5.111.1 Variable Documentation	644
5.111.1.1 tty_channel_ops	644
5.112peripheral/channels/xterm.c File Reference	645
5.112.1 Define Documentation	646
5.112.1.1 MAX_XTERM_ARGS	646
5.112.2 Function Documentation	646
5.112.2.1 basename	646
5.112.2.2 xterm_close	646
5.112.2.3 xterm_init	646
5.112.2.4 xterm_open	646
5.112.3 Variable Documentation	646
5.112.3.1 xterm_channel_ops	646
5.113peripheral/channels/xterm.h File Reference	647
5.113.1 Variable Documentation	647

5.113.1.1	xterm_channel_ops	647
5.114	peripheral/crc32.c File Reference	648
5.114.1	Function Documentation	648
5.114.1.1	crc32	648
5.114.1.2	crc32_close	648
5.114.1.3	crc32_feed_bytes	648
5.114.1.4	crc32_init	648
5.114.2	Variable Documentation	648
5.114.2.1	crc32_table	648
5.115	peripheral/crc32.h File Reference	649
5.115.1	Function Documentation	649
5.115.1.1	crc32	649
5.115.1.2	crc32_close	649
5.115.1.3	crc32_feed_bytes	649
5.115.1.4	crc32_init	649
5.116	peripheral/dma-defs.h File Reference	650
5.116.1	Define Documentation	652
5.116.1.1	DMA_ADDR_SPACE	652
5.116.1.2	DMA_CH_A0	652
5.116.1.3	DMA_CH_A0_ADDR_OFFSET	652
5.116.1.4	DMA_CH_A0_ADDR_WIDTH	652
5.116.1.5	DMA_CH_A1	652
5.116.1.6	DMA_CH_A1_ADDR_OFFSET	652
5.116.1.7	DMA_CH_A1_ADDR_WIDTH	652
5.116.1.8	DMA_CH_AM0	652
5.116.1.9	DMA_CH_AM0_MASK_OFFSET	652
5.116.1.10	DMA_CH_AM0_MASK_WIDTH	652
5.116.1.11	DMA_CH_AM1	652
5.116.1.12	DMA_CH_AM1_MASK_OFFSET	652
5.116.1.13	DMA_CH_AM1_MASK_WIDTH	652
5.116.1.14	DMA_CH_BASE	652
5.116.1.15	DMA_CH_CSR	654
5.116.1.16	DMA_CH_CSR_ARS_OFFSET	654
5.116.1.17	DMA_CH_CSR_BUSY_OFFSET	654
5.116.1.18	DMA_CH_CSR_CH_EN_OFFSET	654
5.116.1.19	DMA_CH_CSR_DONE_OFFSET	654

5.116.1.20	DMA_CH_CSR_DST_SEL_OFFSET	654
5.116.1.21	DMA_CH_CSR_ERR_OFFSET	654
5.116.1.22	DMA_CH_CSR_INC_DST_OFFSET	654
5.116.1.23	DMA_CH_CSR_INC_SRC_OFFSET	654
5.116.1.24	DMA_CH_CSR_INE_CHK_DONE_OFFSET	654
5.116.1.25	DMA_CH_CSR_INE_DONE_OFFSET	654
5.116.1.26	DMA_CH_CSR_INE_ERR_OFFSET	654
5.116.1.27	DMA_CH_CSR_INT_CHUNK_DONE_OFFSET	654
5.116.1.28	DMA_CH_CSR_INT_DONE_OFFSET	654
5.116.1.29	DMA_CH_CSR_INT_ERR_OFFSET	654
5.116.1.30	DMA_CH_CSR_MODE_OFFSET	654
5.116.1.31	DMA_CH_CSR_PRIORITY_OFFSET	654
5.116.1.32	DMA_CH_CSR_PRIORITY_WIDTH	654
5.116.1.33	DMA_CH_CSR_RESERVED_OFFSET	654
5.116.1.34	DMA_CH_CSR_RESERVED_WIDTH	654
5.116.1.35	DMA_CH_CSR_REST_EN_OFFSET	654
5.116.1.36	DMA_CH_CSR_SRC_SEL_OFFSET	654
5.116.1.37	DMA_CH_CSR_STOP_OFFSET	654
5.116.1.38	DMA_CH_CSR_SZ_WB_OFFSET	654
5.116.1.39	DMA_CH_CSR_USE_ED_OFFSET	654
5.116.1.40	DMA_CH_CSR_WRITE_MASK	654
5.116.1.41	DMA_CH_DESC	654
5.116.1.42	DMA_CH_DESC_ADDR_OFFSET	654
5.116.1.43	DMA_CH_DESC_ADDR_WIDTH	654
5.116.1.44	DMA_CH_SIZE	654
5.116.1.45	DMA_CH_SWPTR	656
5.116.1.46	DMA_CH_SWPTR_EN_OFFSET	656
5.116.1.47	DMA_CH_SWPTR_PTR_OFFSET	656
5.116.1.48	DMA_CH_SWPTR_PTR_WIDTH	656
5.116.1.49	DMA_CH_SZ	656
5.116.1.50	DMA_CH_SZ_CHK_SZ_OFFSET	656
5.116.1.51	DMA_CH_SZ_CHK_SZ_WIDTH	656
5.116.1.52	DMA_CH_SZ_TOT_SZ_OFFSET	656
5.116.1.53	DMA_CH_SZ_TOT_SZ_WIDTH	656
5.116.1.54	DMA_CSR	656
5.116.1.55	DMA_CSR_PAUSE_OFFSET	656

5.116.1.56	DMA_DESC_ADR0	656
5.116.1.57	DMA_DESC_ADR1	656
5.116.1.58	DMA_DESC_CSR	656
5.116.1.59	DMA_DESC_CSR_DST_SEL_OFFSET	656
5.116.1.60	DMA_DESC_CSR_EOL_OFFSET	656
5.116.1.61	DMA_DESC_CSR_INC_DST_OFFSET	656
5.116.1.62	DMA_DESC_CSR_INC_SRC_OFFSET	656
5.116.1.63	DMA_DESC_CSR_SRC_SEL_OFFSET	656
5.116.1.64	DMA_DESC_CSR_TOT_SZ_OFFSET	656
5.116.1.65	DMA_DESC_CSR_TOT_SZ_WIDTH	656
5.116.1.66	DMA_DESC_NEXT	656
5.116.1.67	DMA_INT_MSK_A	656
5.116.1.68	DMA_INT_MSK_B	656
5.116.1.69	DMA_INT_SRC_A	656
5.116.1.70	DMA_INT_SRC_B	656
5.116.1.71	DMA_NUM_CHANNELS	656
5.117	peripheral/dma.c File Reference	657
5.117.1	Define Documentation	659
5.117.1.1	CHANNEL_ND_I	659
5.117.2	Function Documentation	659
5.117.2.1	check_dma_ack_o	659
5.117.2.2	clear_dma_nd_i	659
5.117.2.3	clear_dma_req_i	659
5.117.2.4	DEFAULT_DEBUG_CHANNEL	659
5.117.2.5	dma_baseaddr	659
5.117.2.6	dma_channel_clock	659
5.117.2.7	dma_channel_terminate_transfer	660
5.117.2.8	dma_enabled	660
5.117.2.9	dma_init_transfer	660
5.117.2.10	dma_irq	660
5.117.2.11	dma_load_descriptor	660
5.117.2.12	dma_read32	661
5.117.2.13	dma_read_ch_csr	661
5.117.2.14	dma_reset	661
5.117.2.15	dma_sec_end	661
5.117.2.16	dma_sec_start	661

5.117.2.17	<code>dma_status</code>	662
5.117.2.18	<code>dma_vapi_id</code>	662
5.117.2.19	<code>dma_write32</code>	662
5.117.2.20	<code>dma_write_ch_csr</code>	662
5.117.2.21	<code>find_dma_controller_ch</code>	663
5.117.2.22	<code>masked_increase</code>	663
5.117.2.23	<code>reg_dma_sec</code>	663
5.117.2.24	<code>set_dma_nd_i</code>	663
5.117.2.25	<code>set_dma_req_i</code>	663
5.117.3	Variable Documentation	663
5.117.3.1	<code>dmass</code>	663
5.118	<code>peripheral/dma.h</code> File Reference	664
5.118.1	Function Documentation	665
5.118.1.1	<code>check_dma_ack_o</code>	665
5.118.1.2	<code>clear_dma_nd_i</code>	665
5.118.1.3	<code>clear_dma_req_i</code>	665
5.118.1.4	<code>find_dma_controller_ch</code>	665
5.118.1.5	<code>reg_dma_sec</code>	665
5.118.1.6	<code>set_dma_nd_i</code>	665
5.118.1.7	<code>set_dma_req_i</code>	665
5.119	<code>peripheral/eth.c</code> File Reference	666
5.119.1	Define Documentation	672
5.119.1.1	<code>ETH_ADDR_SPACE</code>	672
5.119.1.2	<code>ETH_ALEN</code>	672
5.119.1.3	<code>ETH_BD_BASE</code>	672
5.119.1.4	<code>ETH_BD_COUNT</code>	672
5.119.1.5	<code>ETH_BD_SPACE</code>	672
5.119.1.6	<code>ETH_CMODER_PASSALL_OFFSET</code>	672
5.119.1.7	<code>ETH_CMODER_RXFLOW_OFFSET</code>	672
5.119.1.8	<code>ETH_CMODER_TXFLOW_OFFSET</code>	672
5.119.1.9	<code>ETH_COLLCONF</code>	672
5.119.1.10	<code>ETH_COLLCONF_COLLVALID_OFFSET</code>	672
5.119.1.11	<code>ETH_COLLCONF_COLLVALID_WIDTH</code>	672
5.119.1.12	<code>ETH_COLLCONF_MAXRET_OFFSET</code>	672
5.119.1.13	<code>ETH_COLLCONF_MAXRET_WIDTH</code>	672
5.119.1.14	<code>ETH_CTRLMODER</code>	672

5.119.1.15	ETH_DMA_RX_TX	672
5.119.1.16	ETH_HASH0	672
5.119.1.17	ETH_HASH1	672
5.119.1.18	ETH_INT_MASK	672
5.119.1.19	ETH_INT_MASK_BUSY_M_OFFSET	672
5.119.1.20	ETH_INT_MASK_RXB_M_OFFSET	672
5.119.1.21	ETH_INT_MASK_RXC_M_OFFSET	672
5.119.1.22	ETH_INT_MASK_RXE_M_OFFSET	672
5.119.1.23	ETH_INT_MASK_TXB_M_OFFSET	672
5.119.1.24	ETH_INT_MASK_TXC_M_OFFSET	672
5.119.1.25	ETH_INT_MASK_TXE_M_OFFSET	672
5.119.1.26	ETH_INT_SOURCE	672
5.119.1.27	ETH_INT_SOURCE_BUSY_OFFSET	672
5.119.1.28	ETH_INT_SOURCE_RXB_OFFSET	672
5.119.1.29	ETH_INT_SOURCE_RXC_OFFSET	672
5.119.1.30	ETH_INT_SOURCE_RXE_OFFSET	672
5.119.1.31	ETH_INT_SOURCE_TXB_OFFSET	672
5.119.1.32	ETH_INT_SOURCE_TXC_OFFSET	672
5.119.1.33	ETH_INT_SOURCE_TXE_OFFSET	672
5.119.1.34	ETH_IPGR1	672
5.119.1.35	ETH_IPGR2	672
5.119.1.36	ETH_IPGT	672
5.119.1.37	ETH_MAC_ADDR0	672
5.119.1.38	ETH_MAC_ADDR1	672
5.119.1.39	ETH_MAXPL	672
5.119.1.40	ETH_MIIADDR_FIAD_OFFSET	672
5.119.1.41	ETH_MIIADDR_FIAD_WIDTH	672
5.119.1.42	ETH_MIIADDR_RGAD_OFFSET	672
5.119.1.43	ETH_MIIADDR_RGAD_WIDTH	672
5.119.1.44	ETH_MIIADDRESS	672
5.119.1.45	ETH_MIICOMM_RSTAT_OFFSET	672
5.119.1.46	ETH_MIICOMM_SCANS_OFFSET	672
5.119.1.47	ETH_MIICOMM_WCDATA_OFFSET	672
5.119.1.48	ETH_MIICOMMAND	672
5.119.1.49	ETH_MIIMODER	672
5.119.1.50	ETH_MIIMODER_CLKDIV_OFFSET	672

5.119.1.51	ETH_MIIMODER_CLKDIV_WIDTH	672
5.119.1.52	ETH_MIIMODER_MRST_OFFSET	672
5.119.1.53	ETH_MIIMODER_NOPRE_OFFSET	672
5.119.1.54	ETH_MIIRX_DATA	672
5.119.1.55	ETH_MIISTAT_BUSY_OFFSET	672
5.119.1.56	ETH_MIISTAT_FAIL_OFFSET	672
5.119.1.57	ETH_MIISTAT_NVALID_OFFSET	672
5.119.1.58	ETH_MIISTATUS	672
5.119.1.59	ETH_MIITX_DATA	672
5.119.1.60	ETH_MODER	672
5.119.1.61	ETH_MODER_BRO_OFFSET	672
5.119.1.62	ETH_MODER_CRCEN_OFFSET	672
5.119.1.63	ETH_MODER_DLYCRCEN_OFFSET	672
5.119.1.64	ETH_MODER_DMAEN_OFFSET	672
5.119.1.65	ETH_MODER_EXDFREN_OFFSET	672
5.119.1.66	ETH_MODER_FULLD_OFFSET	672
5.119.1.67	ETH_MODER_HUGEN_OFFSET	672
5.119.1.68	ETH_MODER_IAM_OFFSET	672
5.119.1.69	ETH_MODER_IFG_OFFSET	672
5.119.1.70	ETH_MODER_LOOPBCK_OFFSET	672
5.119.1.71	ETH_MODER_NOBCKOF_OFFSET	672
5.119.1.72	ETH_MODER_NOPRE_OFFSET	672
5.119.1.73	ETH_MODER_PAD_OFFSET	672
5.119.1.74	ETH_MODER_PRO_OFFSET	672
5.119.1.75	ETH_MODER_RECSCALL_OFFSET	672
5.119.1.76	ETH_MODER_RST_OFFSET	672
5.119.1.77	ETH_MODER_RXEN_OFFSET	672
5.119.1.78	ETH_MODER_TXEN_OFFSET	672
5.119.1.79	ETH_PACKETLEN	672
5.119.1.80	ETH_PACKETLEN_MAXFL_OFFSET	672
5.119.1.81	ETH_PACKETLEN_MAXFL_WIDTH	672
5.119.1.82	ETH_PACKETLEN_MINFL_OFFSET	672
5.119.1.83	ETH_PACKETLEN_MINFL_WIDTH	672
5.119.1.84	ETH_RTX_FILE	672
5.119.1.85	ETH_RTX_SOCK	672
5.119.1.86	ETH_RTX_VAPI	672

5.119.1.87	ETH_RX_BD_COLLISION_OFFSET	672
5.119.1.88	ETH_RX_BD_CRC_OFFSET	672
5.119.1.89	ETH_RX_BD_DRIBBLE_OFFSET	672
5.119.1.90	ETH_RX_BD_INVALID_OFFSET	672
5.119.1.91	ETH_RX_BD_IRQ_OFFSET	672
5.119.1.92	ETH_RX_BD_LENGTH_OFFSET	672
5.119.1.93	ETH_RX_BD_LENGTH_WIDTH	672
5.119.1.94	ETH_RX_BD_MISS_OFFSET	672
5.119.1.95	ETH_RX_BD_READY_OFFSET	672
5.119.1.96	ETH_RX_BD_TOOBIG_OFFSET	672
5.119.1.97	ETH_RX_BD_TOOSHORT_OFFSET	672
5.119.1.98	ETH_RX_BD_UVERRUN_OFFSET	672
5.119.1.99	ETH_RX_BD_WRAP_OFFSET	672
5.119.1.100	ETH_RXSTATE_IDLE	672
5.119.1.101	ETH_RXSTATE_RECV	672
5.119.1.102	ETH_RXSTATE_WAIT4BD	672
5.119.1.103	ETH_RXSTATE_WRITEFIFO	672
5.119.1.104	ETH_TX_BD_COLLISION_OFFSET	672
5.119.1.105	ETH_TX_BD_CRC_OFFSET	672
5.119.1.106	ETH_TX_BD_DEFER_OFFSET	672
5.119.1.107	ETH_TX_BD_IRQ_OFFSET	672
5.119.1.108	ETH_TX_BD_LAST_OFFSET	672
5.119.1.109	ETH_TX_BD_LENGTH_OFFSET	672
5.119.1.110	ETH_TX_BD_LENGTH_WIDTH	672
5.119.1.111	ETH_TX_BD_NO_CARRIER_OFFSET	672
5.119.1.112	ETH_TX_BD_NUM	672
5.119.1.113	ETH_TX_BD_PAD_OFFSET	672
5.119.1.114	ETH_TX_BD_PAUSE_OFFSET	672
5.119.1.115	ETH_TX_BD_READY_OFFSET	672
5.119.1.116	ETH_TX_BD_RETRANSMIT_OFFSET	672
5.119.1.117	ETH_TX_BD_RETRY_OFFSET	672
5.119.1.118	ETH_TX_BD_RETRY_WIDTH	672
5.119.1.119	ETH_TX_BD_UNDERRUN_OFFSET	672
5.119.1.120	ETH_TX_BD_WRAP_OFFSET	672
5.119.1.121	ETH_TXSTATE_IDLE	672
5.119.1.122	ETH_TXSTATE_READFIFO	672

5.119.1.12	BTH_TXSTATE_TRANSMIT	672
5.119.1.12	BTH_TXSTATE_WAIT4BD	672
5.119.1.12	BETHER_ADDR_LEN	672
5.119.1.12	BETHER_CRC_LEN	672
5.119.1.12	BETHER_HDR_LEN	672
5.119.1.12	BETHER_IS_VALID_LEN	672
5.119.1.12	BETHER_MAX_LEN	672
5.119.1.13	BETHER_MIN_LEN	672
5.119.1.13	BETHER_TYPE_LEN	672
5.119.1.13	BETHERMIN	672
5.119.1.13	BETHERMTU	672
5.119.1.13	BETHERTYPE_ARP	672
5.119.1.13	BETHERTYPE_IP	672
5.119.1.13	BETHERTYPE_NTRAILER	672
5.119.1.13	BETHERTYPE_PUP	672
5.119.1.13	BETHERTYPE_REVARP	672
5.119.1.13	BETHERTYPE_TRAIL	672
5.119.2	Enumeration Type Documentation	672
5.119.2.1	"@44	672
5.119.3	Function Documentation	673
5.119.3.1	DEFAULT_DEBUG_CHANNEL	673
5.119.3.2	eth_baseaddr	673
5.119.3.3	eth_controller_rx_clock	673
5.119.3.4	eth_controller_tx_clock	674
5.119.3.5	eth_dma	674
5.119.3.6	eth_enabled	674
5.119.3.7	eth_irq	674
5.119.3.8	eth_read32	675
5.119.3.9	eth_read_rx_file	675
5.119.3.10	eth_reset	675
5.119.3.11	eth_rtx_type	675
5.119.3.12	eth_rx_channel	675
5.119.3.13	eth_rx_next_packet	675
5.119.3.14	eth_rxfile	676
5.119.3.15	eth_sec_end	678
5.119.3.16	eth_sec_start	679

5.119.3.17	eth_skip_rx_file	679
5.119.3.18	eth_sockif	679
5.119.3.19	eth_status	679
5.119.3.20	eth_tx_channel	679
5.119.3.21	eth_txfile	679
5.119.3.22	eth_vapi_id	680
5.119.3.23	eth_vapi_read	680
5.119.3.24	eth_write32	680
5.119.3.25	eth_write_tx_bd_num	680
5.119.3.26	reg_ethernet_sec	680
5.120	peripheral/eth.h File Reference	682
5.120.1	Function Documentation	682
5.120.1.1	reg_ethernet_sec	682
5.121	peripheral/fb.c File Reference	684
5.121.1	Define Documentation	685
5.121.1.1	CAM_SIZEX	685
5.121.1.2	CAM_SIZEY	685
5.121.1.3	CNV16	685
5.121.1.4	CNV32	685
5.121.1.5	FB_BUFADDR	685
5.121.1.6	FB_CAMBUFADDR	685
5.121.1.7	FB_CAMPOSADDR	685
5.121.1.8	FB_CTRL	685
5.121.1.9	FB_PAL	685
5.121.1.10	FB_SIZEX	685
5.121.1.11	FB_SIZEY	685
5.121.1.12	FB_WRAP	685
5.121.1.13	REFRESH_DIVIDER	685
5.121.2	Function Documentation	686
5.121.2.1	change_buf_addr	686
5.121.2.2	fb_baseaddr	686
5.121.2.3	fb_dump_image24	686
5.121.2.4	fb_dump_image8	686
5.121.2.5	fb_enabled	686
5.121.2.6	fb_filename	686
5.121.2.7	fb_job	687

5.121.2.8 fb_read32	687
5.121.2.9 fb_refresh_rate	687
5.121.2.10fb_reset	687
5.121.2.11fb_sec_end	688
5.121.2.12fb_sec_start	688
5.121.2.13fb_write32	688
5.121.2.14reg_fb_sec	689
5.122peripheral/fb.h File Reference	690
5.122.1 Function Documentation	690
5.122.1.1 reg_fb_sec	690
5.123peripheral/fields.h File Reference	691
5.123.1 Define Documentation	691
5.123.1.1 ASSIGN_FLAG	691
5.123.1.2 CLEAR_FLAG	692
5.123.1.3 FIELD_MASK	692
5.123.1.4 FIELD_SHIFT	692
5.123.1.5 FLAG_MASK	692
5.123.1.6 FLAG_SHIFT	692
5.123.1.7 GET_FIELD	692
5.123.1.8 SET_FIELD	692
5.123.1.9 SET_FLAG	692
5.123.1.10TEST_FLAG	692
5.124peripheral/generic.c File Reference	693
5.124.1 Function Documentation	694
5.124.1.1 ext_read	694
5.124.1.2 ext_write	694
5.124.1.3 generic_baseaddr	694
5.124.1.4 generic_byte_enabled	694
5.124.1.5 generic_enabled	694
5.124.1.6 generic_hw_enabled	694
5.124.1.7 generic_name	694
5.124.1.8 generic_read_byte	694
5.124.1.9 generic_read_hw	694
5.124.1.10generic_read_word	695
5.124.1.11generic_reset	695
5.124.1.12generic_sec_end	695

5.124.1.13	generic_sec_start	696
5.124.1.14	generic_size	696
5.124.1.15	generic_status	696
5.124.1.16	generic_word_enabled	696
5.124.1.17	generic_write_byte	696
5.124.1.18	generic_write_hw	696
5.124.1.19	generic_write_word	696
5.124.1.20	reg_generic_sec	697
5.125	peripheral/channels/generic.c File Reference	698
5.125.1	Function Documentation	698
5.125.1.1	generic_close	698
5.125.1.2	generic_free	698
5.125.1.3	generic_open	698
5.126	peripheral/generic.h File Reference	699
5.126.1	Function Documentation	699
5.126.1.1	reg_generic_sec	699
5.127	peripheral/channels/generic.h File Reference	700
5.127.1	Function Documentation	700
5.127.1.1	generic_close	700
5.127.1.2	generic_free	700
5.127.1.3	generic_open	700
5.128	peripheral/gpio.c File Reference	701
5.128.1	Define Documentation	703
5.128.1.1	GPIO_ADDR_SPACE	703
5.128.1.2	RGPIO_AUX	703
5.128.1.3	RGPIO_CTRL	703
5.128.1.4	RGPIO_CTRL_ECLK	703
5.128.1.5	RGPIO_CTRL_INTE	703
5.128.1.6	RGPIO_CTRL_INTS	703
5.128.1.7	RGPIO_CTRL_NEC	703
5.128.1.8	RGPIO_IN	703
5.128.1.9	RGPIO_INTE	703
5.128.1.10	RGPIO_INTS	703
5.128.1.11	RGPIO_OE	703
5.128.1.12	RGPIO_OUT	703
5.128.1.13	RGPIO_PTRIG	703

5.128.2 Enumeration Type Documentation	703
5.128.2.1 "@50	703
5.128.3 Function Documentation	704
5.128.3.1 DEFAULT_DEBUG_CHANNEL	704
5.128.3.2 gpio_base_vapi_id	704
5.128.3.3 gpio_baseaddr	704
5.128.3.4 gpio_clock	704
5.128.3.5 gpio_device_clock	704
5.128.3.6 gpio_do_int	705
5.128.3.7 gpio_enabled	705
5.128.3.8 gpio_external_clock	705
5.128.3.9 gpio_irq	706
5.128.3.10 gpio_read32	706
5.128.3.11 gpio_reset	706
5.128.3.12 gpio_sec_end	706
5.128.3.13 gpio_sec_start	707
5.128.3.14 gpio_status	707
5.128.3.15 gpio_vapi_read	707
5.128.3.16 gpio_write32	707
5.128.3.17 reg_gpio_sec	708
5.129 peripheral/gpio.h File Reference	709
5.129.1 Function Documentation	709
5.129.1.1 reg_gpio_sec	709
5.130 peripheral/mc.c File Reference	710
5.130.1 Define Documentation	714
5.130.1.1 MC_ADDR_SPACE	714
5.130.1.2 MC_BA_MASK	714
5.130.1.3 MC_BA_MASK_VALID	714
5.130.1.4 MC_CSC	714
5.130.1.5 MC_CSC_BAS_OFFSET	714
5.130.1.6 MC_CSC_BW_OFFSET	714
5.130.1.7 MC_CSC_BW_WIDTH	714
5.130.1.8 MC_CSC_EN_OFFSET	714
5.130.1.9 MC_CSC_KRO_OFFSET	714
5.130.1.10 MC_CSC_MEMTYPE_ASYNC	714
5.130.1.11 MC_CSC_MEMTYPE_OFFSET	714

5.130.1.12	MC_CSC_MEMTYPE_SDRAM	714
5.130.1.13	MC_CSC_MEMTYPE_SSRAM	714
5.130.1.14	MC_CSC_MEMTYPE_SYNC	714
5.130.1.15	MC_CSC_MEMTYPE_WIDTH	714
5.130.1.16	MC_CSC_MS_OFFSET	714
5.130.1.17	MC_CSC_MS_WIDTH	714
5.130.1.18	MC_CSC_PEN_OFFSET	714
5.130.1.19	MC_CSC_SEL_OFFSET	714
5.130.1.20	MC_CSC_SEL_WIDTH	714
5.130.1.21	MC_CSC_VALID	714
5.130.1.22	MC_CSC_WP_OFFSET	714
5.130.1.23	MC_CSR	714
5.130.1.24	MC_CSR_VALID	714
5.130.1.25	MC_POC	714
5.130.1.26	MC_POC_EN_BW_OFFSET	714
5.130.1.27	MC_POC_EN_BW_WIDTH	714
5.130.1.28	MC_POC_EN_MEMTYPE_OFFSET	714
5.130.1.29	MC_POC_EN_MEMTYPE_WIDTH	714
5.130.1.30	MC_POC_VALID	714
5.130.1.31	MC_TMS	714
5.130.1.32	MC_TMS_ASYNC_TRDV_OFFSET	714
5.130.1.33	MC_TMS_ASYNC_TRDV_WIDTH	714
5.130.1.34	MC_TMS_ASYNC_TRDZ_OFFSET	714
5.130.1.35	MC_TMS_ASYNC_TRDZ_WIDTH	714
5.130.1.36	MC_TMS_ASYNC_TWD_OFFSET	714
5.130.1.37	MC_TMS_ASYNC_TWD_WIDTH	714
5.130.1.38	MC_TMS_ASYNC_TWPW_OFFSET	714
5.130.1.39	MC_TMS_ASYNC_TWPW_WIDTH	714
5.130.1.40	MC_TMS_ASYNC_TWWD_OFFSET	714
5.130.1.41	MC_TMS_ASYNC_TWWD_WIDTH	714
5.130.1.42	MC_TMS_ASYNC_VALID	714
5.130.1.43	MC_TMS_SDRAM_BL_OFFSET	714
5.130.1.44	MC_TMS_SDRAM_BL_WIDTH	714
5.130.1.45	MC_TMS_SDRAM_BT_OFFSET	714
5.130.1.46	MC_TMS_SDRAM_CL_OFFSET	714
5.130.1.47	MC_TMS_SDRAM_CL_WIDTH	714

5.130.1.48	MC_TMS_SDRAM_OM_OFFSET	714
5.130.1.49	MC_TMS_SDRAM_OM_WIDTH	714
5.130.1.50	MC_TMS_SDRAM_TRCD_OFFSET	714
5.130.1.51	MC_TMS_SDRAM_TRCD_WIDTH	714
5.130.1.52	MC_TMS_SDRAM_TRFC_OFFSET	714
5.130.1.53	MC_TMS_SDRAM_TRFC_WIDTH	714
5.130.1.54	MC_TMS_SDRAM_TRP_OFFSET	714
5.130.1.55	MC_TMS_SDRAM_TRP_WIDTH	714
5.130.1.56	MC_TMS_SDRAM_TWR_OFFSET	714
5.130.1.57	MC_TMS_SDRAM_TWR_WIDTH	714
5.130.1.58	MC_TMS_SDRAM_VALID	714
5.130.1.59	MC_TMS_SDRAM_WBL_OFFSET	714
5.130.1.60	MC_TMS_SSRAM_VALID	714
5.130.1.61	MC_TMS_SYNC_TRDV_OFFSET	714
5.130.1.62	MC_TMS_SYNC_TRDV_WIDTH	714
5.130.1.63	MC_TMS_SYNC_TRDZ_OFFSET	714
5.130.1.64	MC_TMS_SYNC_TRDZ_WIDTH	714
5.130.1.65	MC_TMS_SYNC_TTO_OFFSET	714
5.130.1.66	MC_TMS_SYNC_TTO_WIDTH	714
5.130.1.67	MC_TMS_SYNC_TWR_OFFSET	714
5.130.1.68	MC_TMS_SYNC_TWR_WIDTH	714
5.130.1.69	MC_TMS_SYNC_VALID	714
5.130.1.70	MC_TMS_VALID	714
5.130.1.71	IN_CE	714
5.130.2	Function Documentation	714
5.130.2.1	DEFAULT_DEBUG_CHANNEL	714
5.130.2.2	mc_baseaddr	714
5.130.2.3	mc_done	714
5.130.2.4	mc_enabled	715
5.130.2.5	mc_index	715
5.130.2.6	mc_poc	715
5.130.2.7	mc_read_word	715
5.130.2.8	mc_reg_mem_area	715
5.130.2.9	mc_reset	715
5.130.2.10	mc_sec_end	716
5.130.2.11	lmc_sec_start	716

5.130.2.12	mc_status	716
5.130.2.13	mc_write_word	716
5.130.2.14	reg_mc_sec	717
5.130.2.15	set_csc_tms	717
5.130.3	Variable Documentation	717
5.130.3.1	mc_areas	717
5.130.3.2	mcs	717
5.131	peripheral/mc.h File Reference	718
5.131.1	Function Documentation	718
5.131.1.1	mc_done	718
5.131.1.2	mc_reg_mem_area	719
5.131.1.3	reg_mc_sec	719
5.132	peripheral/memory.h File Reference	720
5.132.1	Function Documentation	721
5.132.1.1	reg_memory_sec	721
5.133	peripheral/ps2kbd.c File Reference	722
5.133.1	Define Documentation	725
5.133.1.1	KBD_BAUD_RATE	725
5.133.1.2	KBD_CCMD_DKI	725
5.133.1.3	KBD_CCMD_EKI	725
5.133.1.4	KBD_CCMD_RCB	725
5.133.1.5	KBD_CCMD_ST1	725
5.133.1.6	KBD_CCMD_ST2	725
5.133.1.7	KBD_CCMD_WCB	725
5.133.1.8	KBD_CCMDBYTE_EN	725
5.133.1.9	KBD_CCMDBYTE_EN2	725
5.133.1.10	KBD_CCMDBYTE_INT	725
5.133.1.11	KBD_CCMDBYTE_INT2	725
5.133.1.12	KBD_CCMDBYTE_SYS	725
5.133.1.13	KBD_CCMDBYTE_XLAT	725
5.133.1.14	KBD_CTRL	725
5.133.1.15	KBD_DATA	725
5.133.1.16	KBD_KCMD_DK	725
5.133.1.17	KBD_KCMD_ECHO	725
5.133.1.18	KBD_KCMD_EK	725
5.133.1.19	KBD_KCMD_RST	725

5.133.1.20	KBD_KCMD_SRL	725
5.133.1.21	KBD_KRESP_ACK	725
5.133.1.22	KBD_KRESP_ECHO	725
5.133.1.23	KBD_KRESP_RSTOK	725
5.133.1.24	KBD_MAX_BUF	725
5.133.1.25	KBD_SPACE	725
5.133.1.26	KBD_STATUS_A2	725
5.133.1.27	KBD_STATUS_IBF	725
5.133.1.28	KBD_STATUS_INH	725
5.133.1.29	KBD_STATUS_MOBF	725
5.133.1.30	KBD_STATUS_OBF	725
5.133.1.31	KBD_STATUS_PERR	725
5.133.1.32	KBD_STATUS_SYS	725
5.133.1.33	KBD_STATUS_TO	725
5.133.2	Function Documentation	725
5.133.2.1	kbd_baseaddr	725
5.133.2.2	kbd_enabled	725
5.133.2.3	kbd_info	725
5.133.2.4	kbd_irq	725
5.133.2.5	kbd_job	725
5.133.2.6	kbd_put	726
5.133.2.7	kbd_read8	726
5.133.2.8	kbd_reset	726
5.133.2.9	kbd_rxfile	726
5.133.2.10	kbd_sec_end	727
5.133.2.11	kbd_sec_start	727
5.133.2.12	kbd_write8	727
5.133.2.13	reg_kbd_sec	728
5.133.2.14	scan_decode	728
5.133.3	Variable Documentation	728
5.133.3.1	code	728
5.133.3.2	scan_table	728
5.133.3.3	shift	728
5.134	peripheral/ps2kbd.h File Reference	729
5.134.1	Function Documentation	729
5.134.1.1	reg_kbd_sec	729

5.135peripheral/vga.c File Reference	730
5.135.1 Define Documentation	732
5.135.1.1 VGA_ADDR_SPACE	732
5.135.1.2 VGA_CLUTA	732
5.135.1.3 VGA_CLUTB	732
5.135.1.4 VGA_CTRL	732
5.135.1.5 VGA_CTRL_CD	732
5.135.1.6 VGA_CTRL_PC	732
5.135.1.7 VGA_CTRL_VEN	732
5.135.1.8 VGA_HTIM	732
5.135.1.9 VGA_HVLEN	732
5.135.1.10VGA_MASK	732
5.135.1.11VGA_STAT	732
5.135.1.12VGA_VBARA	732
5.135.1.13VGA_VBARB	732
5.135.1.14VGA_VTIM	732
5.135.2 Function Documentation	732
5.135.2.1 reg_vga_sec	732
5.135.2.2 vga_baseaddr	733
5.135.2.3 vga_dump_image	733
5.135.2.4 vga_enabled	733
5.135.2.5 vga_filename	733
5.135.2.6 vga_irq	734
5.135.2.7 vga_job	734
5.135.2.8 vga_read32	734
5.135.2.9 vga_refresh_rate	734
5.135.2.10vga_reset	734
5.135.2.11vga_sec_end	735
5.135.2.12vga_sec_start	735
5.135.2.13vga_write32	735
5.136peripheral/vga.h File Reference	736
5.136.1 Function Documentation	736
5.136.1.1 reg_vga_sec	736
5.137pic/pic.c File Reference	737
5.137.1 Function Documentation	738
5.137.1.1 clear_interrupt	738

5.137.1.2	DEFAULT_DEBUG_CHANNEL	738
5.137.1.3	pic_edge_trigger	738
5.137.1.4	pic_enabled	738
5.137.1.5	pic_ints_en	739
5.137.1.6	pic_rep_int	739
5.137.1.7	pic_reset	739
5.137.1.8	reg_pic_sec	739
5.137.1.9	report_interrupt	740
5.137.2	Variable Documentation	740
5.137.2.1	pic_state	740
5.137.2.2	pic_state_int	740
5.138	pic/pic.h File Reference	741
5.138.1	Function Documentation	741
5.138.1.1	clear_interrupt	741
5.138.1.2	pic_ints_en	741
5.138.1.3	pic_reset	741
5.138.1.4	reg_pic_sec	741
5.138.1.5	report_interrupt	742
5.139	pm/pm.c File Reference	743
5.139.1	Function Documentation	743
5.139.1.1	pm_enabled	743
5.139.1.2	pm_reset	743
5.139.1.3	reg_pm_sec	744
5.140	pm/pm.h File Reference	745
5.140.1	Function Documentation	745
5.140.1.1	pm_reset	745
5.140.1.2	reg_pm_sec	745
5.141	port/isblank.c File Reference	746
5.141.1	Function Documentation	746
5.141.1.1	isblank	746
5.142	port/port.h File Reference	747
5.142.1	Define Documentation	747
5.142.1.1	PRIx16	747
5.142.1.2	PRIx8	747
5.142.2	Function Documentation	747
5.142.2.1	isblank	747

5.142.2.2 strndup	747
5.143port/strndup.c File Reference	748
5.143.1 Function Documentation	748
5.143.1.1 strndup	748
5.144profiler.c File Reference	749
5.144.1 Define Documentation	750
5.144.1.1 MAX_STACK	750
5.144.2 Function Documentation	750
5.144.2.1 main_profiler	750
5.144.2.2 prof_acquire	750
5.144.2.3 prof_print	751
5.144.2.4 prof_set	751
5.144.3 Variable Documentation	751
5.144.3.1 cumulative	751
5.144.3.2 fprof	751
5.144.3.3 maxstack	751
5.144.3.4 nfunccalls	751
5.144.3.5 nstack	751
5.144.3.6 ntotcalls	751
5.144.3.7 prof_cycles	751
5.144.3.8 prof_func	751
5.144.3.9 prof_nfuncs	751
5.144.3.10quiet	751
5.144.3.11stack	752
5.145profiler.h File Reference	753
5.145.1 Define Documentation	753
5.145.1.1 MAX_FUNCS	753
5.145.2 Function Documentation	753
5.145.2.1 main_profiler	753
5.145.2.2 prof_acquire	754
5.145.2.3 prof_set	754
5.145.3 Variable Documentation	754
5.145.3.1 prof_cycles	754
5.145.3.2 prof_func	754
5.145.3.3 prof_nfuncs	754
5.146sim-cmd.c File Reference	755

5.146.1 Function Documentation	757
5.146.1.1 check_insn_exec	757
5.146.1.2 handle_sim_command	758
5.146.1.3 print_insn_exec	759
5.146.1.4 reenter_int	759
5.146.1.5 reg_sim_stat	760
5.146.1.6 sim_cmd_break	760
5.146.1.7 sim_cmd_breaks	760
5.146.1.8 sim_cmd_cm	760
5.146.1.9 sim_cmd_cuc	762
5.146.1.10sim_cmd_de	763
5.146.1.11sim_cmd_debug	763
5.146.1.12sim_cmd_dh	763
5.146.1.13sim_cmd_dm	764
5.146.1.14sim_cmd_dv	764
5.146.1.15sim_cmd_help	764
5.146.1.16sim_cmd_hist	765
5.146.1.17sim_cmd_info	765
5.146.1.18sim_cmd_mprofile	765
5.146.1.19sim_cmd_pc	766
5.146.1.20sim_cmd_pm	766
5.146.1.21sim_cmd_pr	766
5.146.1.22sim_cmd_profile	766
5.146.1.23sim_cmd_quit	767
5.146.1.24sim_cmd_r	767
5.146.1.25sim_cmd_reset	767
5.146.1.26sim_cmd_run	768
5.146.1.27sim_cmd_set	768
5.146.1.28sim_cmd_setdbch	768
5.146.1.29sim_cmd_stall	768
5.146.1.30sim_cmd_stats	769
5.146.1.31sim_cmd_trace	769
5.146.1.32sim_cmd_unstall	769
5.146.1.33strip_space	770
5.146.2 Variable Documentation	770
5.146.2.1 sim_commands	770

5.146.2.2	sim_stats	770
5.146.2.3	to_insn_num	770
5.147	sim-cmd.h File Reference	771
5.147.1	Function Documentation	771
5.147.1.1	handle_sim_command	771
5.147.1.2	reg_sim_stat	771
5.148	sim-config.c File Reference	772
5.148.1	Define Documentation	774
5.148.1.1	MERROR	774
5.148.1.2	WARNING	774
5.148.2	Function Documentation	774
5.148.2.1	base_include	774
5.148.2.2	DEFAULT_DEBUG_CHANNEL	774
5.148.2.3	get_paramt_str	774
5.148.2.4	init_defconfig	774
5.148.2.5	parse_args	774
5.148.2.6	print_config	775
5.148.2.7	read_script_file	775
5.148.2.8	reg_config_param	775
5.148.2.9	reg_config_sec	777
5.148.2.10	reg_config_secs	777
5.148.2.11	reg_sim_sec	778
5.148.2.12	set_config	778
5.148.2.13	set_config_command	779
5.148.2.14	sim_clkcycle	779
5.148.2.15	sim_debug	779
5.148.2.16	sim_exe_log	779
5.148.2.17	sim_exe_log_end	779
5.148.2.18	sim_exe_log_fn	779
5.148.2.19	sim_exe_log_marker	779
5.148.2.20	sim_exe_log_start	779
5.148.2.21	sim_exe_log_type	779
5.148.2.22	sim_history	780
5.148.2.23	sim_mprof_fn	780
5.148.2.24	sim_mprofile	780
5.148.2.25	sim_prof_fn	780

5.148.2.26	sim_profile	780
5.148.2.27	sim_verbose	780
5.148.2.28	switch_param	780
5.148.3	Variable Documentation	780
5.148.3.1	config	780
5.148.3.2	cur_section	780
5.148.3.3	runtime	780
5.148.3.4	sections	780
5.149	sim-config.h File Reference	781
5.149.1	Define Documentation	782
5.149.1.1	CHECK_INT_TIME	782
5.149.1.2	CONFIG_ERROR	782
5.149.1.3	EXE_LOG_HARDWARE	782
5.149.1.4	EXE_LOG_SIMPLE	782
5.149.1.5	EXE_LOG_SOFTWARE	782
5.149.1.6	MAX_SBUF_LEN	782
5.149.1.7	PRINTF	782
5.149.1.8	STR_SIZE	782
5.149.2	Enumeration Type Documentation	782
5.149.2.1	param_t	782
5.149.3	Function Documentation	783
5.149.3.1	init_defconfig	783
5.149.3.2	parse_args	783
5.149.3.3	print_config	783
5.149.3.4	reg_config_param	784
5.149.3.5	reg_config_sec	786
5.149.3.6	reg_config_secs	786
5.149.3.7	set_config_command	787
5.149.4	Variable Documentation	787
5.149.4.1	config	787
5.149.4.2	cur_section	787
5.149.4.3	do_stats	787
5.149.4.4	runtime	787
5.150	support/dbchs.h File Reference	788
5.151	support/debug.c File Reference	789
5.151.1	Define Documentation	790

5.151.1.1	<code>__ORSIM_NO_DEC_DBCH</code>	790
5.151.1.2	<code>DECLARE_DEBUG_CHANNEL</code>	790
5.151.1.3	<code>DECLARE_DEBUG_CHANNEL</code>	790
5.151.2	Function Documentation	790
5.151.2.1	<code>debug</code>	790
5.151.2.2	<code>orsim_dbcl_set</code>	790
5.151.2.3	<code>orsim_dbcl_set_name</code>	790
5.151.2.4	<code>orsim_dbg_log</code>	790
5.151.2.5	<code>parse_dbchs</code>	790
5.151.3	Variable Documentation	790
5.151.3.1	<code>__orsim_dbchs</code>	790
5.151.3.2	<code>debug_classes</code>	790
5.152	<code>support/debug.h</code> File Reference	791
5.152.1	Define Documentation	792
5.152.1.1	<code>__ORSIM_DBG_USE_FUNC</code>	792
5.152.1.2	<code>__ORSIM_DEBUG_LOG</code>	792
5.152.1.3	<code>__ORSIM_DPRINTF</code>	792
5.152.1.4	<code>__ORSIM_GET_DEBUGGING</code>	792
5.152.1.5	<code>__ORSIM_GET_DEBUGGING_ERR</code>	792
5.152.1.6	<code>__ORSIM_GET_DEBUGGING_FIXME</code>	792
5.152.1.7	<code>__ORSIM_GET_DEBUGGING_TRACE</code>	792
5.152.1.8	<code>__ORSIM_GET_DEBUGGING_WARN</code>	792
5.152.1.9	<code>DECLARE_DEBUG_CHANNEL</code>	792
5.152.1.10	<code>DEFAULT_DEBUG_CHANNEL</code>	792
5.152.1.11	<code>IERR</code>	793
5.152.1.12	<code>IERR_</code>	793
5.152.1.13	<code>IERR_ON</code>	793
5.152.1.14	<code>FIXME</code>	793
5.152.1.15	<code>FIXME_</code>	793
5.152.1.16	<code>FIXME_ON</code>	793
5.152.1.17	<code>TRACE</code>	793
5.152.1.18	<code>TRACE_</code>	793
5.152.1.19	<code>TRACE_ON</code>	793
5.152.1.20	<code>WARN</code>	793
5.152.1.21	<code>IWARN_</code>	793
5.152.1.22	<code>WARN_ON</code>	793

5.152.2 Enumeration Type Documentation	793
5.152.2.1 __ORSIM_DEBUG_CLASS	793
5.152.3 Function Documentation	793
5.152.3.1 debug	793
5.152.3.2 orsim_dbcl_set_name	794
5.152.3.3 orsim_dbg_log	794
5.152.3.4 parse_dbchs	794
5.153support/dumpverilog.c File Reference	795
5.153.1 Define Documentation	796
5.153.1.1 DISWIDTH	796
5.153.1.2 DW	796
5.153.1.3 DWQ	796
5.153.1.4 LABELEND_CHAR	796
5.153.1.5 OR1K_MEM_VERILOG_FOOTER	796
5.153.1.6 OR1K_MEM_VERILOG_HEADER	796
5.153.2 Function Documentation	797
5.153.2.1 dumphex	797
5.153.2.2 dumpverilog	797
5.154support/dumpverilog.h File Reference	798
5.154.1 Function Documentation	799
5.154.1.1 dumphex	799
5.154.1.2 dumpverilog	799
5.155support/misc.c File Reference	800
5.155.1 Function Documentation	800
5.155.1.1 is_power2	800
5.155.1.2 log2_int	800
5.156support/misc.h File Reference	801
5.156.1 Function Documentation	801
5.156.1.1 is_power2	801
5.156.1.2 log2_int	801
5.157support/profile.c File Reference	802
5.157.1 Function Documentation	802
5.157.1.1 mprofile	802
5.158support/profile.h File Reference	803
5.158.1 Define Documentation	804
5.158.1.1 MPROF_16	804

5.158.1.2	MPROF_32	804
5.158.1.3	MPROF_8	804
5.158.1.4	MPROF_FETCH	804
5.158.1.5	MPROF_READ	804
5.158.1.6	MPROF_WRITE	804
5.158.2	Function Documentation	804
5.158.2.1	mprofile	804
5.159	support/sched.c File Reference	805
5.159.1	Define Documentation	806
5.159.1.1	SCHED_HEAP_SIZE	806
5.159.1.2	SCHED_TIME_MAX	806
5.159.2	Function Documentation	806
5.159.2.1	DECLARE_DEBUG_CHANNEL	806
5.159.2.2	DEFAULT_DEBUG_CHANNEL	806
5.159.2.3	do_scheduler	806
5.159.2.4	sched_add	806
5.159.2.5	sched_find_remove	806
5.159.2.6	sched_guard	806
5.159.2.7	sched_init	806
5.159.2.8	sched_next_insn	806
5.159.2.9	sched_print_jobs	806
5.159.2.10	sched_reset	806
5.159.3	Variable Documentation	806
5.159.3.1	scheduler	806
5.160	support/sched.h File Reference	807
5.160.1	Define Documentation	807
5.160.1.1	SCHED_ADD	807
5.160.1.2	SCHED_FIND_REMOVE	807
5.160.2	Function Documentation	808
5.160.2.1	do_scheduler	808
5.160.2.2	sched_add	808
5.160.2.3	sched_find_remove	808
5.160.2.4	sched_init	808
5.160.2.5	sched_next_insn	808
5.160.2.6	sched_reset	808
5.160.3	Variable Documentation	808

5.160.3.1 scheduler	808
5.161 support/simprintf.c File Reference	809
5.161.1 Define Documentation	810
5.161.1.1 FMTLEN	810
5.161.1.2 STACK_ARGS	810
5.161.2 Function Documentation	810
5.161.2.1 DEFAULT_DEBUG_CHANNEL	810
5.161.2.2 simgetstr	810
5.161.2.3 simprintf	810
5.161.3 Variable Documentation	810
5.161.3.1 fmtstr	810
5.162 support/simprintf.h File Reference	811
5.162.1 Function Documentation	812
5.162.1.1 simprintf	812
5.163 tick/tick.c File Reference	813
5.163.1 Function Documentation	814
5.163.1.1 DEFAULT_DEBUG_CHANNEL	814
5.163.1.2 sched_timer_job	814
5.163.1.3 spr_read_ttc	814
5.163.1.4 spr_write_ttc	814
5.163.1.5 spr_write_ttmr	815
5.163.1.6 tick_one_shot	815
5.163.1.7 tick_raise_except	815
5.163.1.8 tick_reset	816
5.163.1.9 tick_restart	816
5.163.2 Variable Documentation	816
5.163.2.1 cycles_start	816
5.163.2.2 tick_count	816
5.164 tick/tick.h File Reference	817
5.164.1 Function Documentation	817
5.164.1.1 spr_read_ttc	817
5.164.1.2 spr_write_ttc	817
5.164.1.3 spr_write_ttmr	818
5.164.1.4 tick_reset	818
5.165 toplevel-mprofile.c File Reference	819
5.165.1 Function Documentation	821

5.165.1.1 main	821
5.166toplevel-profile.c File Reference	822
5.166.1 Function Documentation	824
5.166.1.1 main	824
5.167toplevel-support.c File Reference	825
5.167.1 Function Documentation	826
5.167.1.1 check_int	826
5.167.1.2 ctrl_c	827
5.167.1.3 reg_sim_reset	828
5.167.1.4 sim_done	828
5.167.1.5 sim_init	828
5.167.1.6 sim_reset	829
5.167.2 Variable Documentation	830
5.167.2.1 sim_reset_hooks	830
5.168toplevel-support.h File Reference	831
5.168.1 Function Documentation	831
5.168.1.1 check_int	831
5.168.1.2 ctrl_c	832
5.168.1.3 reg_sim_reset	832
5.168.1.4 sim_done	833
5.168.1.5 sim_init	833
5.168.1.6 sim_reset	834
5.169toplevel.c File Reference	836
5.169.1 Function Documentation	836
5.169.1.1 main	836
5.170vapi/vapi.c File Reference	839
5.170.1 Function Documentation	840
5.170.1.1 add_handler	840
5.170.1.2 DEFAULT_DEBUG_CHANNEL	840
5.170.1.3 find_handler	840
5.170.1.4 get_server_socket	841
5.170.1.5 handler_fits_id	841
5.170.1.6 read_packet	841
5.170.1.7 rebuild_fds	841
5.170.1.8 reg_vapi_sec	841
5.170.1.9 server_request	841

5.170.1.10	vapi_check	842
5.170.1.11	vapi_done	842
5.170.1.12	vapi_enabled	842
5.170.1.13	vapi_hide_device_id	842
5.170.1.14	vapi_init	842
5.170.1.15	vapi_install_handler	842
5.170.1.16	vapi_install_multi_handler	843
5.170.1.17	vapi_log_enabled	843
5.170.1.18	vapi_log_fn	843
5.170.1.19	vapi_num_unconnected	843
5.170.1.20	vapi_read_stream	843
5.170.1.21	vapi_request	843
5.170.1.22	vapi_send	844
5.170.1.23	vapi_server_port	844
5.170.1.24	vapi_write_log_file	844
5.170.1.25	vapi_write_stream	844
5.170.1.26	write_packet	844
5.170.2	Variable Documentation	844
5.170.2.1	fds	844
5.170.2.2	nfds	844
5.170.2.3	nhandlers	844
5.170.2.4	server_fd	844
5.170.2.5	serverIP	844
5.170.2.6	tcp_level	844
5.171	vapi/vapi.h File Reference	845
5.171.1	Define Documentation	845
5.171.1.1	VAPI_MAX_DEVID	845
5.171.2	Enumeration Type Documentation	845
5.171.2.1	VAPI_COMMAND	845
5.171.3	Function Documentation	846
5.171.3.1	reg_vapi_sec	846
5.171.3.2	vapi_check	846
5.171.3.3	vapi_done	846
5.171.3.4	vapi_init	847
5.171.3.5	vapi_install_handler	847
5.171.3.6	vapi_install_multi_handler	847

5.171.3.7 vapi_num_unconnected	847
5.171.3.8 vapi_send	847
5.171.3.9 vapi_write_log_file	848

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

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

<code>_csm_list</code>	11
<code>_cuc_func</code>	13
<code>_dep_list_t</code>	15
<code>archf</code>	16
<code>ata_device</code>	17
<code>ata_devices</code>	22
<code>ata_host</code>	23
<code>bff</code>	26
<code>BMP_HEADER</code>	27
<code>bpb_entry</code>	28
<code>bpbstat</code>	29
<code>branchstat</code>	30
<code>breakpoint_entry</code>	31
<code>btic_entry</code>	32
<code>bticstat</code>	33
<code>cachestats_entry</code>	34
<code>channel</code>	35
<code>channel_factory</code>	36
<code>channel_ops</code>	37
<code>COFF_AOUTHDR</code>	38
<code>COFF_auxent</code>	39
<code>COFF_filehdr</code>	43
<code>COFF_lineno</code>	44
<code>COFF_reloc</code>	45
<code>COFF_scnhdr</code>	46
<code>COFF_slib</code>	47
<code>COFF_syment</code>	48
<code>config</code>	49
<code>config::pic</code>	54
<code>config_param</code>	55
<code>config_section</code>	56
<code>cpu_state</code>	57
<code>cuc_bb</code>	59

cuc_conv	61
cuc_insn	62
cuc_known_insn	63
cuc_shared_item	64
cuc_timing_table	65
cuc_timings	66
dc_set	67
dev_16450	68
dev_generic	73
dev_memarea	75
dma_channel	77
dma_controller	80
dmmu	82
dmmustats_entry	84
dstats_entry	85
dyn_page	86
dynamic	87
elf32_hdr	88
elf32_note	90
elf32_phdr	91
elf32_rel	92
elf32_rela	93
elf32_shdr	94
elf32_sym	95
Elf64_Dyn	96
elf64_hdr	97
elf64_note	99
elf64_phdr	100
elf64_rel	101
elf64_rela	102
elf64_shdr	103
elf64_sym	104
elf_obj	105
eth_device	107
ether_addr	112
ether_header	113
fb_state	114
fd_channel	116
file_channel	117
fstats_entry	118
func_struct	119
gpio_device	120
hist_exec	122
ic	123
immu	125
immustats_entry	127
INFOHEADER	128
iqueue_entry	129
jtr_chain_message	130
jtr_chain_response	131
jtr_failure_response	132
jtr_read_block_message	133
jtr_read_block_response	134
jtr_read_message	135

jtr_read_response	136
jtr_write_block_message	137
jtr_write_block_response	138
jtr_write_message	139
jtr_write_response	140
kbd_state	141
label_entry	142
mc	143
mc_area	145
mem_config	146
mem_ops	148
memory_hash	150
mprofentry_struct	151
mstats_entry	152
op_queue	153
param_val	155
raw_stats	156
reloc	157
runtime	158
sched_entry	162
scheduler_struct	163
sim_command	164
sim_reset_hook	165
sim_stat	166
spr_bit_def	167
spr_def	168
sstats_entry	169
stack_struct	170
tcp_channel	171
tty_channel	172
vapi_handler	173
vga_state	174
xterm_channel	176

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

cpu-config.c	200
cpu-config.h	204
libtoplevel.c	532
mainpage	538
mprofiler.c	556
mprofiler.h	560
or1ksim.h	561
profiler.c	749
profiler.h	753
sim-cmd.c	755
sim-cmd.h	771
sim-config.c	772
sim-config.h	781
toplevel-mprofile.c	819
toplevel-profile.c	822
toplevel-support.c	825
toplevel-support.h	831
toplevel.c	836
bpb/branch-predict.c	177
bpb/branch-predict.h	182
cache/dcache-model.c	184
cache/dcache-model.h	189
cache/icache-model.c	192
cache/icache-model.h	197
cpu/common/abstract.c	205
cpu/common/abstract.h	222
cpu/common/coff.h	237
cpu/common/elf.h	244
cpu/common/execute.h	254
cpu/common/labels.c	261
cpu/common/labels.h	263
cpu/common/parse.c	265
cpu/common/parse.h	271

cpu/common/stats.c	273
cpu/common/stats.h	277
cpu/common/trace.c	279
cpu/common/trace.h	281
cpu/or1k/arch.h	283
cpu/or1k/except.c	285
cpu/or1k/except.h	287
cpu/or1k/spr-defs.h	291
cpu/or1k/spr-dump.c	301
cpu/or1k/spr-dump.h	313
cpu/or1k/sprs.c	314
cpu/or1k/sprs.h	317
cpu/or32/common_i386.h	319
cpu/or32/def_op_t.h	320
cpu/or32/dyn32_defs.h	322
cpu/or32/dyn_rec.c	323
cpu/or32/dyn_rec.h	347
cpu/or32/dyngen.c	352
cpu/or32/dyngen.h	355
cpu/or32/dyngen_elf.c	356
cpu/or32/dyngen_i386.c	358
cpu/or32/execute.c	360
cpu/or32/generate.c	370
cpu/or32/i386_regs.h	373
cpu/or32/insnset.c	374
cpu/or32/op.c	387
cpu/or32/op_1t.h	402
cpu/or32/op_1t_op.h	403
cpu/or32/op_2t.h	404
cpu/or32/op_2t_op.h	405
cpu/or32/op_3t.h	406
cpu/or32/op_3t_op.h	407
cpu/or32/op_arith_op.h	408
cpu/or32/op_comp_op.h	409
cpu/or32/op_extend_op.h	410
cpu/or32/op_ff1_op.h	411
cpu/or32/op_i386.h	412
cpu/or32/op_lwhb_op.h	413
cpu/or32/op_mac_op.h	414
cpu/or32/op_mftspr_op.h	415
cpu/or32/op_support.c	416
cpu/or32/op_support.h	419
cpu/or32/op_swhb_op.h	421
cpu/or32/op_t_reg_mov_op.h	422
cpu/or32/or32.c	426
cpu/or32/rec_i386.h	433
cpu/or32/sched_i386.h	434
cpu/or32/simpl32_defs.h	435
cuc/adv.c	436
cuc/bb.c	438
cuc/cuc.c	444
cuc/cuc.h	460
cuc/insn.c	478
cuc/insn.h	484

cuc/load.c	490
cuc/memory.c	495
cuc/timings.c	503
cuc/verilog.c	506
cuc/verilog.h	509
debug/debug-unit.c	510
debug/debug-unit.h	519
debug/gdb.h	524
debug/gdbcomm.c	526
debug/gdbcomm.h	530
mmu/dmmu.c	539
mmu/dmmu.h	545
mmu/immu.c	548
mmu/immu.h	553
peripheral/16450.c	564
peripheral/16450.h	586
peripheral/atacmd.h	590
peripheral/atadevice.c	594
peripheral/atadevice.h	598
peripheral/atadevice_cmdi.c	603
peripheral/atadevice_cmdi.h	608
peripheral/atahost.c	613
peripheral/atahost.h	623
peripheral/atahost_define.h	629
peripheral/crc32.c	648
peripheral/crc32.h	649
peripheral/dma-defs.h	650
peripheral/dma.c	657
peripheral/dma.h	664
peripheral/eth.c	666
peripheral/eth.h	682
peripheral/fb.c	684
peripheral/fb.h	690
peripheral/fields.h	691
peripheral/generic.c	693
peripheral/generic.h	699
peripheral/gpio.c	701
peripheral/gpio.h	709
peripheral/mc.c	710
peripheral/mc.h	718
peripheral/memory.c	498
peripheral/memory.h	720
peripheral/ps2kbd.c	722
peripheral/ps2kbd.h	729
peripheral/vga.c	730
peripheral/vga.h	736
peripheral/channels/channel.c	630
peripheral/channels/channel.h	632
peripheral/channels/fd.c	633
peripheral/channels/fd.h	635
peripheral/channels/file.c	636
peripheral/channels/file.h	638
peripheral/channels/generic.c	698
peripheral/channels/generic.h	700

peripheral/channels/tcp.c	639
peripheral/channels/tcp.h	641
peripheral/channels/tty.c	642
peripheral/channels/tty.h	644
peripheral/channels/xterm.c	645
peripheral/channels/xterm.h	647
pic/pic.c	737
pic/pic.h	741
pm/pm.c	743
pm/pm.h	745
port/isblank.c	746
port/port.h	747
port/strndup.c	748
support/dbchs.h	788
support/debug.c	789
support/debug.h	791
support/dumpverilog.c	795
support/dumpverilog.h	798
support/misc.c	800
support/misc.h	801
support/profile.c	802
support/profile.h	803
support/sched.c	805
support/sched.h	807
support/simprintf.c	809
support/simprintf.h	811
tick/tick.c	813
tick/tick.h	817
vapi/vapi.c	839
vapi/vapi.h	845

Chapter 4

Data Structure Documentation

4.1 `_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`

4.1.1 Field Documentation

4.1.1.1 `int _csm_list::ref`

4.1.1.2 `int _csm_list::cnt`

4.1.1.3 `int _csm_list::cmovs`

4.1.1.4 `double _csm_list::size`

4.1.1.5 `double _csm_list::osize`

4.1.1.6 `int _csm_list::cmatch`

4.1.1.7 `int _csm_list::dead`

4.1.1.8 `int _csm_list::ninsn`

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

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

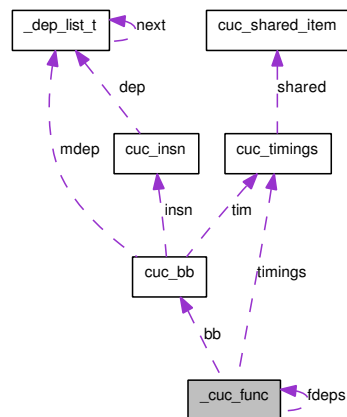
The documentation for this struct was generated from the following file:

- [cuc/cuc.h](#)

4.2 `_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`

4.2.1 Field Documentation

- 4.2.1.1 `int _cuc_func::num_bb`
- 4.2.1.2 `cuc_bb _cuc_func::bb[MAX_BB]`
- 4.2.1.3 `int _cuc_func::saved_regs[MAX_REGS]`
- 4.2.1.4 `int _cuc_func::lur[MAX_REGS]`
- 4.2.1.5 `int _cuc_func::used_regs[MAX_REGS]`
- 4.2.1.6 `int _cuc_func::nmsched`
- 4.2.1.7 `int _cuc_func::msched[MAX_INSNS]`
- 4.2.1.8 `int _cuc_func::mtype[MAX_INSNS]`
- 4.2.1.9 `int _cuc_func::num_init_bb`
- 4.2.1.10 `int* _cuc_func::init_bb_reloc`
- 4.2.1.11 `int _cuc_func::orig_time`
- 4.2.1.12 `int _cuc_func::num_runs`
- 4.2.1.13 `cuc_timings _cuc_func::timings`
- 4.2.1.14 `unsigned long _cuc_func::start_addr`
- 4.2.1.15 `unsigned long _cuc_func::end_addr`
- 4.2.1.16 `int _cuc_func::memory_order`
- 4.2.1.17 `int _cuc_func::nfdeps`
- 4.2.1.18 `struct _cuc_func** _cuc_func::fdeps` [read]
- 4.2.1.19 `int _cuc_func::tmp`

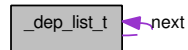
The documentation for this struct was generated from the following file:

- [cuc/cuc.h](#)

4.3 `_dep_list_t` Struct Reference

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

Collaboration diagram for `_dep_list_t`:



Data Fields

- unsigned long [ref](#)
- struct `_dep_list_t` * [next](#)

4.3.1 Field Documentation

4.3.1.1 unsigned long `_dep_list_t::ref`

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

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

- [cuc/cuc.h](#)

4.4 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)

4.4.1 Field Documentation

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

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

4.4.1.3 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](#)

4.5 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
```


4.5.1 Field Documentation

4.5.1.1 void* ata_device::host

4.5.1.2 int ata_device::dev

4.5.1.3 int ata_device::pio_mode

4.5.1.4 int ata_device::dma_mode

4.5.1.5 uint16_t ata_device::dbuf[4096]

4.5.1.6 uint16_t* ata_device::dbuf_ptr

4.5.1.7 uint16_t ata_device::dbuf_cnt

4.5.1.8 int ata_device::state

4.5.1.9 unsigned int ata_device::heads_per_cylinder

4.5.1.10 unsigned int ata_device::sectors_per_track

4.5.1.11 uint32_t ata_device::lba

4.5.1.12 int ata_device::nr_sect

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

4.5.1.14 struct { ... } ata_device::internals

4.5.1.15 uint8_t ata_device::command

4.5.1.16 uint8_t ata_device::cylinder_low

4.5.1.17 uint8_t ata_device::cylinder_high

4.5.1.18 uint8_t ata_device::device_control

4.5.1.19 uint8_t ata_device::device_head

4.5.1.20 uint8_t ata_device::error

4.5.1.21 uint8_t ata_device::features

4.5.1.22 uint8_t ata_device::sector_count

4.5.1.23 uint8_t ata_device::sector_number

4.5.1.24 uint8_t ata_device::status

4.5.1.25 uint16_t ata_device::dataport_i

4.5.1.26 struct { ... } ata_device::regs

4.5.1.27 int ata_device::iordy

4.5.1.28 int ata_device::intrq

4.5.1.29 int ata_device::dmarq

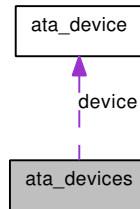
4.5.1.30 int ata_device::ndiagi

- [peripheral/atadevice.h](#)

4.6 ata_devices Struct Reference

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

Collaboration diagram for ata_devices:



Data Fields

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

4.6.1 Field Documentation

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

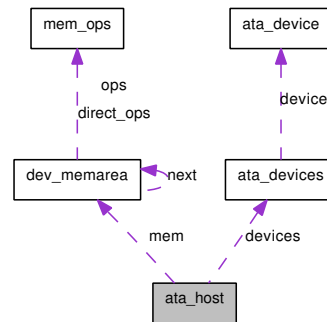
The documentation for this struct was generated from the following file:

- [peripheral/atadevice.h](#)

4.7 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

4.7.1 Field Documentation

- 4.7.1.1 `int ata_host::enabled`
- 4.7.1.2 `oraddr_t ata_host::baseaddr`
- 4.7.1.3 `struct dev_memarea* ata_host::mem` [read]
- 4.7.1.4 `int ata_host::irq`
- 4.7.1.5 `int ata_host::dev_id`
- 4.7.1.6 `int ata_host::rev`
- 4.7.1.7 `int ata_host::dev_sel`
- 4.7.1.8 `uint8_t ata_host::pio_mode0_t1`
- 4.7.1.9 `uint8_t ata_host::pio_mode0_t2`
- 4.7.1.10 `uint8_t ata_host::pio_mode0_t4`
- 4.7.1.11 `uint8_t ata_host::pio_mode0_teoc`
- 4.7.1.12 `uint8_t ata_host::dma_mode0_tm`
- 4.7.1.13 `uint8_t ata_host::dma_mode0_td`
- 4.7.1.14 `uint8_t ata_host::dma_mode0_teoc`
- 4.7.1.15 `uint32_t ata_host::ctrl`
- 4.7.1.16 `uint32_t ata_host::stat`
- 4.7.1.17 `uint32_t ata_host::pctr`
- 4.7.1.18 `uint32_t ata_host::pftr0`
- 4.7.1.19 `uint32_t ata_host::pftr1`
- 4.7.1.20 `uint32_t ata_host::dtr0`
- 4.7.1.21 `uint32_t ata_host::dtr1`
- 4.7.1.22 `uint32_t ata_host::txb`
- 4.7.1.23 `uint32_t ata_host::rxb`
- 4.7.1.24 `struct { ... } ata_host::regs`
- 4.7.1.25 `struct ata_devices ata_host::devices` [read]

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

- [peripheral/atahost.h](#)

4.8 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](#))

4.8.1 Field Documentation

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

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

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

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

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

4.8.1.6 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](#)

4.9 BMP_HEADER Struct Reference

Data Fields

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

4.9.1 Field Documentation

4.9.1.1 unsigned short int BMP_HEADER::type

4.9.1.2 unsigned int BMP_HEADER::size

4.9.1.3 unsigned short int BMP_HEADER::reserved1

4.9.1.4 unsigned short int BMP_HEADER::reserved2

4.9.1.5 unsigned int BMP_HEADER::offset

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

- [peripheral/vga.c](#)

4.10 bpb_entry Struct Reference

Data Fields

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

4.10.1 Field Documentation

4.10.1.1 oraddr_t bpb_entry::addr

4.10.1.2 int bpb_entry::taken

4.10.1.3 int bpb_entry::lru

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

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

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

4.11 bpbstat Struct Reference

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

Data Fields

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

4.11.1 Field Documentation

4.11.1.1 int `bpbstat::hit`

4.11.1.2 int `bpbstat::miss`

4.11.1.3 int `bpbstat::correct`

4.11.1.4 int `bpbstat::incorrect`

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

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

4.12 branchstat Struct Reference

Data Fields

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

4.12.1 Field Documentation

4.12.1.1 int branchstat::taken

4.12.1.2 int branchstat::nottaken

4.12.1.3 int branchstat::forward

4.12.1.4 int branchstat::backward

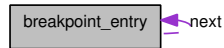
The documentation for this struct was generated from the following file:

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

4.13 breakpoint_entry Struct Reference

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

Collaboration diagram for breakpoint_entry:



Data Fields

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

4.13.1 Detailed Description

Structure representing a breakpoint

4.13.2 Field Documentation

4.13.2.1 [oraddr_t](#) breakpoint_entry::addr

4.13.2.2 [struct breakpoint_entry*](#) breakpoint_entry::next [read]

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

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

4.14 btic_entry Struct Reference

Data Fields

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

4.14.1 Field Documentation

4.14.1.1 `oraddr_t btic_entry::addr`

4.14.1.2 `int btic_entry::lru`

4.14.1.3 `char* btic_entry::insn`

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

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

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

4.15 bticstat Struct Reference

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

Data Fields

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

4.15.1 Field Documentation

4.15.1.1 int bticstat::hit

4.15.1.2 int bticstat::miss

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

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

4.16 cachestats_entry Struct Reference

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

Data Fields

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

4.16.1 Field Documentation

4.16.1.1 int cachestats_entry::readhit

4.16.1.2 int cachestats_entry::readmiss

4.16.1.3 int cachestats_entry::writehit

4.16.1.4 int cachestats_entry::writemiss

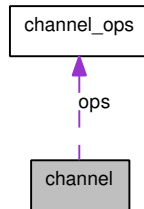
The documentation for this struct was generated from the following file:

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

4.17 channel Struct Reference

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

Collaboration diagram for channel:



Data Fields

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

4.17.1 Detailed Description

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

4.17.2 Field Documentation

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

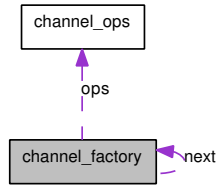
4.17.2.2 void* [channel::data](#)

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

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

4.18 channel_factory Struct Reference

Collaboration diagram for channel_factory:



Data Fields

- `const char * name`
- `struct channel_ops * ops`
- `struct channel_factory * next`

4.18.1 Field Documentation

4.18.1.1 `const char* channel_factory::name`

4.18.1.2 `struct channel_ops* channel_factory::ops` [read]

4.18.1.3 `struct channel_factory* channel_factory::next` [read]

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

- `peripheral/channels/channel.c`

4.19 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 *****)

4.19.1 Detailed Description

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

4.19.2 Field Documentation

4.19.2.1 void(***** [channel_ops::init](#))(const char *****)

4.19.2.2 int(***** [channel_ops::open](#))(void *****)

4.19.2.3 void(***** [channel_ops::close](#))(void *****)

4.19.2.4 int(***** [channel_ops::read](#))(void *****, char *****, int)

4.19.2.5 int(***** [channel_ops::write](#))(void *****, const char *****, int)

4.19.2.6 void(***** [channel_ops::free](#))(void *****)

4.19.2.7 int(***** [channel_ops::isok](#))(void *****)

4.19.2.8 char(***** [channel_ops::status](#))(void *****)

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

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

4.20 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]

4.20.1 Field Documentation

4.20.1.1 char COFF_AOUTHDR::magic[2]

4.20.1.2 char COFF_AOUTHDR::vstamp[2]

4.20.1.3 char COFF_AOUTHDR::tsize[4]

4.20.1.4 char COFF_AOUTHDR::dsize[4]

4.20.1.5 char COFF_AOUTHDR::bsize[4]

4.20.1.6 char COFF_AOUTHDR::entry[4]

4.20.1.7 char COFF_AOUTHDR::text_start[4]

4.20.1.8 char COFF_AOUTHDR::data_start[4]

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

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

4.21 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](#)

4.21.1 Field Documentation

- 4.21.1.1 char COFF_auxent::x_tagndx[4]
- 4.21.1.2 char COFF_auxent::x_inno[2]
- 4.21.1.3 char COFF_auxent::x_size[2]
- 4.21.1.4 struct { ... } ::@25 COFF_auxent::x_insz
- 4.21.1.5 char COFF_auxent::x_fsize[4]
- 4.21.1.6 union { ... } COFF_auxent::x_misc
- 4.21.1.7 char COFF_auxent::x_innoptr[4]
- 4.21.1.8 char COFF_auxent::x_endndx[4]
- 4.21.1.9 struct { ... } ::@26 COFF_auxent::x_fcn
- 4.21.1.10 char COFF_auxent::x_dimen[E_DIMNUM][2]
- 4.21.1.11 struct { ... } ::@27 COFF_auxent::x_ary
- 4.21.1.12 union { ... } COFF_auxent::x_fcary
- 4.21.1.13 char COFF_auxent::x_tvndx[2]
- 4.21.1.14 struct { ... } COFF_auxent::x_sym
- 4.21.1.15 char COFF_auxent::x_fname[E_FILNMLEN]
- 4.21.1.16 char COFF_auxent::x_zeroes[4]
- 4.21.1.17 char COFF_auxent::x_offset[4]
- 4.21.1.18 struct { ... } COFF_auxent::x_n
- 4.21.1.19 union { ... } COFF_auxent::x_file
- 4.21.1.20 char COFF_auxent::x_scrlen[4]
- 4.21.1.21 char COFF_auxent::x_nreloc[2]
- 4.21.1.22 char COFF_auxent::x_nlinno[2]
- 4.21.1.23 struct { ... } COFF_auxent::x_scn
- 4.21.1.24 char COFF_auxent::x_tvfill[4]
- 4.21.1.25 char COFF_auxent::x_tvlen[2]
- 4.21.1.26 char COFF_auxent::x_tvran[2][2]

4.21.1.27 struct { ... } COFF_auxent::x_tv

Generated on Sun Oct 12 09:05:05 2008 for OpenRISC 1000 Architectural Simulator by Doxygen

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

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

4.22 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]

4.22.1 Field Documentation

4.22.1.1 char COFF_filehdr::f_magic[2]

4.22.1.2 char COFF_filehdr::f_nscns[2]

4.22.1.3 char COFF_filehdr::f_timdat[4]

4.22.1.4 char COFF_filehdr::f_symptr[4]

4.22.1.5 char COFF_filehdr::f_nsyms[4]

4.22.1.6 char COFF_filehdr::f_opthdr[2]

4.22.1.7 char COFF_filehdr::f_flags[2]

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

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

4.23 COFF_lineno Struct Reference

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

Data Fields

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

4.23.1 Field Documentation

4.23.1.1 char COFF_lineno::l_symndx[4]

4.23.1.2 char COFF_lineno::l_paddr[4]

4.23.1.3 union { ... } COFF_lineno::l_addr

4.23.1.4 char COFF_lineno::l_inno[2]

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

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

4.24 COFF_reloc Struct Reference

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

Data Fields

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

4.24.1 Field Documentation

4.24.1.1 char COFF_reloc::r_vaddr[4]

4.24.1.2 char COFF_reloc::r_symndx[4]

4.24.1.3 char COFF_reloc::r_type[2]

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

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

4.25 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]

4.25.1 Field Documentation

4.25.1.1 char COFF_scnhdr::s_name[8]

4.25.1.2 char COFF_scnhdr::s_paddr[4]

4.25.1.3 char COFF_scnhdr::s_vaddr[4]

4.25.1.4 char COFF_scnhdr::s_size[4]

4.25.1.5 char COFF_scnhdr::s_scnptr[4]

4.25.1.6 char COFF_scnhdr::s_relptr[4]

4.25.1.7 char COFF_scnhdr::s_innoptr[4]

4.25.1.8 char COFF_scnhdr::s_nreloc[2]

4.25.1.9 char COFF_scnhdr::s_nlno[2]

4.25.1.10 char COFF_scnhdr::s_flags[4]

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

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

4.26 COFF_slib Struct Reference

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

Data Fields

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

4.26.1 Field Documentation

4.26.1.1 char COFF_slib::sl_entz[4]

4.26.1.2 char COFF_slib::sl_pathndx[4]

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

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

4.27 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]

4.27.1 Field Documentation

4.27.1.1 char COFF_syment::e_name[E_SYMNMLEN]

4.27.1.2 char COFF_syment::e_zeroes[4]

4.27.1.3 char COFF_syment::e_offset[4]

4.27.1.4 struct { ... } COFF_syment::e

4.27.1.5 union { ... } COFF_syment::e

4.27.1.6 char COFF_syment::e_value[4]

4.27.1.7 char COFF_syment::e_snum[2]

4.27.1.8 char COFF_syment::e_type[2]

4.27.1.9 char COFF_syment::e_sclass[1]

4.27.1.10 char COFF_syment::e_numaux[1]

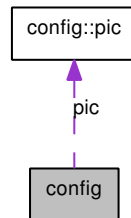
The documentation for this struct was generated from the following file:

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

4.28 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 server\_port  
 unsigned long vapi\_id  
} debug

## Data Structures

- struct pic

### 4.28.1 Detailed Description

Data structure for configuration data



## 4.28.2 Field Documentation

4.28.2.1 void\* config::class\_ptr

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

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

4.28.2.4 struct { ... } config::ext

4.28.2.5 int config::debug

4.28.2.6 int config::verbose

4.28.2.7 int config::profile

4.28.2.8 char\* config::prof\_fn

4.28.2.9 int config::mprofile

4.28.2.10 char\* config::mprof\_fn

4.28.2.11 int config::history

4.28.2.12 int config::exe\_log

4.28.2.13 int config::exe\_log\_type

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

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

4.28.2.16 int config::exe\_log\_marker

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

4.28.2.18 long config::clkcycle\_ps

4.28.2.19 struct { ... } config::sim

4.28.2.20 int config::enabled

4.28.2.21 int config::server\_port

4.28.2.22 int config::log\_enabled

4.28.2.23 int config::hide\_device\_id

4.28.2.24 char\* config::vapi\_fn

4.28.2.25 struct { ... } config::vapi

4.28.2.26 char\* config::timings\_fn

4.28.2.27 int config::memory\_order

4.28.2.28 int config::calling\_convention

4.28.2.29 int config::enable\_bursts

- [sim-config.h](#)

## 4.29 config::pic Struct Reference

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

### Data Fields

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

### 4.29.1 Field Documentation

#### 4.29.1.1 int config::pic::enabled

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

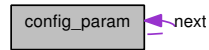
The documentation for this struct was generated from the following file:

- [sim-config.h](#)



## 4.30 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](#)

### 4.30.1 Field Documentation

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

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

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

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

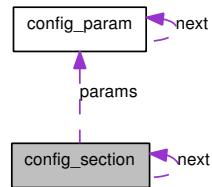
The documentation for this struct was generated from the following file:

- [sim-config.c](#)

## 4.31 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](#)

#### 4.31.1 Field Documentation

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

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

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

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

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

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

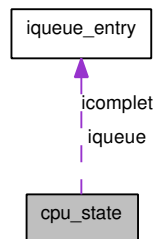
The documentation for this struct was generated from the following file:

- [sim-config.h](#)

## 4.32 `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`

### 4.32.1 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()`.

### 4.32.2 Field Documentation

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

General purpose registers

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

Special purpose registers

**4.32.2.3 oraddr\_t cpu\_state::insn\_ea**

EA of instrs that have an EA

**4.32.2.4 int cpu\_state::delay\_insn**

Is current instr in delay slot

**4.32.2.5 oraddr\_t cpu\_state::pc**

PC (and translated PC)

**4.32.2.6 oraddr\_t cpu\_state::pc\_delay**

Delay instr EA register

**4.32.2.7 uint32\_t cpu\_state::pic\_lines**

State of PIC lines

**4.32.2.8 struct iqueue\_entry cpu\_state::iqueue** [read]

Decode of just executed instr

**4.32.2.9 struct iqueue\_entry cpu\_state::icomplet** [read]

Decode of instr before this

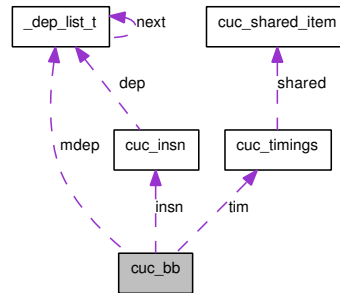
The documentation for this struct was generated from the following file:

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

## 4.33 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`

### 4.33.1 Field Documentation

- 4.33.1.1 unsigned long `cuc_bb::type`
- 4.33.1.2 int `cuc_bb::first`
- 4.33.1.3 int `cuc_bb::last`
- 4.33.1.4 int `cuc_bb::prev[2]`
- 4.33.1.5 int `cuc_bb::next[2]`
- 4.33.1.6 int `cuc_bb::tmp`
- 4.33.1.7 `cuc_insn*` `cuc_bb::insn`
- 4.33.1.8 int `cuc_bb::ninsn`
- 4.33.1.9 int `cuc_bb::last_used_reg[MAX_REGS]`
- 4.33.1.10 `dep_list*` `cuc_bb::mdep`
- 4.33.1.11 int `cuc_bb::nmemory`
- 4.33.1.12 int `cuc_bb::cnt`
- 4.33.1.13 int `cuc_bb::unrolled`
- 4.33.1.14 int `cuc_bb::ntim`
- 4.33.1.15 `cuc_timings*` `cuc_bb::tim`
- 4.33.1.16 int `cuc_bb::selected_tim`

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

- [cuc/cuc.h](#)

## 4.34 `cuc_conv` Struct Reference

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

### Data Fields

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

### 4.34.1 Field Documentation

#### 4.34.1.1 const char\* `cuc_conv::from`

#### 4.34.1.2 const int `cuc_conv::to`

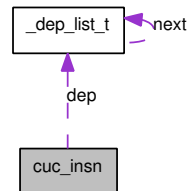
The documentation for this struct was generated from the following file:

- [cuc/insn.h](#)

## 4.35 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`

### 4.35.1 Field Documentation

4.35.1.1 int `cuc_insn::type`

4.35.1.2 int `cuc_insn::index`

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

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

4.35.1.5 `dep_list`\* `cuc_insn::dep`

4.35.1.6 unsigned long `cuc_insn::insn`

4.35.1.7 char `cuc_insn::disasm`[40]

4.35.1.8 unsigned long `cuc_insn::max`

4.35.1.9 int `cuc_insn::tmp`

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

- [cuc/cuc.h](#)



## 4.36 `cuc_known_insn` Struct Reference

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

### Data Fields

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

### 4.36.1 Field Documentation

**4.36.1.1** char\* `cuc_known_insn::name`

**4.36.1.2** int `cuc_known_insn::comutative`

**4.36.1.3** char\* `cuc_known_insn::rtl`

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

- [cuc/insn.h](#)

## 4.37 `cuc_shared_item` Struct Reference

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

### Data Fields

- int [ref](#)
- int [cmatch](#)

### 4.37.1 Field Documentation

#### 4.37.1.1 int `cuc_shared_item::ref`

#### 4.37.1.2 int `cuc_shared_item::cmatch`

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

- [cuc/cuc.h](#)

## 4.38 `cuc_timing_table` Struct Reference

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

### Data Fields

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

### 4.38.1 Field Documentation

**4.38.1.1** double `cuc_timing_table::delay`

**4.38.1.2** double `cuc_timing_table::size`

**4.38.1.3** double `cuc_timing_table::delayi`

**4.38.1.4** double `cuc_timing_table::sizei`

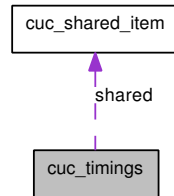
The documentation for this struct was generated from the following file:

- [cuc/insn.h](#)

## 4.39 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](#)

### 4.39.1 Field Documentation

**4.39.1.1 int cuc\_timings::b**

**4.39.1.2 int cuc\_timings::preroll**

**4.39.1.3 int cuc\_timings::unroll**

**4.39.1.4 int cuc\_timings::nshared**

**4.39.1.5 cuc\_shared\_item\* cuc\_timings::shared**

**4.39.1.6 int cuc\_timings::new\_time**

**4.39.1.7 double cuc\_timings::size**

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

- [cuc/cuc.h](#)

## 4.40 dc\_set Struct Reference

### Data Fields

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

### 4.40.1 Field Documentation

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

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

**4.40.1.3** [int dc\\_set::lru](#)

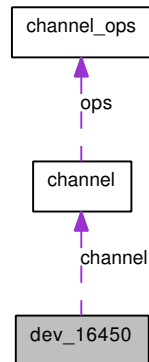
**4.40.1.4** [struct { ... } dc\\_set::way](#)[MAX\_DC\_WAYS]

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

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

## 4.41 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`





### 4.41.1 Field Documentation

4.41.1.1 `uint8_t dev_16450::txbuf[UART_MAX_FIFO_LEN]`

4.41.1.2 `uint16_t dev_16450::rxbuf[UART_MAX_FIFO_LEN]`

4.41.1.3 `uint8_t dev_16450::dll`

4.41.1.4 `uint8_t dev_16450::dlh`

4.41.1.5 `uint8_t dev_16450::ier`

4.41.1.6 `uint8_t dev_16450::iir`

4.41.1.7 `uint8_t dev_16450::fcr`

4.41.1.8 `uint8_t dev_16450::lcr`

4.41.1.9 `uint8_t dev_16450::mcr`

4.41.1.10 `uint8_t dev_16450::lsr`

4.41.1.11 `uint8_t dev_16450::msr`

4.41.1.12 `uint8_t dev_16450::scr`

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

4.41.1.14 `uint8_t dev_16450::txser`

4.41.1.15 `uint16_t dev_16450::rxser`

4.41.1.16 `uint8_t dev_16450::loopback`

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

4.41.1.18 `int dev_16450::txbuf_head`

4.41.1.19 `int dev_16450::txbuf_tail`

4.41.1.20 `int dev_16450::rxbuf_head`

4.41.1.21 `int dev_16450::rxbuf_tail`

4.41.1.22 `unsigned int dev_16450::txbuf_full`

4.41.1.23 `unsigned int dev_16450::rxbuf_full`

4.41.1.24 `int dev_16450::receiveing`

4.41.1.25 `int dev_16450::recv_break`

4.41.1.26 `int dev_16450::ints`

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

4.41.1.28 `unsigned long dev_16450::char_clks`

4.41.1.29 `int dev_16450::skew`

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

- [peripheral/16450.c](#)

## 4.42 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](#)

### 4.42.1 Detailed Description

State associated with the generic device.

### 4.42.2 Member Enumeration Documentation

#### 4.42.2.1 anonymous enum

Enumerator:

*GENERIC\_READ*

*GENERIC\_WRITE*

#### 4.42.2.2 anonymous enum

Enumerator:

*GENERIC\_BYTE*

*GENERIC\_HW*

*GENERIC\_WORD*

### 4.42.3 Field Documentation

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

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

4.42.3.3 `uint32_t dev_generic::value`

4.42.3.4 `int dev_generic::enabled`

4.42.3.5 `int dev_generic::byte_enabled`

4.42.3.6 `int dev_generic::hw_enabled`

4.42.3.7 `int dev_generic::word_enabled`

4.42.3.8 `char* dev_generic::name`

4.42.3.9 `oraddr_t dev_generic::baseaddr`

4.42.3.10 `uint32_t dev_generic::size`

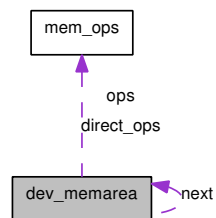
The documentation for this struct was generated from the following file:

- [peripheral/generic.c](#)

## 4.43 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

#### 4.43.1 Detailed Description

Memory regions assigned to devices

## 4.43.2 Field Documentation

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

4.43.2.2 `oraddr_t dev_memarea::addr_mask`

4.43.2.3 `oraddr_t dev_memarea::addr_compare`

4.43.2.4 `uint32_t dev_memarea::size`

4.43.2.5 `oraddr_t dev_memarea::size_mask`

4.43.2.6 `int dev_memarea::valid`

4.43.2.7 `FILE* dev_memarea::log`

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

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

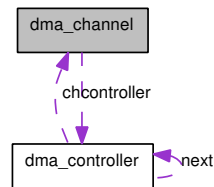
The documentation for this struct was generated from the following file:

- `cpu/common/abstract.h`

## 4.44 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`

### 4.44.1 Field Documentation

- 4.44.1.1 `struct dma_controller* dma_channel::controller` [read]
- 4.44.1.2 `unsigned dma_channel::channel_number`
- 4.44.1.3 `unsigned long dma_channel::channel_mask`
- 4.44.1.4 `unsigned dma_channel::referenced`
- 4.44.1.5 `unsigned dma_channel::load_next_descriptor_when_done`
- 4.44.1.6 `unsigned long dma_channel::current_descriptor`
- 4.44.1.7 `oraddr_t dma_channel::source`
- 4.44.1.8 `oraddr_t dma_channel::destination`
- 4.44.1.9 `oraddr_t dma_channel::source_mask`
- 4.44.1.10 `oraddr_t dma_channel::destination_mask`
- 4.44.1.11 `unsigned long dma_channel::chunk_size`
- 4.44.1.12 `unsigned long dma_channel::total_size`
- 4.44.1.13 `unsigned long dma_channel::words_transferred`
- 4.44.1.14 `unsigned long dma_channel::csr`
- 4.44.1.15 `unsigned long dma_channel::sz`
- 4.44.1.16 `unsigned long dma_channel::a0`
- 4.44.1.17 `unsigned long dma_channel::am0`
- 4.44.1.18 `unsigned long dma_channel::a1`
- 4.44.1.19 `unsigned long dma_channel::am1`
- 4.44.1.20 `unsigned long dma_channel::desc`
- 4.44.1.21 `unsigned long dma_channel::swptr`
- 4.44.1.22 `struct { ... } dma_channel::regs`
- 4.44.1.23 `unsigned dma_channel::dma_req_i`
- 4.44.1.24 `unsigned dma_channel::dma_ack_o`
- 4.44.1.25 `unsigned dma_channel::dma_nd_i`

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

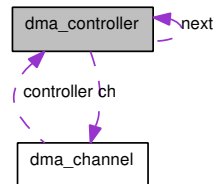


- [peripheral/dma.h](#)

## 4.45 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](#)

## 4.45.1 Field Documentation

4.45.1.1 `int dma_controller::enabled`

4.45.1.2 `oraddr_t dma_controller::baseaddr`

4.45.1.3 `unsigned dma_controller::irq`

4.45.1.4 `int dma_controller::vapi_id`

4.45.1.5 `unsigned long dma_controller::csr`

4.45.1.6 `unsigned long dma_controller::int_msk_a`

4.45.1.7 `unsigned long dma_controller::int_msk_b`

4.45.1.8 `unsigned long dma_controller::int_src_a`

4.45.1.9 `unsigned long dma_controller::int_src_b`

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

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

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

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

- [peripheral/dma.h](#)

## 4.46 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`

## 4.46.1 Field Documentation

- 4.46.1.1 int dmmu::enabled
- 4.46.1.2 int dmmu::nways
- 4.46.1.3 int dmmu::nsets
- 4.46.1.4 int dmmu::pagesize
- 4.46.1.5 int dmmu::pagesize\_log2
- 4.46.1.6 oraddr\_t dmmu::page\_offset\_mask
- 4.46.1.7 oraddr\_t dmmu::page\_mask
- 4.46.1.8 oraddr\_t dmmu::vpn\_mask
- 4.46.1.9 int dmmu::lru\_reload
- 4.46.1.10 oraddr\_t dmmu::set\_mask
- 4.46.1.11 int dmmu::entrysize
- 4.46.1.12 int dmmu::ustates
- 4.46.1.13 int dmmu::missdelay
- 4.46.1.14 int dmmu::hitdelay

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

- [mmu/dmmu.h](#)

## 4.47 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](#)

### 4.47.1 Field Documentation

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

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

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

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

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

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

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

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

## 4.48 `dstats_entry` Struct Reference

### Data Fields

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

### 4.48.1 Field Documentation

4.48.1.1 int `dstats_entry::insn1`

4.48.1.2 int `dstats_entry::insn2`

4.48.1.3 int `dstats_entry::cnt_dynamic`

4.48.1.4 int `dstats_entry::depend`

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

- `cpu/common/stats.c`

## 4.49 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](#)

### 4.49.1 Field Documentation

**4.49.1.1 oraddr\_t dyn\_page::or\_page**

**4.49.1.2 void\* dyn\_page::host\_page**

**4.49.1.3 unsigned int dyn\_page::host\_len**

**4.49.1.4 int dyn\_page::dirty**

**4.49.1.5 int dyn\_page::delayr**

**4.49.1.6 uint16\_t dyn\_page::ts\_bound[2049]**

**4.49.1.7 void\*\* dyn\_page::locs**

**4.49.1.8 uint32\_t\* dyn\_page::insns**

**4.49.1.9 unsigned int\* dyn\_page::insn\_indexes**

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

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



## 4.50 dynamic Struct Reference

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

### Data Fields

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

### 4.50.1 Field Documentation

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

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

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

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

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

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

## 4.51 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](#)

### 4.51.1 Field Documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 4.52 elf32\_note Struct Reference

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

### Data Fields

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

### 4.52.1 Field Documentation

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

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

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

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

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

## 4.53 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](#)

### 4.53.1 Field Documentation

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

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

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

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

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

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

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

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

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

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

## 4.54 elf32\_rel Struct Reference

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

### Data Fields

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

### 4.54.1 Field Documentation

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

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

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

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

## 4.55 elf32\_rela Struct Reference

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

### Data Fields

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

### 4.55.1 Field Documentation

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

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

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

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

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

## 4.56 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](#)

### 4.56.1 Field Documentation

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

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

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

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

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

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

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

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

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

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

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

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



## 4.57 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](#)

### 4.57.1 Field Documentation

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

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

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

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

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

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

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

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

## 4.58 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](#)

### 4.58.1 Field Documentation

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

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

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

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

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

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

## 4.59 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](#)

### 4.59.1 Field Documentation

- 4.59.1.1 unsigned char elf64\_hdr::e\_ident[16]
- 4.59.1.2 short int elf64\_hdr::e\_type
- 4.59.1.3 short unsigned int elf64\_hdr::e\_machine
- 4.59.1.4 int elf64\_hdr::e\_version
- 4.59.1.5 unsigned long long elf64\_hdr::e\_entry
- 4.59.1.6 unsigned long long elf64\_hdr::e\_phoff
- 4.59.1.7 unsigned long long elf64\_hdr::e\_shoff
- 4.59.1.8 int elf64\_hdr::e\_flags
- 4.59.1.9 short int elf64\_hdr::e\_ehsize
- 4.59.1.10 short int elf64\_hdr::e\_phentsize
- 4.59.1.11 short int elf64\_hdr::e\_phnum
- 4.59.1.12 short int elf64\_hdr::e\_shentsize
- 4.59.1.13 short int elf64\_hdr::e\_shnum
- 4.59.1.14 short int elf64\_hdr::e\_shstrndx

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

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

## 4.60 elf64\_note Struct Reference

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

### Data Fields

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

### 4.60.1 Field Documentation

**4.60.1.1** unsigned int `elf64_note::n_namesz`

**4.60.1.2** unsigned int `elf64_note::n_descsz`

**4.60.1.3** unsigned int `elf64_note::n_type`

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

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

## 4.61 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](#)

### 4.61.1 Field Documentation

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

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

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

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

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

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

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

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

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

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

## 4.62 elf64\_rel Struct Reference

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

### Data Fields

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

### 4.62.1 Field Documentation

**4.62.1.1 unsigned long long elf64\_rel::r\_offset**

**4.62.1.2 unsigned long long elf64\_rel::r\_info**

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

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

## 4.63 elf64\_rela Struct Reference

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

### Data Fields

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

### 4.63.1 Field Documentation

**4.63.1.1** unsigned long long `elf64_rela::r_offset`

**4.63.1.2** unsigned long long `elf64_rela::r_info`

**4.63.1.3** unsigned long long `elf64_rela::r_addend`

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

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



## 4.64 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](#)

### 4.64.1 Field Documentation

**4.64.1.1** unsigned int `elf64_shdr::sh_name`

**4.64.1.2** unsigned int `elf64_shdr::sh_type`

**4.64.1.3** unsigned long long `elf64_shdr::sh_flags`

**4.64.1.4** unsigned long long `elf64_shdr::sh_addr`

**4.64.1.5** unsigned long long `elf64_shdr::sh_offset`

**4.64.1.6** unsigned long long `elf64_shdr::sh_size`

**4.64.1.7** unsigned int `elf64_shdr::sh_link`

**4.64.1.8** unsigned int `elf64_shdr::sh_info`

**4.64.1.9** unsigned long long `elf64_shdr::sh_addralign`

**4.64.1.10** unsigned long long `elf64_shdr::sh_entsize`

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

- `cpu/common/elf.h`

## 4.65 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](#)

### 4.65.1 Field Documentation

**4.65.1.1** unsigned int `elf64_sym::st_name`

**4.65.1.2** unsigned char `elf64_sym::st_info`

**4.65.1.3** unsigned char `elf64_sym::st_other`

**4.65.1.4** unsigned short `elf64_sym::st_shndx`

**4.65.1.5** unsigned long long `elf64_sym::st_value`

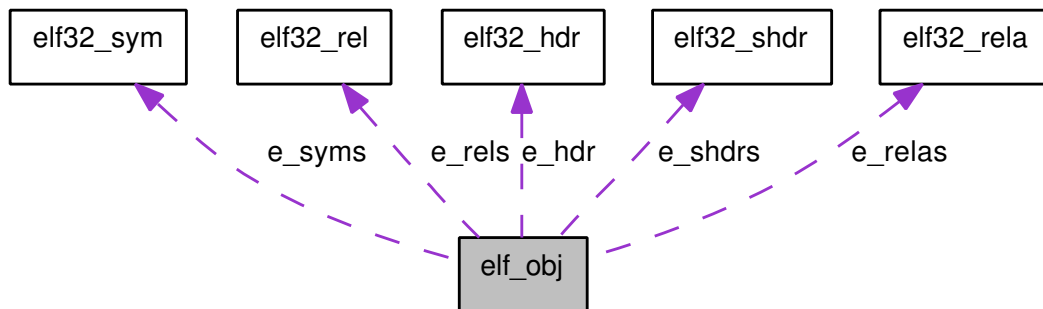
**4.65.1.6** unsigned long long `elf64_sym::st_size`

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

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

## 4.66 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`

## 4.66.1 Field Documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 4.67 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`
- 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]

4.67.1 Field Documentation

4.67.1.1 int eth_device::enabled

4.67.1.2 oraddr_t eth_device::baseaddr

4.67.1.3 unsigned eth_device::dma

4.67.1.4 unsigned eth_device::tx_channel

4.67.1.5 unsigned eth_device::rx_channel

4.67.1.6 unsigned char eth_device::mac_address[ETHER_ADDR_LEN]

4.67.1.7 unsigned long eth_device::mac_int

4.67.1.8 unsigned long eth_device::base_vapi_id

4.67.1.9 char* eth_device::rxfile

4.67.1.10 char * eth_device::txfile

4.67.1.11 int eth_device::txfd

4.67.1.12 int eth_device::rxfd

4.67.1.13 off_t eth_device::loopback_offset

4.67.1.14 char* eth_device::sockif

4.67.1.15 int eth_device::rtx_sock

4.67.1.16 int eth_device::rtx_type

4.67.1.17 struct ifreq eth_device::ifr [read]

4.67.1.18 fd_set eth_device::rfd

4.67.1.19 fd_set eth_device::wfd

4.67.1.20 unsigned long eth_device::state

4.67.1.21 unsigned long eth_device::bd_index

4.67.1.22 unsigned long eth_device::bd

4.67.1.23 unsigned long eth_device::bd_addr

4.67.1.24 unsigned eth_device::working

4.67.1.25 unsigned eth_device::waiting_for_dma

4.67.1.26 unsigned eth_device::error

4.67.1.27 long eth_device::packet_length

4.67.1.28 unsigned eth_device::minimum_length

4.67.1.29 unsigned eth_device::maximum_length

4.67.1.30 unsigned eth_device::add_crc

- [peripheral/eth.c](#)

4.68 ether_addr Struct Reference

Data Fields

- `u_int8_t ether_addr_octet` [ETH_ALEN]

4.68.1 Field Documentation

4.68.1.1 `u_int8_t ether_addr::ether_addr_octet`[ETH_ALEN]

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

- `peripheral/eth.c`

4.69 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`

4.69.1 Field Documentation

4.69.1.1 `u_int8_t ether_header::ether_dhost`[ETH_ALEN]

4.69.1.2 `u_int8_t ether_header::ether_shost`[ETH_ALEN]

4.69.1.3 `u_int16_t ether_header::ether_type`

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

- `peripheral/eth.c`

4.70 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`

4.70.1 Field Documentation

4.70.1.1 int fb_state::enabled

4.70.1.2 unsigned long fb_state::pal[256]

4.70.1.3 int fb_state::ctrl

4.70.1.4 int fb_state::pic

4.70.1.5 int fb_state::in_refresh

4.70.1.6 int fb_state::refresh_count

4.70.1.7 oraddr_t fb_state::addr

4.70.1.8 oraddr_t fb_state::cam_addr

4.70.1.9 int fb_state::camerax

4.70.1.10 int fb_state::cameray

4.70.1.11 int fb_state::camera_pos

4.70.1.12 oraddr_t fb_state::baseaddr

4.70.1.13 int fb_state::refresh

4.70.1.14 int fb_state::refresh_rate

4.70.1.15 char* fb_state::filename

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

- [peripheral/fb.c](#)

4.71 fd_channel Struct Reference

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

Data Fields

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

4.71.1 Detailed Description

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

4.71.2 Field Documentation

4.71.2.1 int fd_channel::fdin

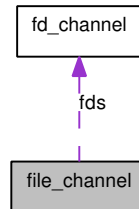
4.71.2.2 int fd_channel::fdout

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

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

4.72 file_channel Struct Reference

Collaboration diagram for file_channel:



Data Fields

- struct [fd_channel](#) `fds`
- char * `namein`
- char * `nameout`

4.72.1 Detailed Description

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

4.72.2 Field Documentation

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

4.72.2.2 char* `file_channel::namein`

4.72.2.3 char* `file_channel::nameout`

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

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

4.73 fstats_entry Struct Reference

Data Fields

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

4.73.1 Field Documentation

4.73.1.1 enum insn_type fstats_entry::insn1

4.73.1.2 enum insn_type fstats_entry::insn2

4.73.1.3 int fstats_entry::cnt_dynamic

4.73.1.4 int fstats_entry::depend

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

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

4.74 func_struct Struct Reference

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

Data Fields

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

4.74.1 Detailed Description

Data structure for information about functions

4.74.2 Field Documentation

4.74.2.1 unsigned int func_struct::addr

Start address of function

4.74.2.2 char func_struct::name[33]

Name of the function

4.74.2.3 long func_struct::cum_cycles

Total cycles spent in function

4.74.2.4 long func_struct::calls

Calls to this function

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

- [profiler.h](#)

4.75 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`

4.75.1 Field Documentation

- 4.75.1.1 `int gpio_device::enabled`
- 4.75.1.2 `oraddr_t gpio_device::baseaddr`
- 4.75.1.3 `int gpio_device::irq`
- 4.75.1.4 `unsigned gpio_device::gpio_number`
- 4.75.1.5 `unsigned long gpio_device::base_vapi_id`
- 4.75.1.6 `unsigned long gpio_device::auxiliary_inputs`
- 4.75.1.7 `unsigned long gpio_device::in`
- 4.75.1.8 `unsigned long gpio_device::out`
- 4.75.1.9 `unsigned long gpio_device::oe`
- 4.75.1.10 `unsigned long gpio_device::inte`
- 4.75.1.11 `unsigned long gpio_device::ptrig`
- 4.75.1.12 `unsigned long gpio_device::aux`
- 4.75.1.13 `unsigned long gpio_device::ctrl`
- 4.75.1.14 `unsigned long gpio_device::ints`
- 4.75.1.15 `int gpio_device::external_clock`
- 4.75.1.16 `struct { ... } gpio_device::curr`
- 4.75.1.17 `struct { ... } gpio_device::next`

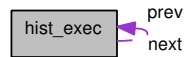
The documentation for this struct was generated from the following file:

- [peripheral/gpio.c](#)

4.76 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](#)

4.76.1 Detailed Description

History of execution

4.76.2 Field Documentation

4.76.2.1 oraddr_t hist_exec::addr

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

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

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

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

4.77 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](#)

4.77.1 Field Documentation

4.77.1.1 `uint8_t* ic::mem`

4.77.1.2 `unsigned int* ic::lrus`

4.77.1.3 `oraddr_t* ic::tags`

4.77.1.4 `int ic::enabled`

4.77.1.5 `unsigned int ic::nways`

4.77.1.6 `unsigned int ic::nsets`

4.77.1.7 `unsigned int ic::blocksize`

4.77.1.8 `unsigned int ic::ustates`

4.77.1.9 `int ic::missdelay`

4.77.1.10 `int ic::hitdelay`

4.77.1.11 `unsigned int ic::blocksize_log2`

4.77.1.12 `oraddr_t ic::set_mask`

4.77.1.13 `oraddr_t ic::tagaddr_mask`

4.77.1.14 `oraddr_t ic::last_way`

4.77.1.15 `oraddr_t ic::block_offset_mask`

4.77.1.16 `oraddr_t ic::block_mask`

4.77.1.17 `unsigned int ic::ustates_reload`

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

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

4.78 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`

4.78.1 Field Documentation

4.78.1.1 `int immu::enabled`

4.78.1.2 `int immu::nways`

4.78.1.3 `int immu::nsets`

4.78.1.4 `oraddr_t immu::pagesize`

4.78.1.5 `int immu::pagesize_log2`

4.78.1.6 `oraddr_t immu::page_offset_mask`

4.78.1.7 `oraddr_t immu::page_mask`

4.78.1.8 `oraddr_t immu::vpn_mask`

4.78.1.9 `int immu::lru_reload`

4.78.1.10 `oraddr_t immu::set_mask`

4.78.1.11 `int immu::entrysize`

4.78.1.12 `int immu::ustates`

4.78.1.13 `int immu::missdelay`

4.78.1.14 `int immu::hitdelay`

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

- [mmu/immu.h](#)

4.79 immustats_entry Struct Reference

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

Data Fields

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

4.79.1 Field Documentation

4.79.1.1 int immustats_entry::fetch_tlbhit

4.79.1.2 int immustats_entry::fetch_tlbmiss

4.79.1.3 int immustats_entry::fetch_pagefaults

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

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

4.80 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](#)

4.80.1 Field Documentation

4.80.1.1 unsigned int INFOHEADER::size

4.80.1.2 int INFOHEADER::width

4.80.1.3 int INFOHEADER::height

4.80.1.4 unsigned short int INFOHEADER::planes

4.80.1.5 unsigned short int INFOHEADER::bits

4.80.1.6 unsigned int INFOHEADER::compression

4.80.1.7 unsigned int INFOHEADER::imagesize

4.80.1.8 int INFOHEADER::xresolution

4.80.1.9 int INFOHEADER::yresolution

4.80.1.10 unsigned int INFOHEADER::ncolours

4.80.1.11 unsigned int INFOHEADER::importantcolours

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

- [peripheral/vga.c](#)

4.81 iqueue_entry Struct Reference

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

Data Fields

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

4.81.1 Detailed Description

Instruction queue

4.81.2 Field Documentation

4.81.2.1 int `iqueue_entry::insn_index`

4.81.2.2 uint32_t `iqueue_entry::insn`

4.81.2.3 oraddr_t `iqueue_entry::insn_addr`

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

- `cpu/common/abstract.h`

4.82 jtr_chain_message Struct Reference

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

Data Fields

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

4.82.1 Field Documentation

4.82.1.1 uint32_t jtr_chain_message::command

4.82.1.2 uint32_t jtr_chain_message::length

4.82.1.3 uint32_t jtr_chain_message::chain

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

- [debug/gdb.h](#)

4.83 jtr_chain_response Struct Reference

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

Data Fields

- [int32_t status](#)

4.83.1 Field Documentation

4.83.1.1 int32_t jtr_chain_response::status

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

- [debug/gdb.h](#)

4.84 jtr_failure_response Struct Reference

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

Data Fields

- [int32_t status](#)

4.84.1 Field Documentation

4.84.1.1 int32_t jtr_failure_response::status

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

- [debug/gdb.h](#)

4.85 jtr_read_block_message Struct Reference

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

Data Fields

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

4.85.1 Field Documentation

4.85.1.1 uint32_t jtr_read_block_message::command

4.85.1.2 uint32_t jtr_read_block_message::length

4.85.1.3 uint32_t jtr_read_block_message::address

4.85.1.4 int32_t jtr_read_block_message::num_regs

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

- [debug/gdb.h](#)

4.86 jtr_read_block_response Struct Reference

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

Data Fields

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

4.86.1 Field Documentation

4.86.1.1 `int32_t jtr_read_block_response::status`

4.86.1.2 `int32_t jtr_read_block_response::num_regs`

4.86.1.3 `uint32_t jtr_read_block_response::data[1]`

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

- [debug/gdb.h](#)

4.87 jtr_read_message Struct Reference

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

Data Fields

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

4.87.1 Field Documentation

4.87.1.1 uint32_t jtr_read_message::command

4.87.1.2 uint32_t jtr_read_message::length

4.87.1.3 uint32_t jtr_read_message::address

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

- [debug/gdb.h](#)

4.88 jtr_read_response Struct Reference

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

Data Fields

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

4.88.1 Field Documentation

4.88.1.1 [int32_t jtr_read_response::status](#)

4.88.1.2 [uint32_t jtr_read_response::data_h](#)

4.88.1.3 [uint32_t jtr_read_response::data_l](#)

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

- [debug/gdb.h](#)

4.89 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 \[1\]](#)

4.89.1 Field Documentation

4.89.1.1 [uint32_t jtr_write_block_message::command](#)

4.89.1.2 [uint32_t jtr_write_block_message::length](#)

4.89.1.3 [uint32_t jtr_write_block_message::address](#)

4.89.1.4 [int32_t jtr_write_block_message::num_regs](#)

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

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

- [debug/gdb.h](#)

4.90 jtr_write_block_response Struct Reference

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

Data Fields

- [int32_t status](#)

4.90.1 Field Documentation

4.90.1.1 int32_t jtr_write_block_response::status

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

- [debug/gdb.h](#)

4.91 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](#)

4.91.1 Field Documentation

4.91.1.1 [uint32_t jtr_write_message::command](#)

4.91.1.2 [uint32_t jtr_write_message::length](#)

4.91.1.3 [uint32_t jtr_write_message::address](#)

4.91.1.4 [uint32_t jtr_write_message::data_h](#)

4.91.1.5 [uint32_t jtr_write_message::data_l](#)

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

- [debug/gdb.h](#)

4.92 jtr_write_response Struct Reference

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

Data Fields

- [int32_t status](#)

4.92.1 Field Documentation

4.92.1.1 int32_t jtr_write_response::status

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

- [debug/gdb.h](#)

4.93 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](#)

4.93.1 Field Documentation

4.93.1.1 uint8_t kbd_state::buf[KBD_MAX_BUF]

4.93.1.2 unsigned long kbd_state::buf_count

4.93.1.3 unsigned long kbd_state::buf_head

4.93.1.4 unsigned long kbd_state::buf_tail

4.93.1.5 FILE* kbd_state::rxfs

4.93.1.6 int kbd_state::ccmd

4.93.1.7 uint8_t kbd_state::kcmd

4.93.1.8 uint8_t kbd_state::ccmdbyte

4.93.1.9 unsigned long kbd_state::kresp

4.93.1.10 long kbd_state::slowdown

4.93.1.11 int kbd_state::enabled

4.93.1.12 int kbd_state::irq

4.93.1.13 oraddr_t kbd_state::baseaddr

4.93.1.14 char* kbd_state::rxfile

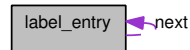
The documentation for this struct was generated from the following file:

- [peripheral/ps2kbd.c](#)

4.94 label_entry Struct Reference

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

Collaboration diagram for label_entry:



Data Fields

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

4.94.1 Detailed Description

Structure for holding one label per particular memory location

4.94.2 Field Documentation

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

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

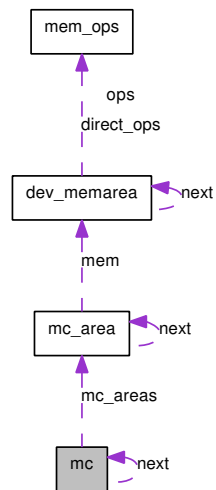
4.94.2.3 struct [label_entry](#)* [label_entry::next](#) [read]

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

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

4.95 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`

4.95.1 Field Documentation

4.95.1.1 `uint32_t mc::csr`

4.95.1.2 `uint32_t mc::poc`

4.95.1.3 `uint32_t mc::ba_mask`

4.95.1.4 `uint32_t mc::csc[N_CE]`

4.95.1.5 `uint32_t mc::tms[N_CE]`

4.95.1.6 `oraddr_t mc::baseaddr`

4.95.1.7 `int mc::enabled`

4.95.1.8 `int mc::index`

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

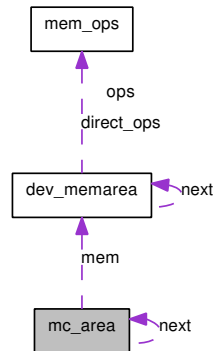
4.95.1.10 `struct mc* mc::next` [read]

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

- [peripheral/mc.c](#)

4.96 mc_area Struct Reference

Collaboration diagram for mc_area:



Data Fields

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

4.96.1 Field Documentation

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

4.96.1.2 unsigned int mc_area::cs

4.96.1.3 int mc_area::mc

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

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

- peripheral/[mc.c](#)

4.97 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](#)

4.97.1 Member Enumeration Documentation

4.97.1.1 anonymous enum

Enumerator:

MT_UNKNOWN

MT_PATTERN

MT_RANDOM

4.97.2 Field Documentation

4.97.2.1 `int mem_config::ce`

4.97.2.2 `int mem_config::mc`

4.97.2.3 `oraddr_t mem_config::baseaddr`

4.97.2.4 `unsigned int mem_config::size`

4.97.2.5 `char* mem_config::name`

4.97.2.6 `char* mem_config::log`

4.97.2.7 `int mem_config::delayr`

4.97.2.8 `int mem_config::delayw`

4.97.2.9 `void* mem_config::mem`

4.97.2.10 `int mem_config::pattern`

4.97.2.11 `int mem_config::random_seed`

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

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

- [peripheral/memory.c](#)

4.98 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](#)

4.98.1 Detailed Description

All the memory operations possible

4.98.2 Field Documentation

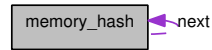
- 4.98.2.1 `uint32_t(* mem_ops::readfunc32)(oraddr_t, void *)`
- 4.98.2.2 `uint16_t(* mem_ops::readfunc16)(oraddr_t, void *)`
- 4.98.2.3 `uint8_t(* mem_ops::readfunc8)(oraddr_t, void *)`
- 4.98.2.4 `void* mem_ops::read_dat8`
- 4.98.2.5 `void* mem_ops::read_dat16`
- 4.98.2.6 `void* mem_ops::read_dat32`
- 4.98.2.7 `void(* mem_ops::writefunc32)(oraddr_t, uint32_t, void *)`
- 4.98.2.8 `void(* mem_ops::writefunc16)(oraddr_t, uint16_t, void *)`
- 4.98.2.9 `void(* mem_ops::writefunc8)(oraddr_t, uint8_t, void *)`
- 4.98.2.10 `void* mem_ops::write_dat8`
- 4.98.2.11 `void* mem_ops::write_dat16`
- 4.98.2.12 `void* mem_ops::write_dat32`
- 4.98.2.13 `void(* mem_ops::writeprog32)(oraddr_t, uint32_t, void *)`
- 4.98.2.14 `void(* mem_ops::writeprog8)(oraddr_t, uint8_t, void *)`
- 4.98.2.15 `void* mem_ops::writeprog32_dat`
- 4.98.2.16 `void* mem_ops::writeprog8_dat`
- 4.98.2.17 `int mem_ops::delayr`
- 4.98.2.18 `int mem_ops::delayw`
- 4.98.2.19 `const char* mem_ops::log`

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

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

4.99 memory_hash Struct Reference

Collaboration diagram for memory_hash:



Data Fields

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

4.99.1 Detailed Description

Hash table data structure

4.99.2 Field Documentation

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

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

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

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

- [mprofiler.c](#)

4.100 mprofentry_struct Struct Reference

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

Data Fields

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

4.100.1 Field Documentation

4.100.1.1 oraddr_t mprofentry_struct::addr

4.100.1.2 unsigned char mprofentry_struct::type

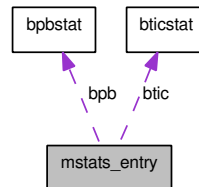
The documentation for this struct was generated from the following file:

- [support/profile.h](#)

4.101 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](#)

4.101.1 Field Documentation

4.101.1.1 int [mstats_entry::byteadd](#)

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

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

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

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

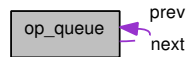
The documentation for this struct was generated from the following file:

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

4.102 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](#)

4.102.1 Field Documentation

- 4.102.1.1 `unsigned int op_queue::num_ops`
- 4.102.1.2 `unsigned int op_queue::ops_len`
- 4.102.1.3 `unsigned int* op_queue::ops`
- 4.102.1.4 `unsigned int op_queue::num_ops_param`
- 4.102.1.5 `unsigned int op_queue::ops_param_len`
- 4.102.1.6 `unsigned int* op_queue::ops_param`
- 4.102.1.7 `unsigned int op_queue::jump_local`
- 4.102.1.8 `oraddr_t op_queue::jump_local_loc`
- 4.102.1.9 `unsigned int op_queue::not_jump_loc`
- 4.102.1.10 `int op_queue::xref`
- 4.102.1.11 `oraddr_t op_queue::insn_addr`
- 4.102.1.12 `unsigned int op_queue::reg_t[3]`
- 4.102.1.13 `unsigned int op_queue::tflags[3]`
- 4.102.1.14 `int op_queue::insn_index`
- 4.102.1.15 `unsigned int op_queue::param_type[5]`
- 4.102.1.16 `orreg_t op_queue::param[5]`
- 4.102.1.17 `unsigned int op_queue::param_num`
- 4.102.1.18 `uint32_t op_queue::insn`
- 4.102.1.19 `struct op_queue* op_queue::prev` [read]
- 4.102.1.20 `struct op_queue* op_queue::next` [read]

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

- [cpu/or32/dyn32_defs.h](#)

4.103 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](#)

4.103.1 Detailed Description

Union of all possible paramter values

4.103.2 Field Documentation

4.103.2.1 char* [param_val::str_val](#)

4.103.2.2 int [param_val::int_val](#)

4.103.2.3 long long int [param_val::longlong_val](#)

4.103.2.4 [oraddr_t](#) [param_val::addr_val](#)

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

- [sim-config.h](#)

4.104 raw_stats Struct Reference

Data Fields

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

4.104.1 Field Documentation

4.104.1.1 int raw_stats::reg[64]

4.104.1.2 int raw_stats::range[RAW_RANGE]

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

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

4.105 reloc Struct Reference

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

Data Fields

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

4.105.1 Field Documentation

4.105.1.1 unsigned int `reloc::func_offset`

4.105.1.2 unsigned int `reloc::addend`

4.105.1.3 int `reloc::type`

4.105.1.4 const char* `reloc::name`

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

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

4.106 runtime Struct Reference

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

Data Fields

- struct {
 - FILE * [fprof](#)
 - FILE * [fmprof](#)
 - FILE * [fexe_log](#)
 - FILE * [fout](#)
 - int [init](#)
 - 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](#)

4.106.1 Detailed Description

Data structure for run time data

4.106.2 Field Documentation

4.106.2.1 FILE* runtime::fprof

4.106.2.2 FILE* runtime::fmprof

4.106.2.3 FILE* runtime::fexe_log

4.106.2.4 FILE* runtime::fout

4.106.2.5 int runtime::init

4.106.2.6 char* runtime::filename

4.106.2.7 int runtime::iprompt

4.106.2.8 int runtime::iprompt_run

4.106.2.9 long long runtime::cycles

4.106.2.10 long long int runtime::end_cycles

4.106.2.11 double runtime::time_point

4.106.2.12 unsigned long int runtime::ext_int

4.106.2.13 int runtime::mem_cycles

4.106.2.14 int runtime::loadcycles

4.106.2.15 int runtime::storecycles

4.106.2.16 long long runtime::reset_cycles

4.106.2.17 int runtime::hush

4.106.2.18 struct { ... } runtime::sim

4.106.2.19 long long runtime::instructions

4.106.2.20 long long runtime::reset_instructions

4.106.2.21 int runtime::stalled

4.106.2.22 int runtime::hazardwait

4.106.2.23 int runtime::supercycles

4.106.2.24 struct { ... } runtime::cpu

4.106.2.25 int runtime::enabled

4.106.2.26 FILE* runtime::vapi_file

4.106.2.27 int runtime::server_port

4.106.2.28 struct { ... } runtime::vapi

4.106.2.29 int runtime::mdelay[4]

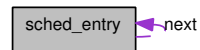
4.106.2.30 double runtime::cycle_duration

- [sim-config.h](#)

4.107 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](#)

4.107.1 Detailed Description

Structure for holding one job entry

4.107.2 Field Documentation

4.107.2.1 `int32_t` `sched_entry::time`

4.107.2.2 `void*` `sched_entry::param`

4.107.2.3 `void(* sched_entry::func)(void *)`

4.107.2.4 `struct sched_entry*` `sched_entry::next` [read]

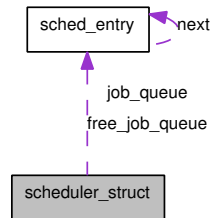
The documentation for this struct was generated from the following file:

- [support/sched.h](#)

4.108 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`

4.108.1 Detailed Description

Heap of jobs

4.108.2 Field Documentation

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

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

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

- support/[sched.h](#)

4.109 `sim_command` Struct Reference

Data Fields

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

4.109.1 Field Documentation

4.109.1.1 `const char* sim_command::name`

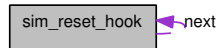
4.109.1.2 `int(* sim_command::cmd_handle)(int argc, char **argv)`

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

- [sim-cmd.c](#)

4.110 `sim_reset_hook` Struct Reference

Collaboration diagram for `sim_reset_hook`:



Data Fields

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

4.110.1 Detailed Description

Struct for list of reset hooks

4.110.2 Field Documentation

4.110.2.1 `void* sim_reset_hook::dat`

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

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

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

- [toplevel-support.c](#)

4.111 `sim_stat` Struct Reference

Collaboration diagram for `sim_stat`:



Data Fields

- `void(* stat_func)(void *dat)`
- `void * dat`
- `struct sim_stat * next`

4.111.1 Field Documentation

4.111.1.1 `void(* sim_stat::stat_func)(void *dat)`

4.111.1.2 `void* sim_stat::dat`

4.111.1.3 `struct sim_stat* sim_stat::next [read]`

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

- [sim-cmd.c](#)

4.112 spr_bit_def Struct Reference

Data Fields

- const char * [name](#)
- [uorreg_t](#) mask

4.112.1 Field Documentation

4.112.1.1 const char* [spr_bit_def::name](#)

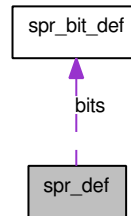
4.112.1.2 [uorreg_t](#) [spr_bit_def::mask](#)

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

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

4.113 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](#)

4.113.1 Field Documentation

4.113.1.1 [uint16_t spr_def::from_spr](#)

4.113.1.2 [uint16_t spr_def::to_spr](#)

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

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

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

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

4.114 sstats_entry Struct Reference

Data Fields

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

4.114.1 Field Documentation

4.114.1.1 int sstats_entry::insn

4.114.1.2 int sstats_entry::cnt_dynamic

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

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

4.115 `stack_struct` Struct Reference

Data Fields

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

4.115.1 Detailed Description

Data structure representing information about a stack frame

4.115.2 Field Documentation

4.115.2.1 unsigned int `stack_struct::addr`

Function address

4.115.2.2 unsigned int `stack_struct::cycles`

Cycles of func start; subfuncs added later

4.115.2.3 unsigned int `stack_struct::raddr`

Return address

4.115.2.4 char `stack_struct::name[33]`

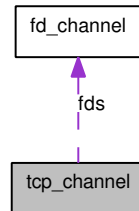
Name of the function

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

- [profiler.c](#)

4.116 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](#)

4.116.1 Detailed Description

Structure to represent a TCP/IP [channel](#)

4.116.2 Field Documentation

4.116.2.1 struct `fd_channel` `tcp_channel::fds` [`read`]

4.116.2.2 int `tcp_channel::socket_fd`

4.116.2.3 int `tcp_channel::port_number`

4.116.2.4 int `tcp_channel::connected`

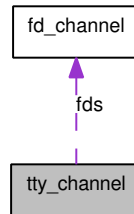
4.116.2.5 int `tcp_channel::nonblocking`

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

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

4.117 tty_channel Struct Reference

Collaboration diagram for tty_channel:



Data Fields

- struct [fd_channel](#) `fds`

4.117.1 Detailed Description

Data structure representing a TTY [channel](#)

4.117.2 Field Documentation

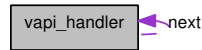
4.117.2.1 struct `fd_channel` `tty_channel::fds` [read]

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

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

4.118 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](#)

4.118.1 Field Documentation

4.118.1.1 int `vapi_handler::fd`

4.118.1.2 unsigned long `vapi_handler::base_id`

4.118.1.3 unsigned long `vapi_handler::num_ids`

4.118.1.4 void(* `vapi_handler::read_func`)(unsigned long, unsigned long, void *)

4.118.1.5 void* `vapi_handler::priv_dat`

4.118.1.6 struct `vapi_handler`* `vapi_handler::next` [`read`]

4.118.1.7 int `vapi_handler::temp`

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

- [vapi/vapi.c](#)

4.119 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](#)

4.119.1 Field Documentation

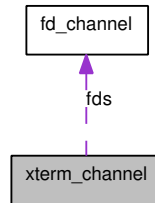
- 4.119.1.1 `int vga_state::enabled`
- 4.119.1.2 `int vga_state::pics`
- 4.119.1.3 `unsigned long vga_state::ctrl`
- 4.119.1.4 `unsigned long vga_state::stat`
- 4.119.1.5 `unsigned long vga_state::htim`
- 4.119.1.6 `unsigned long vga_state::vtim`
- 4.119.1.7 `int vga_state::vindex`
- 4.119.1.8 `unsigned long vga_state::vbar[2]`
- 4.119.1.9 `unsigned vga_state::hlen`
- 4.119.1.10 `unsigned vga_state::vlen`
- 4.119.1.11 `int vga_state::pindex`
- 4.119.1.12 `unsigned long vga_state::palette[2][256]`
- 4.119.1.13 `oraddr_t vga_state::baseaddr`
- 4.119.1.14 `int vga_state::refresh_rate`
- 4.119.1.15 `int vga_state::irq`
- 4.119.1.16 `char* vga_state::filename`

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

- [peripheral/vga.c](#)

4.120 xterm_channel Struct Reference

Collaboration diagram for xterm_channel:



Data Fields

- struct [fd_channel](#) `fds`
- int `pid`
- char ** `argv`

4.120.1 Detailed Description

Data structure to represent the connection to the xterm

4.120.2 Field Documentation

4.120.2.1 struct `fd_channel` `xterm_channel::fds` [`read`]

4.120.2.2 int `xterm_channel::pid`

4.120.2.3 char** `xterm_channel::argv`

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

- `peripheral/channels/xterm.c`

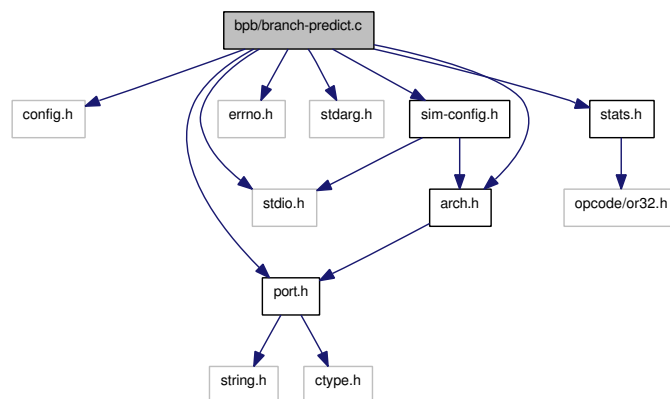
Chapter 5

File Documentation

5.1 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 [btic_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`]

5.1.1 Define Documentation

5.1.1.1 `#define BPB_LEN 64`

5.1.1.2 `#define BPB_PSTATES 2`

5.1.1.3 `#define BPB_USTATES 2`

5.1.1.4 `#define BPB_WAYS 1`

5.1.1.5 `#define BTIC_BLOCKSIZE 4`

5.1.1.6 `#define BTIC_LEN 128`

5.1.1.7 `#define BTIC_USTATES 2`

5.1.1.8 `#define BTIC_WAYS 2`

5.1.2 Function Documentation

5.1.2.1 `static void bpb_btic (union param_val val, void * dat) [static]`

5.1.2.2 `static void bpb_enabled (union param_val val, void * dat) [static]`

5.1.2.3 `static void bpb_hitdelay (union param_val val, void * dat) [static]`

5.1.2.4 `void bpb_info ()`

5.1.2.5 `static void bpb_missdelay (union param_val val, void * dat) [static]`

5.1.2.6 `static void bpb_sbp_bf_fwd (union param_val val, void * dat) [static]`

5.1.2.7 `static void bpb_sbp_bnf_fwd (union param_val val, void * dat) [static]`

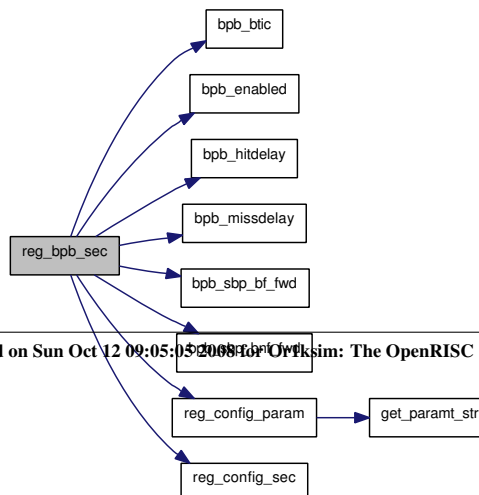
5.1.2.8 `void bpb_update (oraddr_t addr, int taken)`

5.1.2.9 `void btic_info ()`

5.1.2.10 `void btic_update (oraddr_t targetaddr)`

5.1.2.11 `void reg_bpb_sec ()`

Here is the call graph for this function:



5.1.3 Variable Documentation

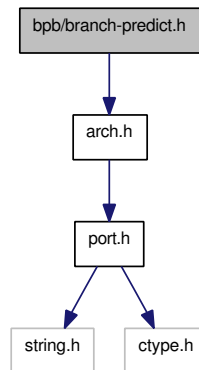
5.1.3.1 struct bpb_entry bpb[BPB_LEN]

5.1.3.2 struct btic_entry btic[BTIC_LEN]

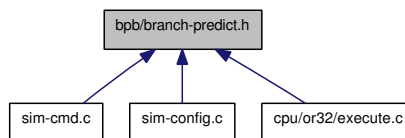
5.2 bpb/branch-predict.h File Reference

```
#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` ()

5.2.1 Function Documentation

5.2.1.1 void `bpb_info ()`

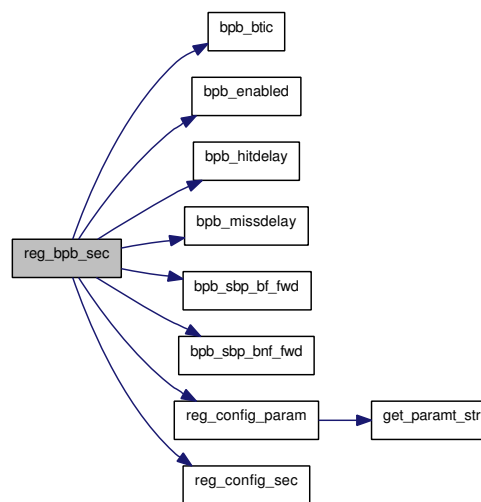
5.2.1.2 void `bpb_update (oraddr_t addr, int taken)`

5.2.1.3 void `btic_info ()`

5.2.1.4 void `btic_update (oraddr_t targetaddr)`

5.2.1.5 void `reg_bpb_sec ()`

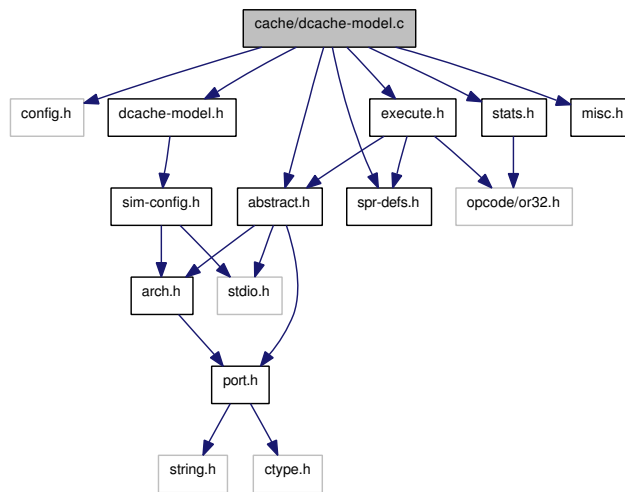
Here is the call graph for this function:



5.3 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]

5.3.1 Function Documentation

5.3.1.1 `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)

5.3.1.2 `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)

5.3.1.3 `void dc_info (void)`

5.3.1.4 `void dc_inv (oraddr_t dataaddr)`

5.3.1.5 `static void dc_load_hitdelay (union param_val val, void * dat)` [static]

5.3.1.6 `static void dc_load_missdelay (union param_val val, void * dat)` [static]

5.3.1.7 `static void dc_nsets (union param_val val, void * dat)` [static]

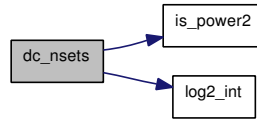
Set the number of data cache sets

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:



5.3.1.8 static void dc_nways (union param_val val, void * dat) [static]

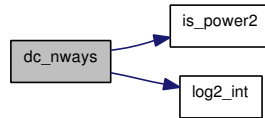
Set the number of data cache ways

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:

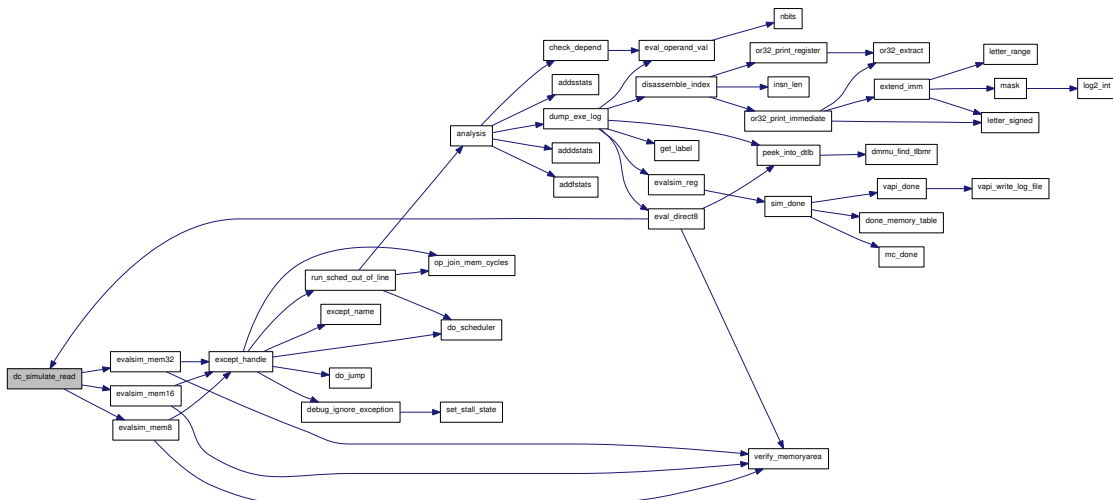
- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

Here is the call graph for this function:



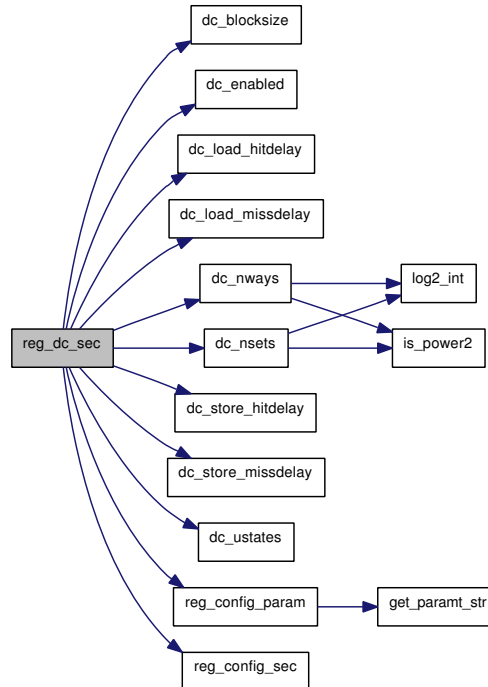
5.3.1.9 uint32_t dc_simulate_read (oraddr_t dataaddr, oraddr_t virt_addr, int width)

Here is the call graph for this function:



5.3.1.14 void reg_dc_sec (void)

Here is the call graph for this function:



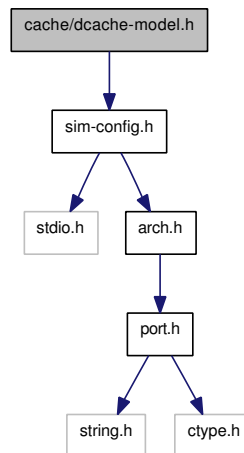
5.3.2 Variable Documentation

5.3.2.1 struct dc_set dc[MAX_DC_SETS]

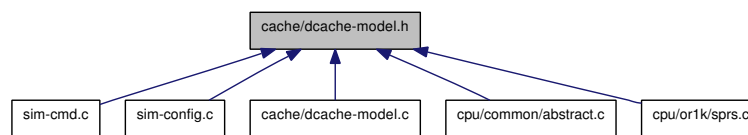
5.4 cache/dcache-model.h File Reference

```
#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](#) ()

5.4.1 Define Documentation

5.4.1.1 `#define MAX_DC_BLOCK_SIZE 32`

5.4.1.2 `#define MAX_DC_SETS 1024`

5.4.1.3 `#define MAX_DC_WAYS 32`

5.4.1.4 `#define MIN_DC_BLOCK_SIZE 16`

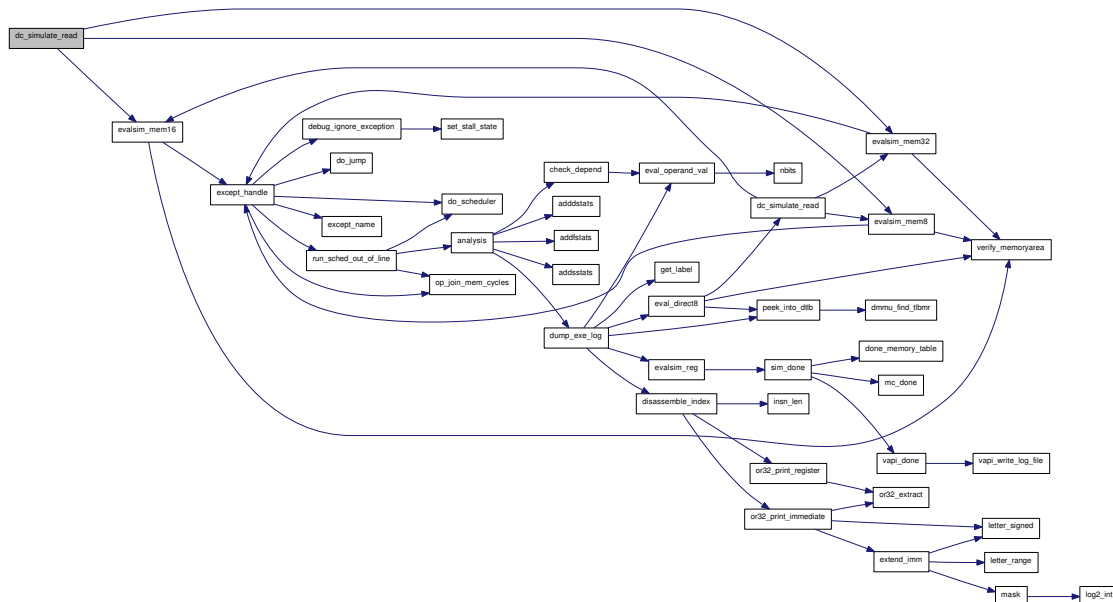
5.4.2 Function Documentation

5.4.2.1 `void dc_info ()`

5.4.2.2 `void dc_inv (oraddr_t dataaddr)`

5.4.2.3 `uint32_t dc_simulate_read (oraddr_t dataaddr, oraddr_t virt_addr, int width)`

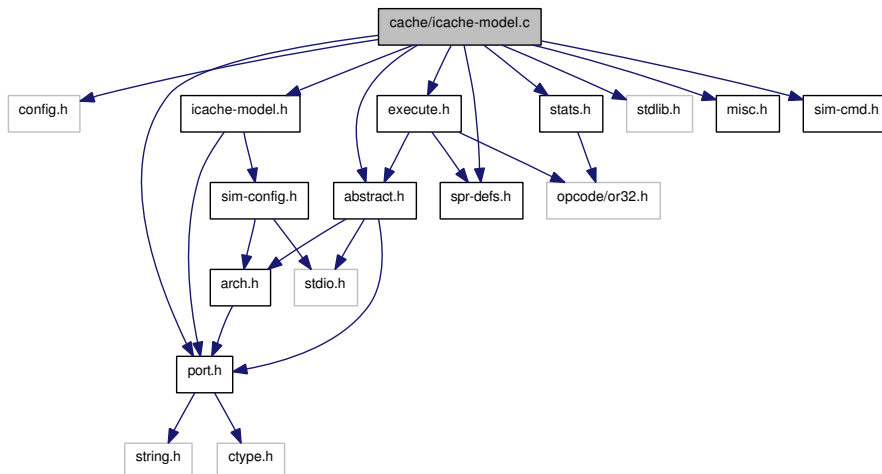
Here is the call graph for this function:



5.5 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

5.5.1 Define Documentation

5.5.1.1 `#define MAX_IC_BLOCK_SIZE 32`

5.5.1.2 `#define MAX_IC_SETS 1024`

5.5.1.3 `#define MAX_IC_WAYS 32`

5.5.1.4 `#define MIN_IC_BLOCK_SIZE 16`

5.5.2 Function Documentation

5.5.2.1 `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

5.5.2.2 `static void ic_enabled (union param_val val, void * dat)` [static]

Enable or disable the instruction cache

Set the corresponding field in the UPR

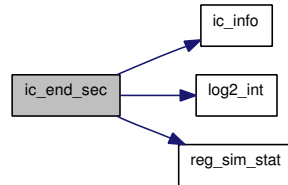
Parameters:

← *val* The value to use

← *dat* The `config` data structure

5.5.2.3 static void ic_end_sec (void * *dat*) [static]

Here is the call graph for this function:



5.5.2.4 static void ic_hitdelay (union param_val *val*, void * *dat*) [static]

5.5.2.5 static void ic_info (void * *dat*) [static]

5.5.2.6 void ic_inv (oraddr_t *dataaddr*)

5.5.2.7 static void ic_missdelay (union param_val *val*, void * *dat*) [static]

5.5.2.8 static void ic_nsets (union param_val *val*, void * *dat*) [static]

Set the number of instruction cache sets

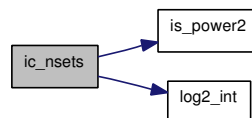
Set the corresponding field in the UPR

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



5.5.2.9 static void ic_nways (union param_val *val*, void * *dat*) [static]

Set the number of instruction cache ways

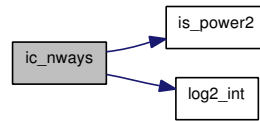
Set the corresponding field in the UPR

Parameters:

← *val* The value to use

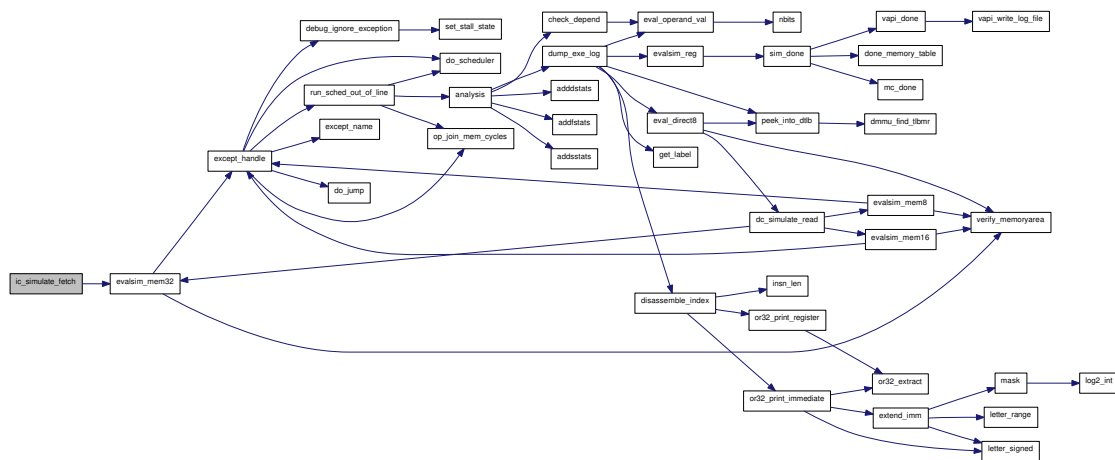
← *dat* The `config` data structure

Here is the call graph for this function:



5.5.2.10 uint32_t ic_simulate_fetch (oraddr_t fetchaddr, oraddr_t virt_addr)

Here is the call graph for this function:



5.5.2.11 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:



5.5.2.12 static void ic_ustates (union param_val val, void * dat) [static]

Set the number of instruction cache usage states

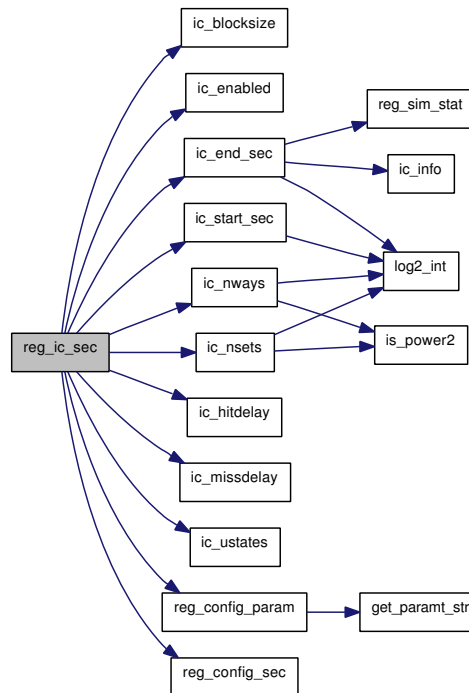
Value must be 2, 3 or 4. If not issue a warning and ignore.

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure

5.5.2.13 void reg_ic_sec (void)

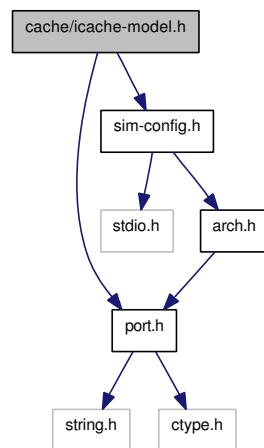
Here is the call graph for this function:

**5.5.3 Variable Documentation****5.5.3.1 struct ic* ic_state = NULL**

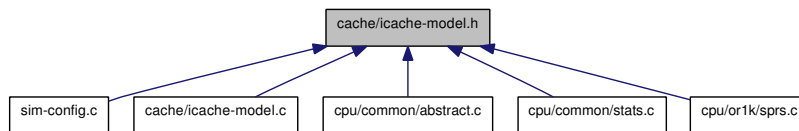
5.6 cache/icache-model.h File Reference

```
#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

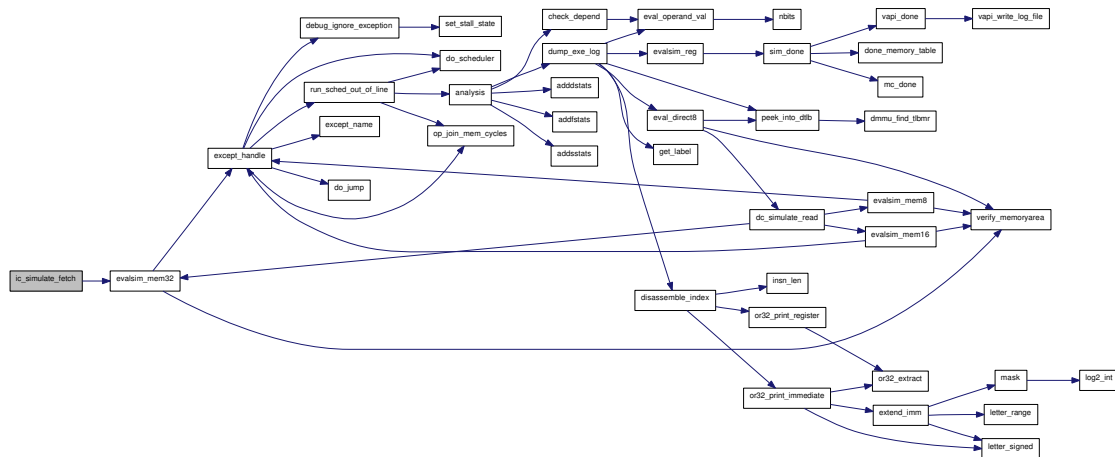
- struct `ic * ic_state`

5.6.1 Function Documentation

5.6.1.1 void ic_inv (oraddr_t dataaddr)

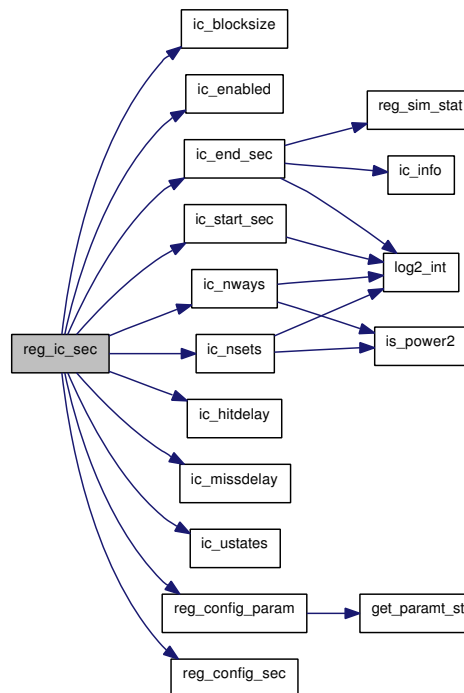
5.6.1.2 uint32_t ic_simulate_fetch (oraddr_t fetchaddr, oraddr_t virt_addr)

Here is the call graph for this function:



5.6.1.3 void reg_ic_sec ()

Here is the call graph for this function:



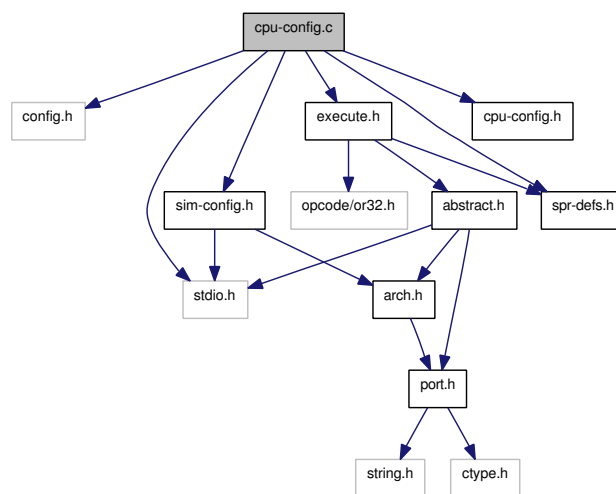
5.6.2 Variable Documentation

5.6.2.1 struct ic* ic_state

5.7 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](#) ()

5.7.1 Define Documentation

5.7.1.1 `#define WARNING(s) fprintf(stderr, "Warning: config.%s: %s\n", cur_section → name, (s))`

5.7.2 Function Documentation

5.7.2.1 `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)

5.7.2.2 `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)

5.7.2.3 `static void cpu_dependstats (union param_val val, void * dat) [static]`

5.7.2.4 `static void cpu_hazards (union param_val val, void * dat) [static]`

5.7.2.5 `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)

5.7.2.6 `static void cpu_sbuf_len (union param_val val, void * dat) [static]`

5.7.2.7 `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)

5.7.2.8 `static void cpu_superscalar (union param_val val, void * dat) [static]`

5.7.2.9 `static void cpu_upr (union param_val val, void * dat) [static]`

5.7.2.10 `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

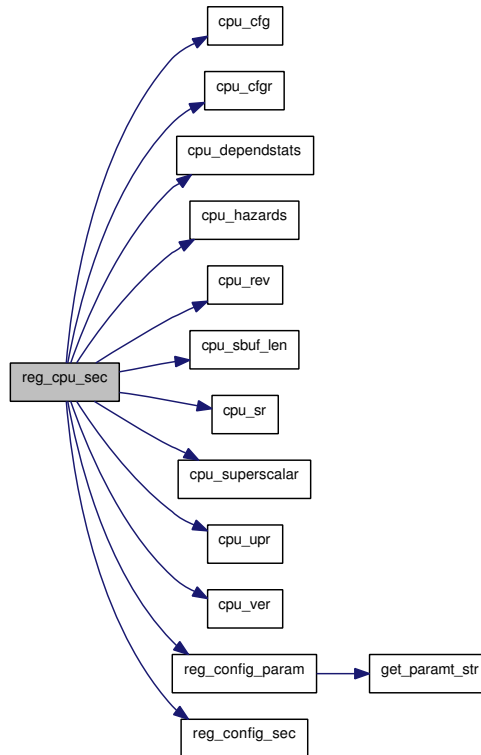
← *dat* The [config](#) data structure (not used here)

5.7.2.11 `void reg_cpu_sec ()`

Register the functions to handle a section cpu

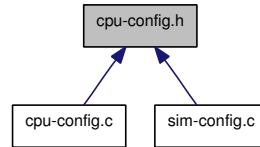
This section does not allocate dynamically a data structure holding its [config](#) information. It's all in the global [config.sim](#) data structure. Therefore it does not need a start and end function to initialize default values (although it might be clearer to do so). The default values are set in [init_defconfig\(\)](#).

Here is the call graph for this function:



5.8 cpu-config.h File Reference

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



Functions

- void [reg_cpu_sec](#) ()

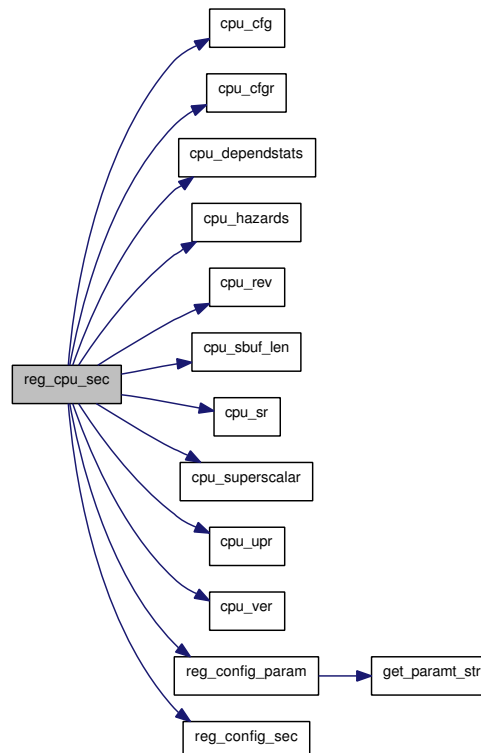
5.8.1 Function Documentation

5.8.1.1 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](#)().

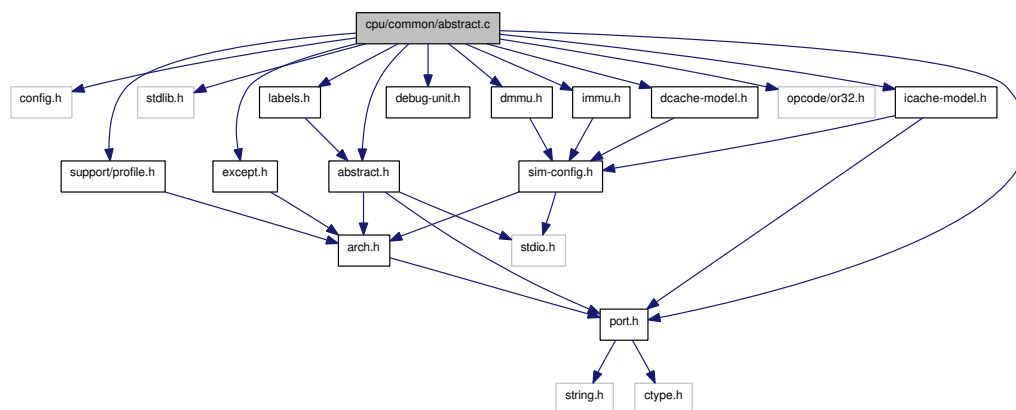
Here is the call graph for this function:



5.9 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)

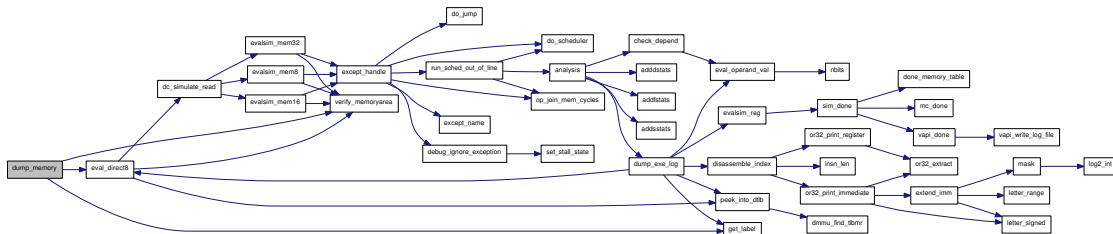
Variables

- 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`

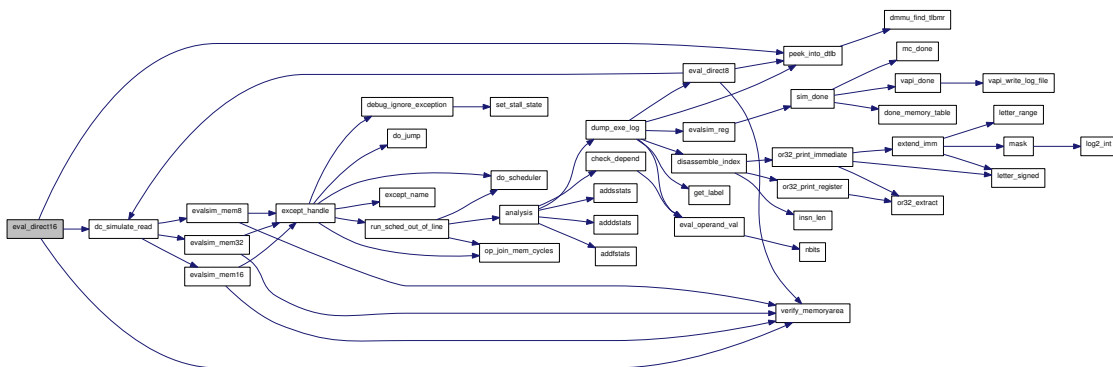
Parameters:

- ← *from* Start address of the area of memory
- ← *to* End address of the area of memory

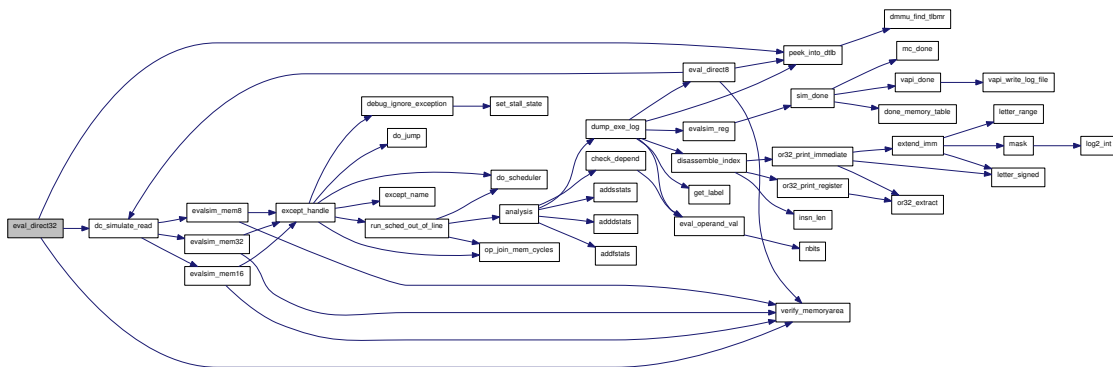
Here is the call graph for this function:

**5.9.1.6 uint16_t eval_direct16 (oraddr_t memaddr, int through_mmu, int through_dc)**

Here is the call graph for this function:

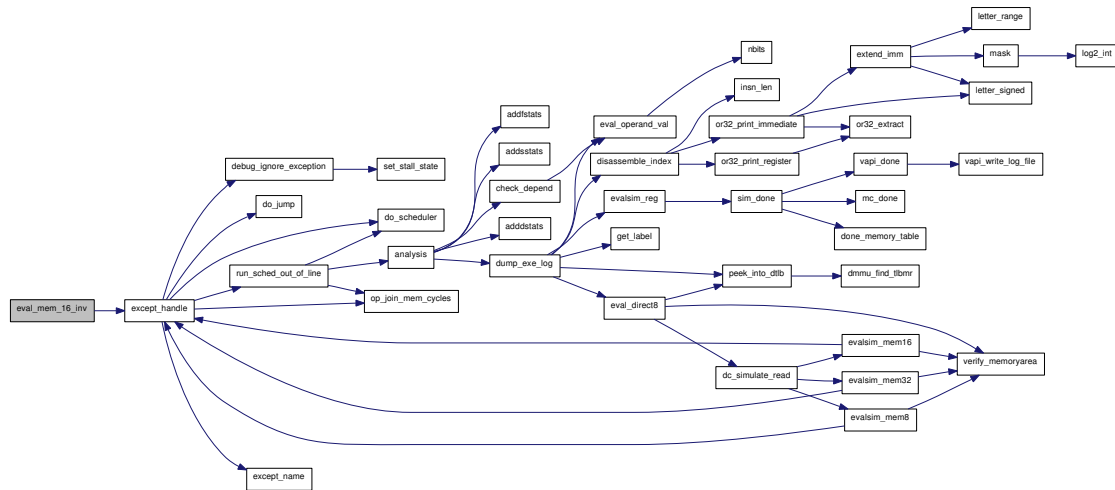
**5.9.1.7 uint32_t eval_direct32 (oraddr_t memaddr, int through_mmu, int through_dc)**

Here is the call graph for this function:



5.9.1.13 static uint16_t eval_mem_16_inv (oraddr_t memaddr, void * dat) [static]

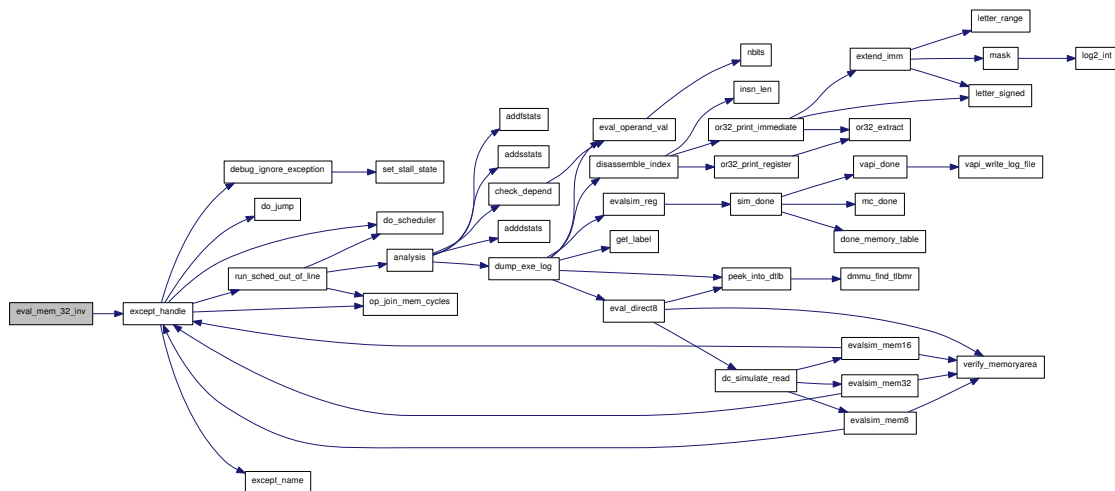
Here is the call graph for this function:



5.9.1.14 uint16_t eval_mem_16_inv_direct (oraddr_t memaddr, void * dat) [static]

5.9.1.15 static uint32_t eval_mem_32_inv (oraddr_t memaddr, void * dat) [static]

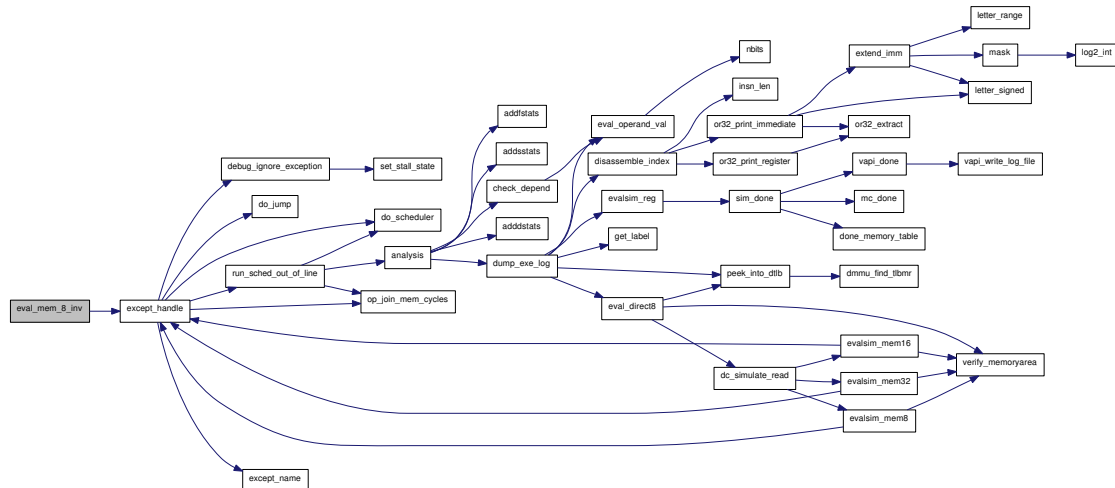
Here is the call graph for this function:



5.9.1.16 `uint32_t eval_mem_32_inv_direct (oraddr_t memaddr, void * dat)` [static]

5.9.1.17 `static uint8_t eval_mem_8_inv (oraddr_t memaddr, void * dat)` [static]

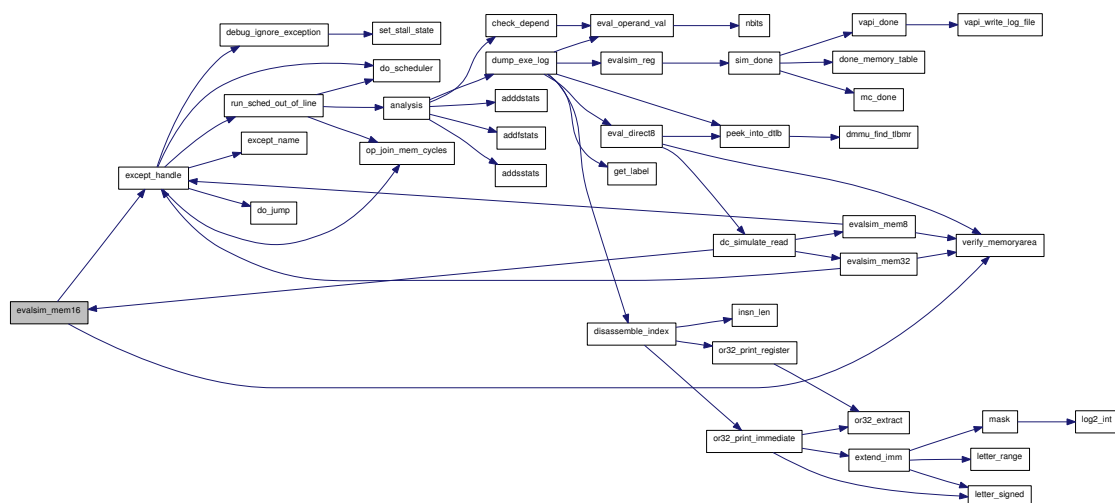
Here is the call graph for this function:



5.9.1.18 `uint8_t eval_mem_8_inv_direct (oraddr_t memaddr, void * dat)` [static]

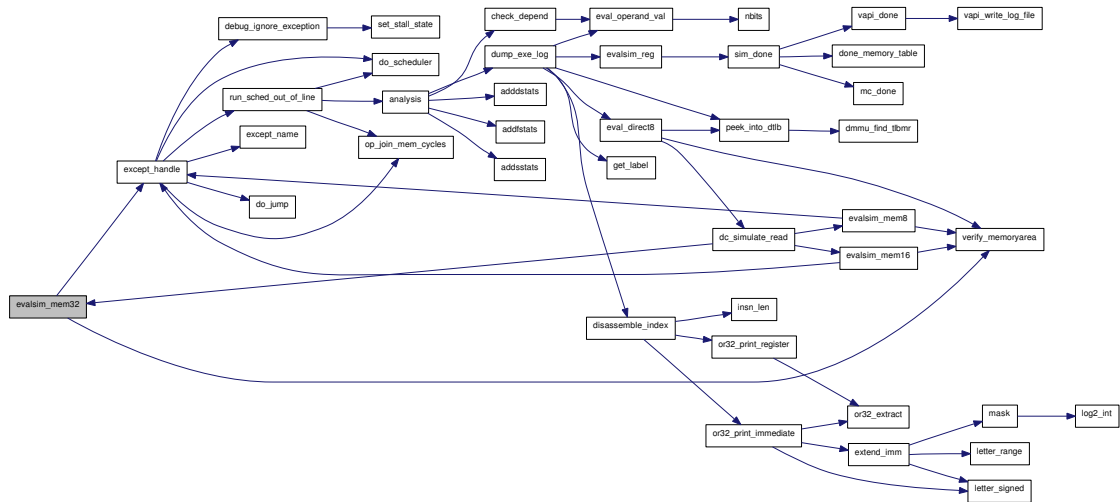
5.9.1.19 `uint16_t evalsim_mem16 (oraddr_t memaddr, oraddr_t vaddr)`

Here is the call graph for this function:



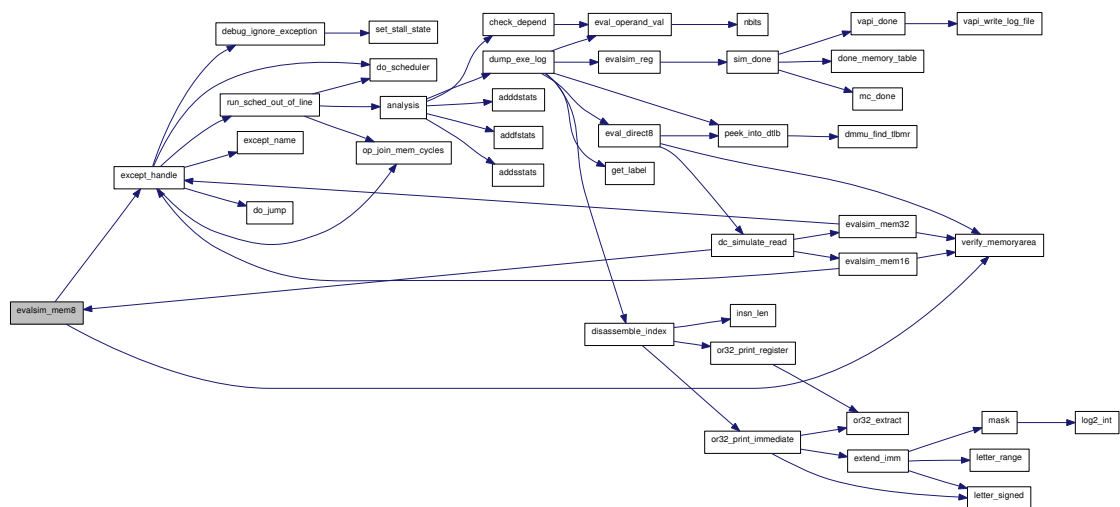
5.9.1.20 uint32_t evalsim_mem32 (oraddr_t memaddr, oraddr_t vaddr)

Here is the call graph for this function:



5.9.1.21 uint8_t evalsim_mem8 (oraddr_t memaddr, oraddr_t vaddr)

Here is the call graph for this function:



5.9.1.22 `char* generate_time_pretty (char * dest, long time_ps)`

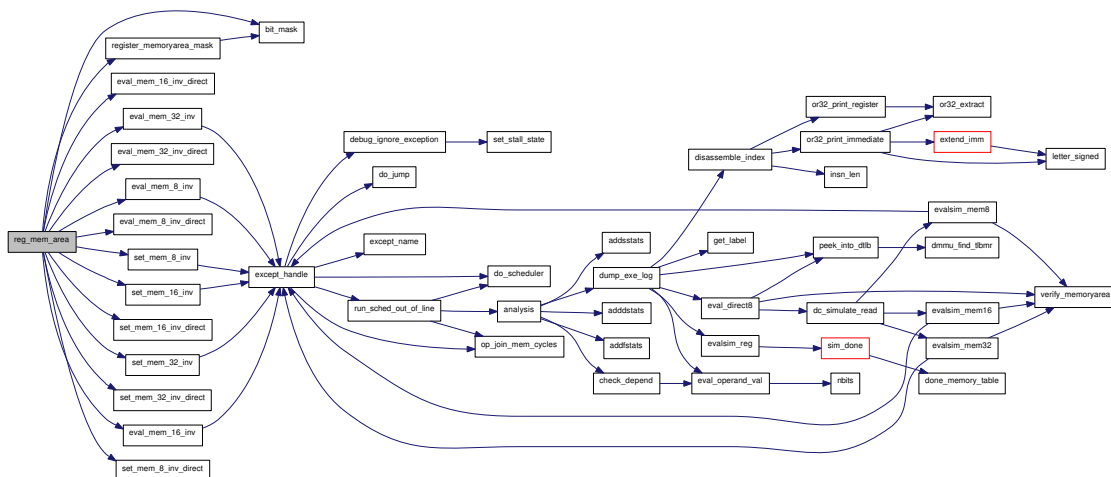
5.9.1.23 `void memory_table_status (void)`

Here is the call graph for this function:



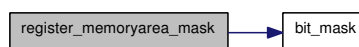
5.9.1.24 `struct dev_memarea* reg_mem_area (oraddr_t addr, uint32_t size, unsigned mc_dev, struct mem_ops * ops) [read]`

Here is the call graph for this function:



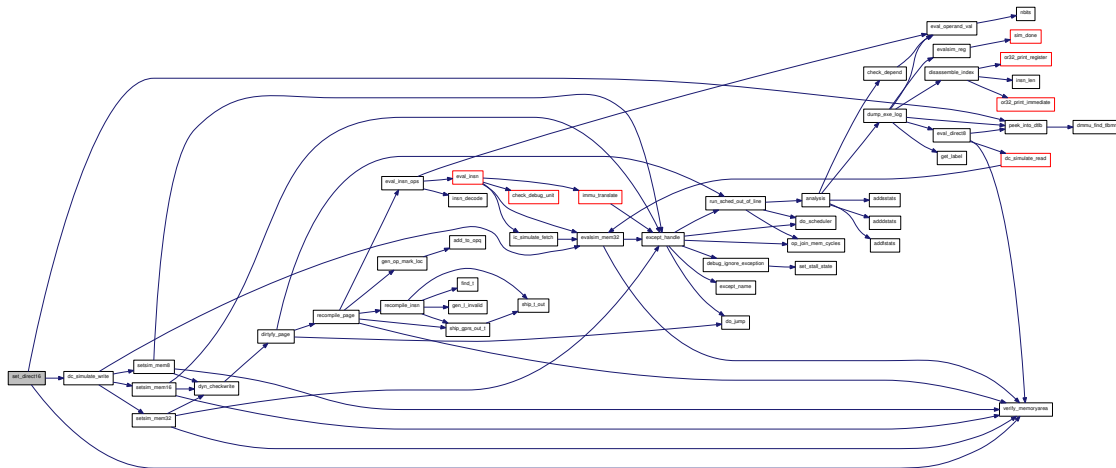
5.9.1.25 `static struct dev_memarea* register_memoryarea_mask (oraddr_t addr_mask, oraddr_t addr_compare, uint32_t size, unsigned mc_dev) [static, read]`

Here is the call graph for this function:



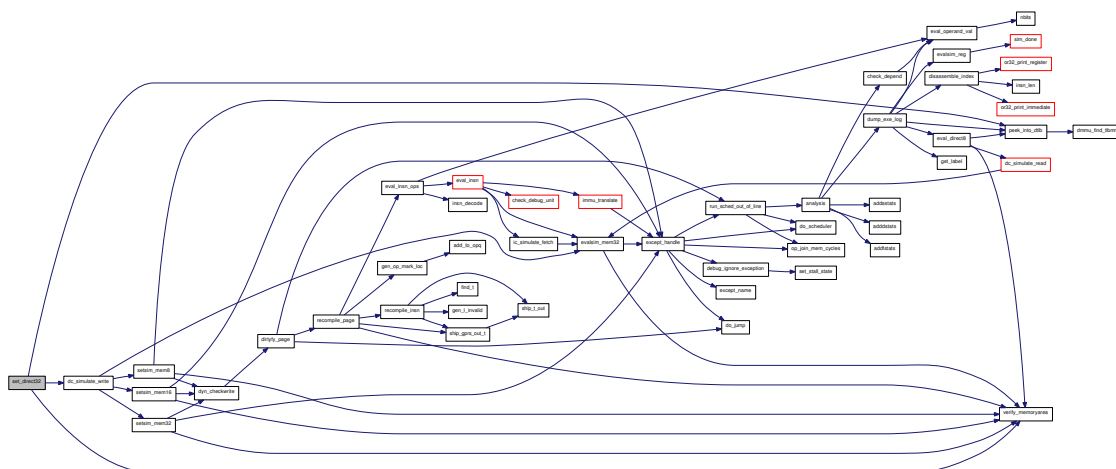
5.9.1.26 void set_direct16 (oraddr_t memaddr, uint16_t value, int through_mmu, int through_dc)

Here is the call graph for this function:



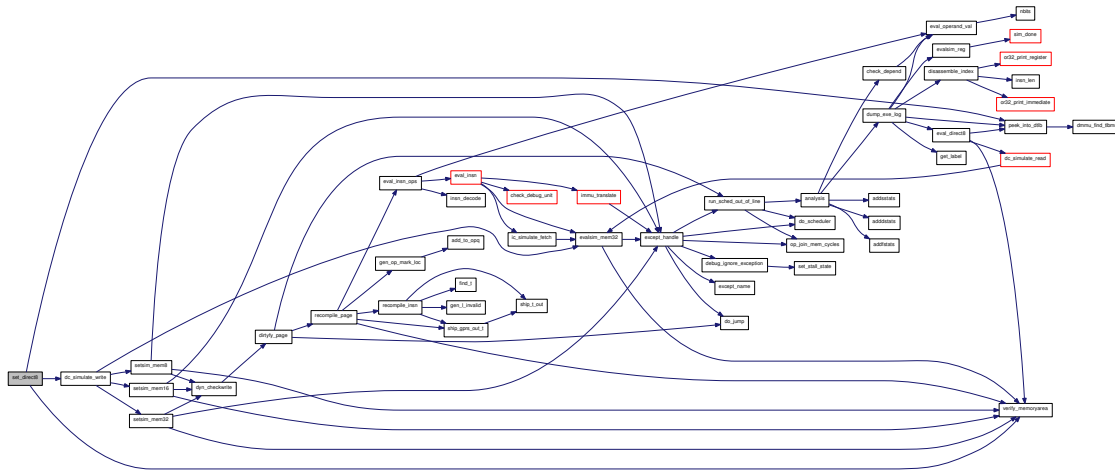
5.9.1.27 void set_direct32 (oraddr_t memaddr, uint32_t value, int through_mmu, int through_dc)

Here is the call graph for this function:



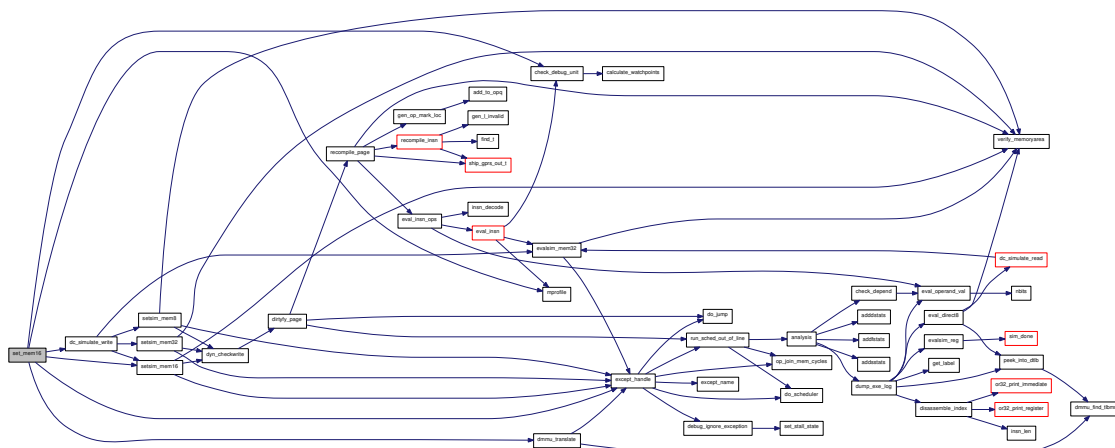
5.9.1.28 void set_direct8 (oraddr_t memaddr, uint8_t value, int through_mmu, int through_dc)

Here is the call graph for this function:



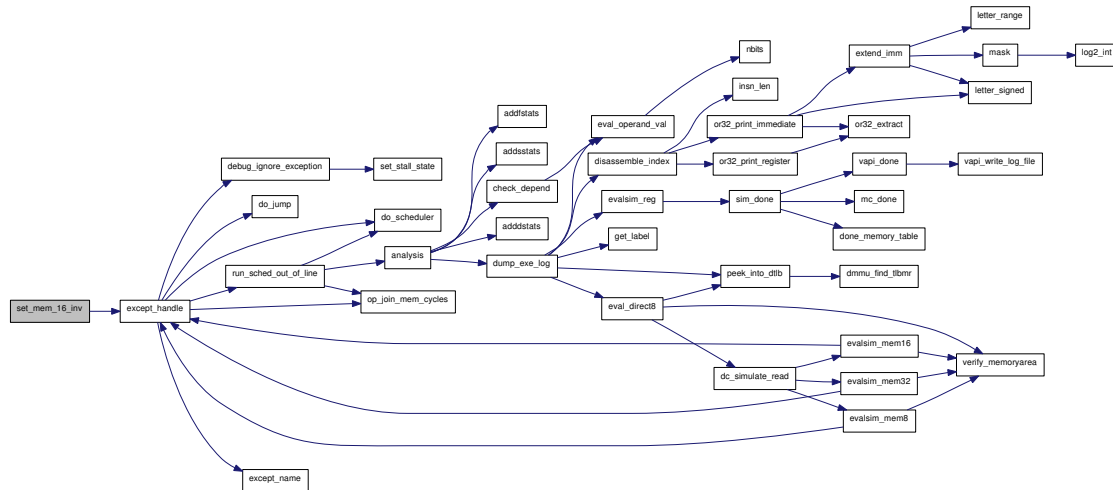
5.9.1.29 void set_mem16 (oraddr_t memaddr, uint16_t value, int * breakpoint)

Here is the call graph for this function:



5.9.1.32 static void set_mem_16_inv (oraddr_t memaddr, uint16_t val, void * dat) [static]

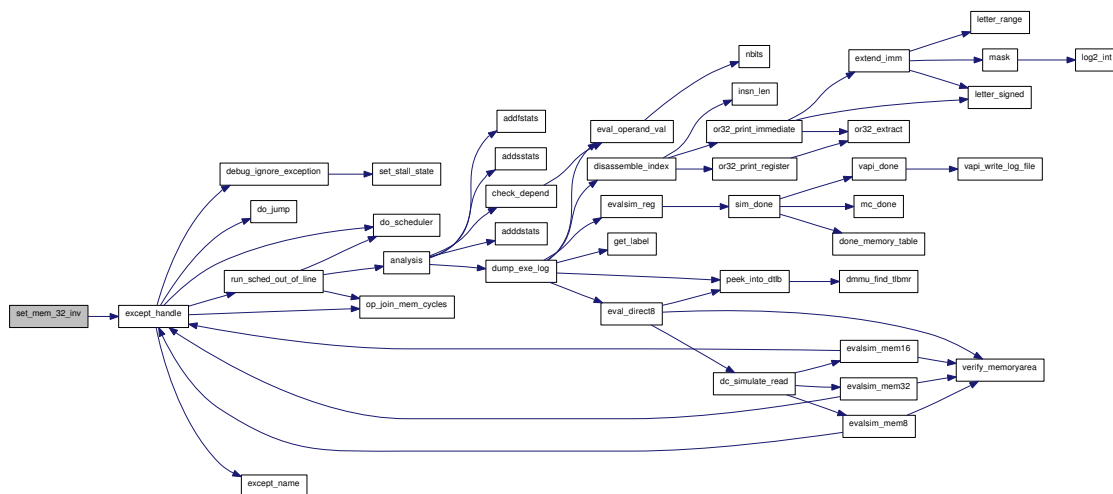
Here is the call graph for this function:



5.9.1.33 void set_mem_16_inv_direct (oraddr_t memaddr, uint16_t val, void * dat) [static]

5.9.1.34 static void set_mem_32_inv (oraddr_t memaddr, uint32_t val, void * dat) [static]

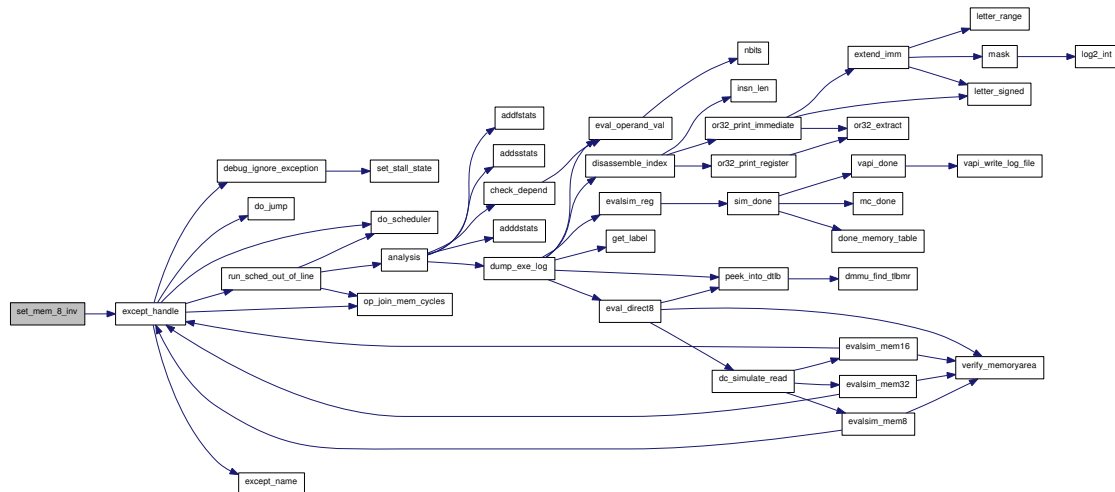
Here is the call graph for this function:



5.9.1.35 void `set_mem_32_inv_direct` (`oraddr_t memaddr`, `uint32_t val`, `void * dat`) [static]

5.9.1.36 static void `set_mem_8_inv` (`oraddr_t memaddr`, `uint8_t val`, `void * dat`) [static]

Here is the call graph for this function:



5.9.1.37 void `set_mem_8_inv_direct` (`oraddr_t memaddr`, `uint8_t val`, `void * dat`) [static]

5.9.1.38 void `set_mem_valid` (`struct dev_memarea * mem`, `int valid`)

5.9.1.39 void `set_program32` (`oraddr_t memaddr`, `uint32_t value`)

Here is the call graph for this function:



5.9.1.40 void `set_program8` (`oraddr_t memaddr`, `uint8_t value`)

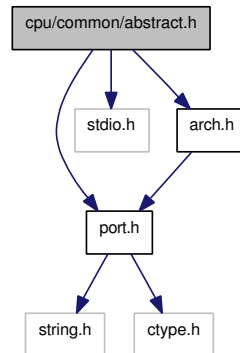
Here is the call graph for this function:



5.10 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`

5.10.1 Define Documentation

- 5.10.1.1 `#define CT_NONE 0`
- 5.10.1.2 `#define CT_PHYSICAL 2`
- 5.10.1.3 `#define CT_VIRTUAL 1`
- 5.10.1.4 `#define DEFAULT_MEMORY_LEN 0x800000`
- 5.10.1.5 `#define DEFAULT_MEMORY_START 0`
- 5.10.1.6 `#define HISTEXEC_LEN 200`
- 5.10.1.7 `#define INSNAME_LEN 15`
- 5.10.1.8 `#define LABELNAME_LEN 50`
- 5.10.1.9 `#define LE16(x) bswap_16(x)`
- 5.10.1.10 `#define LONGEST long long`
- 5.10.1.11 `#define MAX_OPERANDS 5`
- 5.10.1.12 `#define OP_MEM_ACCESS 0x80000000`
- 5.10.1.13 `#define OPERANDNAME_LEN 50`
- 5.10.1.14 `#define STACK_SIZE 20`
- 5.10.1.15 `#define ULONGEST unsigned long long`

5.10.2 Function Documentation

- 5.10.2.1 `void adjust_rw_delay (struct dev_memarea * mem, int delayr, int delayw)`
- 5.10.2.2 `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.

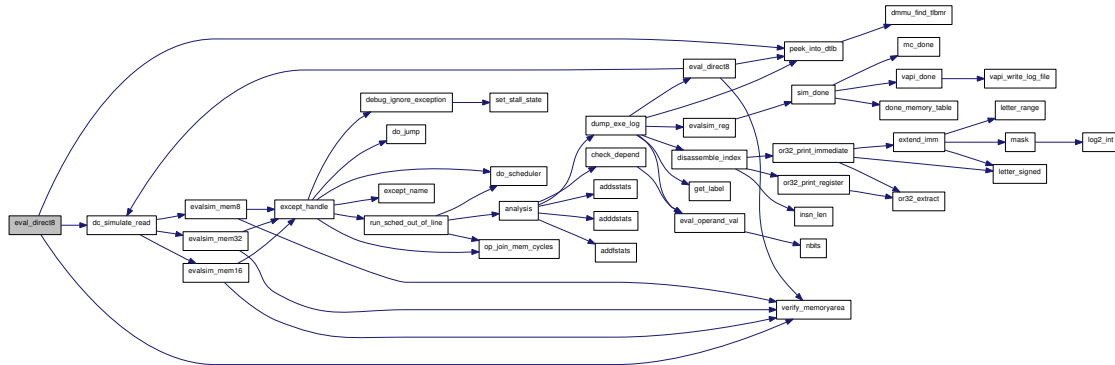
There are all sorts of ways to trip this up, but they are unlikely. The validity of a memory area is taken from the address of the start of a line to be printed, so assumes the following 3 bytes are present. This could be fooled by ridiculous memory declarations.

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

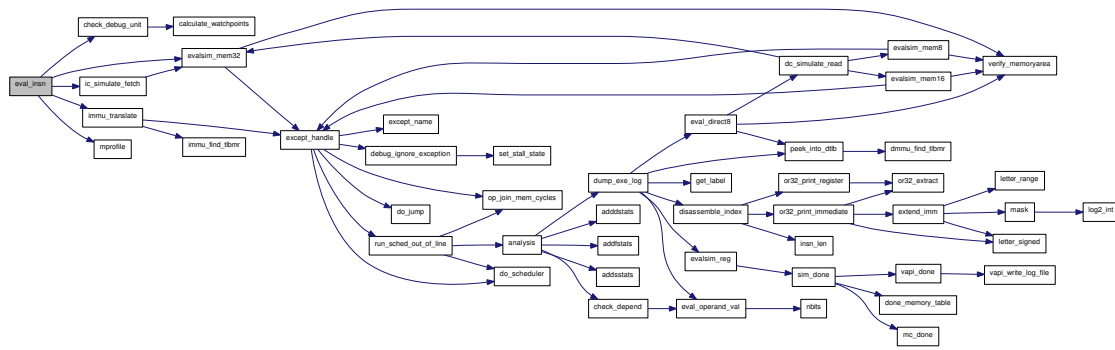
5.10.2.7 uint8_t eval_direct8 (oraddr_t memaddr, int through_mmu, int through_dc)

Here is the call graph for this function:



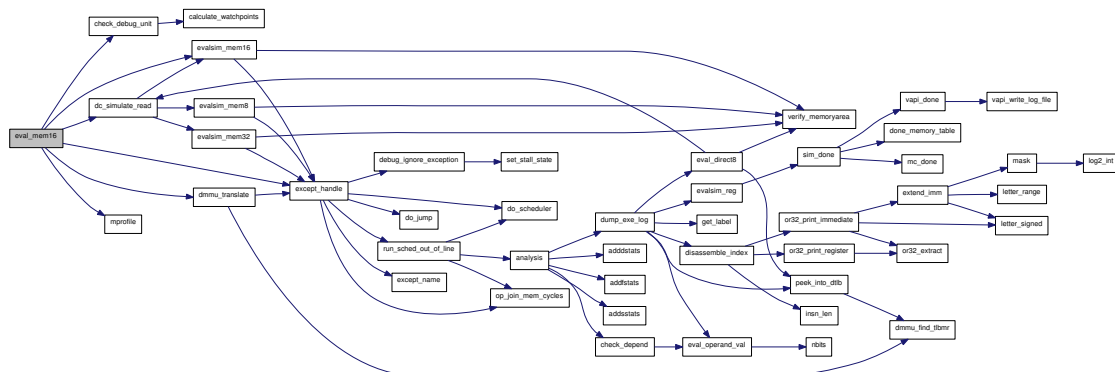
5.10.2.8 uint32_t eval_insn (oraddr_t, int *)

Here is the call graph for this function:



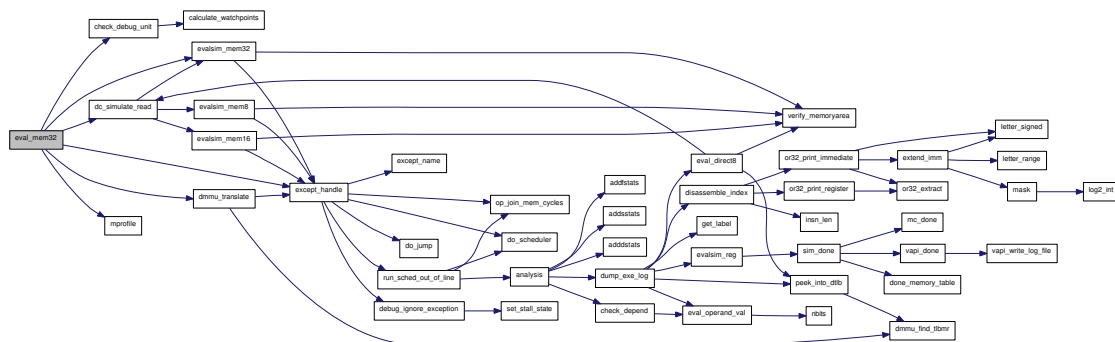
5.10.2.9 uint16_t eval_mem16 (oraddr_t memaddr, int *)

Here is the call graph for this function:



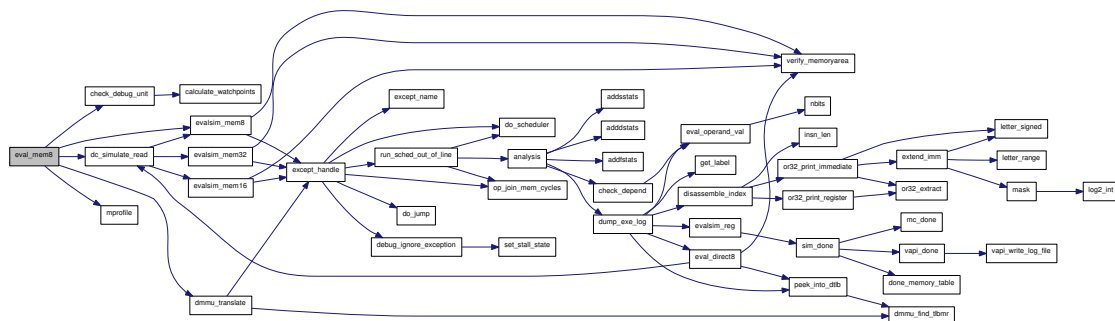
5.10.2.10 uint32_t eval_mem32 (oraddr_t memaddr, int *)

Here is the call graph for this function:



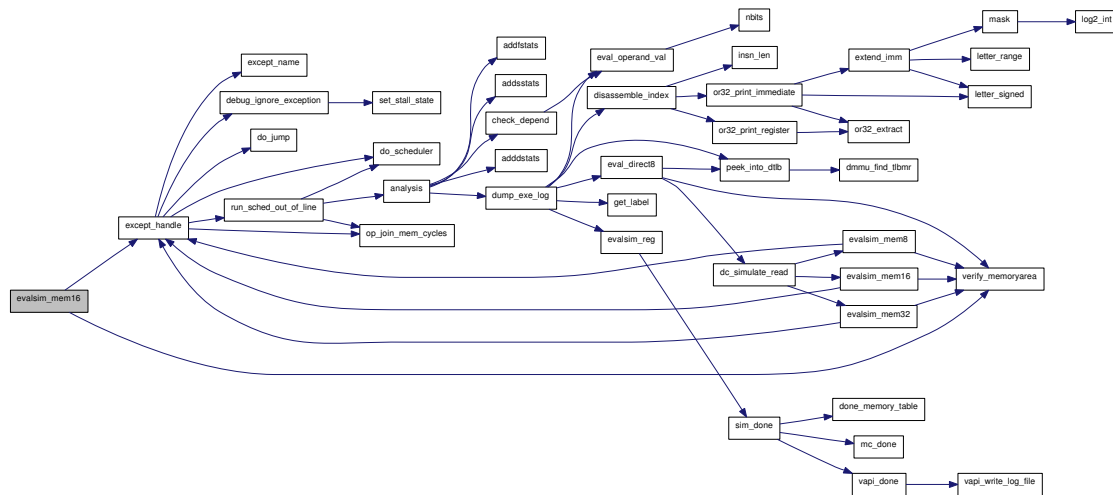
5.10.2.11 uint8_t eval_mem8 (oraddr_t memaddr, int *)

Here is the call graph for this function:



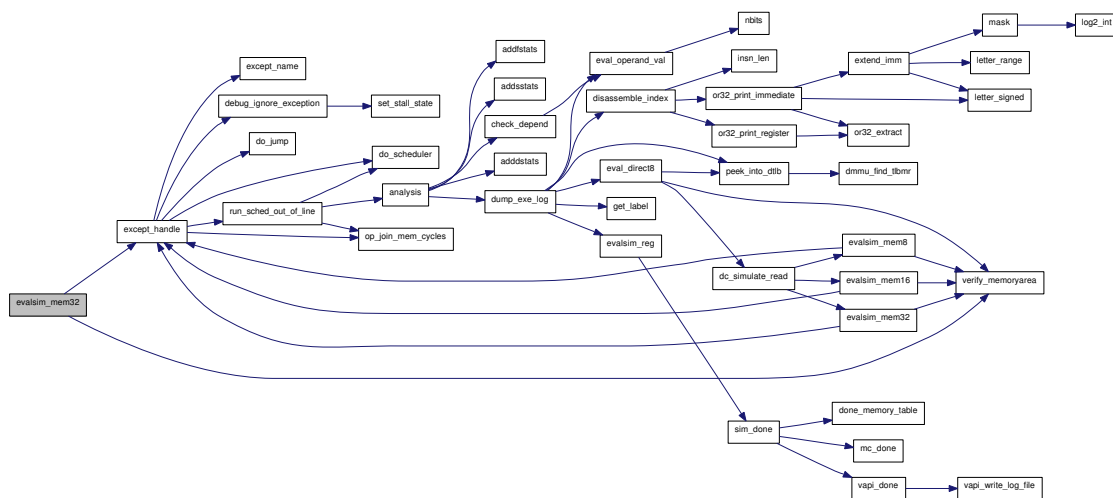
5.10.2.12 uint16_t evalsim_mem16 (oraddr_t, oraddr_t)

Here is the call graph for this function:



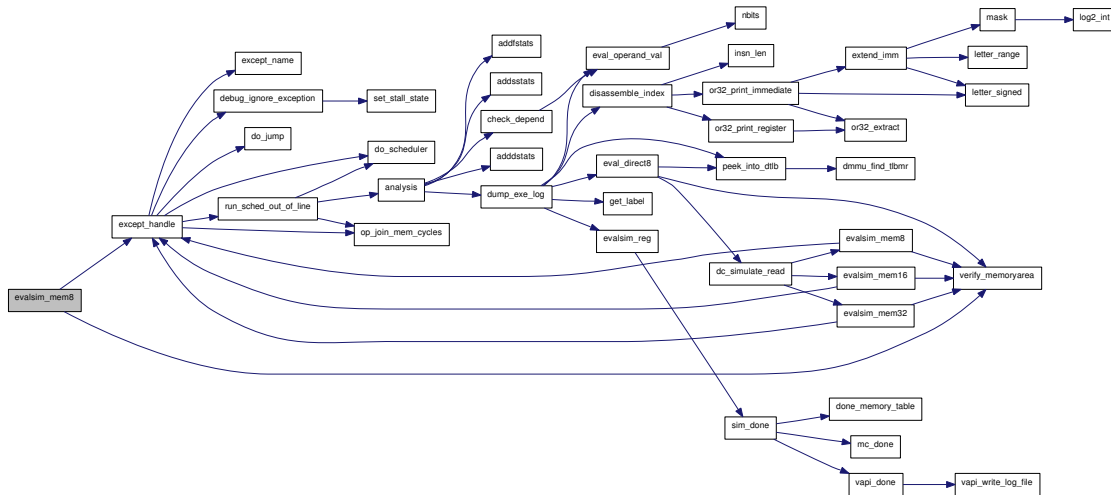
5.10.2.13 uint32_t evalsim_mem32 (oraddr_t, oraddr_t)

Here is the call graph for this function:



5.10.2.14 uint8_t evalsim_mem8 (oraddr_t, oraddr_t)

Here is the call graph for this function:



5.10.2.15 char* generate_time_pretty (char * dest, long time_ps)

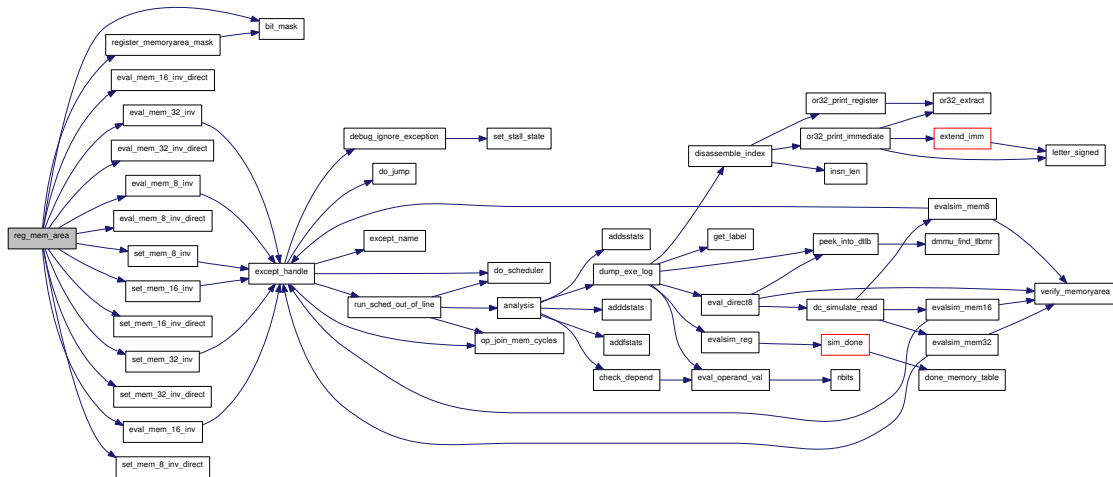
5.10.2.16 void memory_table_status (void)

Here is the call graph for this function:



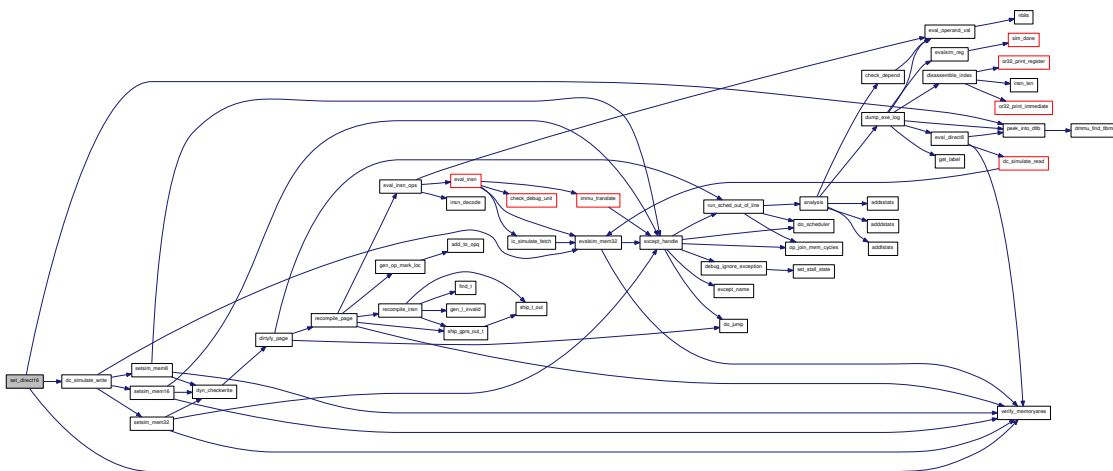
5.10.2.17 struct dev_memarea* reg_mem_area (oraddr_t addr, uint32_t size, unsigned mc_dev, struct mem_ops * ops) [read]

Here is the call graph for this function:



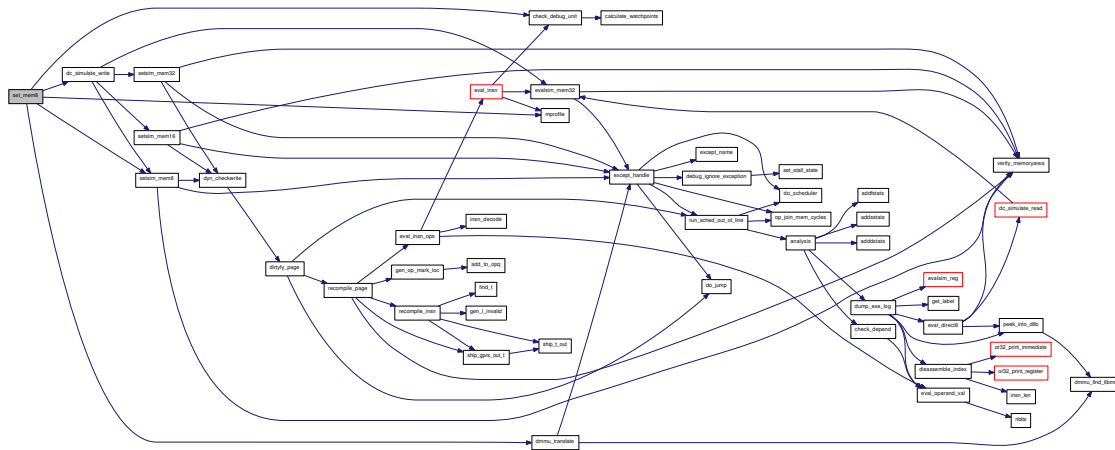
5.10.2.18 void set_direct16 (oraddr_t, uint16_t, int, int)

Here is the call graph for this function:



5.10.2.23 void set_mem8 (oraddr_t memaddr, uint8_t value, int *)

Here is the call graph for this function:



5.10.2.24 void set_mem_valid (struct dev_memarea * mem, int valid)

5.10.2.25 void set_program32 (oraddr_t memaddr, uint32_t value)

Here is the call graph for this function:



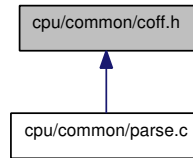
5.10.2.26 void set_program8 (oraddr_t memaddr, uint8_t value)

Here is the call graph for this function:



5.11 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_symtab](#)
- 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

5.11.1 Define Documentation

5.11.1.1 `#define C_ALIAS 105`

5.11.1.2 `#define C_ARG 9`

5.11.1.3 `#define C_AUTO 1`

5.11.1.4 `#define C_AUTOARG 19`

5.11.1.5 `#define C_BCOMM (0x87)`

5.11.1.6 `#define C_BINCL 108`

5.11.1.7 `#define C_BLOCK 100`

5.11.1.8 `#define C_BSTAT (0x8f)`

5.11.1.9 `#define C_DECL (0x8c)`

5.11.1.10 `#define C_DEFINE 110`

5.11.1.11 `#define C_ECOML (0x88)`

5.11.1.12 `#define C_ECOMM (0x89)`

5.11.1.13 `#define C_EFCN 0xff`

5.11.1.14 `#define C_EINCL 109`

5.11.1.15 `#define C_ENTAG 15`

5.11.1.16 `#define C_ENTRY (0x8d)`

5.11.1.17 `#define C_EOS 102`

5.11.1.18 `#define C_ESTAT (0x90)`

5.11.1.19 `#define C_EXT 2`

5.11.1.20 `#define C_EXTDEF 5`

5.11.1.21 `#define C_EXTLAB 21`

5.11.1.22 `#define C_FCN 101`

5.11.1.23 `#define C_FIELD 18`

5.11.1.24 `#define C_FILE 103`

5.11.1.25 `#define C_FUN (0x8e)`

5.11.1.26 `#define C_GSYM (0x80)`

5.11.1.27 `#define C_HIDDEN 106`

5.11.1.28 `#define C_HIDEXT 107`

5.11.1.29 `#define C_LABEL 6`

5.11.1.30 `#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])))))
```

5.11.1.148 #define KEEP_ENDIAN_SHORT(ps)

Value:

```
((short) (((unsigned short) ((unsigned char)ps[0])<<8) |\
        ((unsigned short) ((unsigned char)ps[1]))))
```

5.11.1.149 #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]))))
```

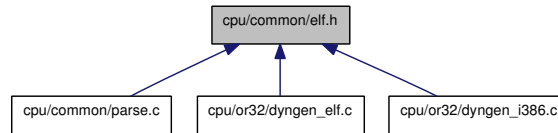
5.11.1.150 #define SWAP_ENDIAN_SHORT(ps)

Value:

```
((short) (((unsigned short) ((unsigned char)ps[1])<<8) |\
        ((unsigned short) ((unsigned char)ps[0]))))
```

5.12 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` 0xfff1
- #define `SHN_COMMON` 0xfff2
- #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](#) []

5.12.1 Define Documentation

5.12.1.1 **#define AT_BASE 7**

5.12.1.2 **#define AT_EGID 14**

5.12.1.3 **#define AT_ENTRY 9**

5.12.1.4 **#define AT_EUID 12**

5.12.1.5 **#define AT_EXECFD 2**

5.12.1.6 **#define AT_FLAGS 8**

5.12.1.7 **#define AT_GID 13**

5.12.1.8 **#define AT_IGNORE 1**

5.12.1.9 **#define AT_NOTELF 10**

5.12.1.10 **#define AT_NULL 0**

5.12.1.11 **#define AT_PAGESZ 6**

5.12.1.12 **#define AT_PHDR 3**

5.12.1.13 **#define AT_PHEMT 4**

5.12.1.14 **#define AT_PHNUM 5**

5.12.1.15 **#define AT_UID 11**

5.12.1.16 **#define DT_DEBUG 21**

5.12.1.17 **#define DT_FINI 13**

5.12.1.18 **#define DT_HASH 4**

5.12.1.19 **#define DT_HIPROC 0x7fffffff**

5.12.1.20 **#define DT_INIT 12**

5.12.1.21 **#define DT_JMPREL 23**

5.12.1.22 **#define DT_LOPROC 0x70000000**

5.12.1.23 **#define DT_NEEDED 1**

5.12.1.24 **#define DT_NULL 0**

5.12.1.25 **#define DT_PLTGOT 3**

5.12.1.26 **#define DT_PLTREL 20**

5.12.1.27 **#define DT_PLTRELSZ 2**

5.12.1.28 **#define DT_REL 17**

5.12.1.29 **#define DT_RELA 7**

5.12.1.30 **#define DT_RELAENT 9**

```
(((unsigned long) (ps) >> 24) & 0xff) |\
    ((unsigned long) (ps) >> 8) & 0xff00) |\
    ((unsigned long) (ps) << 8) & 0xff0000) |\
    ((unsigned long) (ps) << 24) & 0xff000000))
```

5.12.1.56 #define elf_note elf64_note

5.12.1.57 #define elf_phdr elf64_phdr

5.12.1.58 #define ELF_SHORT_H(ps)

Value:

```
(((unsigned short) (ps) >> 8) & 0xff) |\
    ((unsigned short) (ps) << 8) & 0xff00))
```


- 5.12.1.59 **#define ELFCLASS32 1**
- 5.12.1.60 **#define ELFCLASS64 2**
- 5.12.1.61 **#define ELFCLASSNONE 0**
- 5.12.1.62 **#define ELFCLASSNUM 3**
- 5.12.1.63 **#define ELFDATA2LSB 1**
- 5.12.1.64 **#define ELFDATA2MSB 2**
- 5.12.1.65 **#define ELFDATANONE 0**
- 5.12.1.66 **#define elfhdr elf64_hdr**
- 5.12.1.67 **#define ELFMAG "\177ELF"**
- 5.12.1.68 **#define ELFMAG0 0x7f**
- 5.12.1.69 **#define ELFMAG1 'E'**
- 5.12.1.70 **#define ELFMAG2 'L'**
- 5.12.1.71 **#define ELFMAG3 'F'**
- 5.12.1.72 **#define EM_386 3**
- 5.12.1.73 **#define EM_486 6**
- 5.12.1.74 **#define EM_68K 4**
- 5.12.1.75 **#define EM_860 7**
- 5.12.1.76 **#define EM_88K 5**
- 5.12.1.77 **#define EM_ALPHA 0x9026**
- 5.12.1.78 **#define EM_M32 1**
- 5.12.1.79 **#define EM_MIPS 8**
- 5.12.1.80 **#define EM_MIPS_RS4_BE 10**
- 5.12.1.81 **#define EM_NONE 0**
- 5.12.1.82 **#define EM_PARISC 15**
- 5.12.1.83 **#define EM_PPC 20**
- 5.12.1.84 **#define EM_SPARC 2**
- 5.12.1.85 **#define EM_SPARC32PLUS 18**

5.12.1.86 **#define EM_SPARC64 41**
Generated on Sun Oct 12 09:49:02 2008 for OpenRISC: The OpenRISC 1000 Architectural Simulator by Doxygen

5.12.1.87 **#define ET_CORE 4**

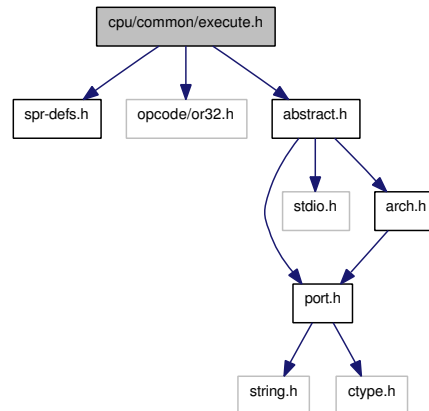
5.12.1.88 **#define ET_DYN 3**

5.12.1.89 **#define ET_EXEC 2**

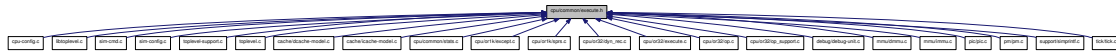
5.13 cpu/common/execute.h File Reference

```
#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)

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`

5.13.1 Define Documentation

5.13.1.1 `#define CURINSN(INSN) (strcmp(cur → insn, (INSN)) == 0)`

5.13.2 Function Documentation

5.13.2.1 `void analysis (struct iqueue_entry * current)`

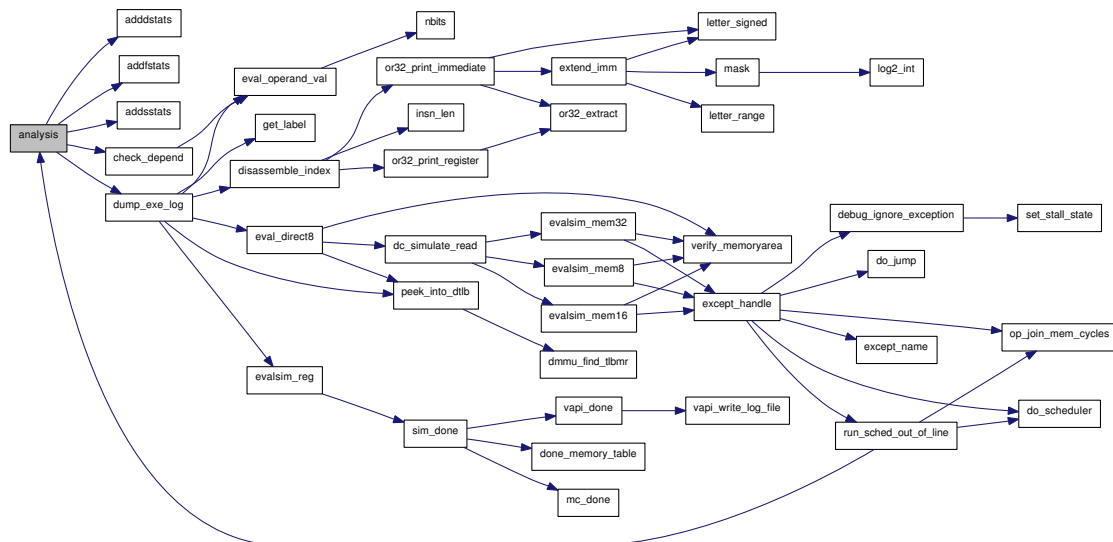
Perform analysis of the instruction being executed

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:



5.13.2.2 `int cpu_clock ()`

Simulates one CPU clock cycle

Returns:

non-zero if a breakpoint is hit, zero otherwise.

Parameters:

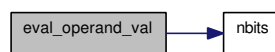
← *insn* The instruction

← *opd* The operand

Returns:

The value of the source operand

Here is the call graph for this function:

**5.13.2.8 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\(\)](#).

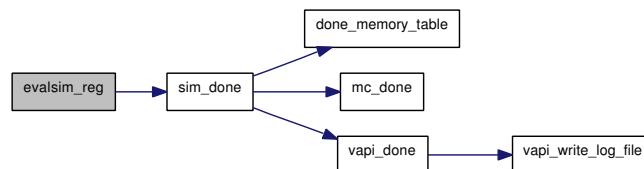
Parameters:

← *regno* The register of interest

Returns:

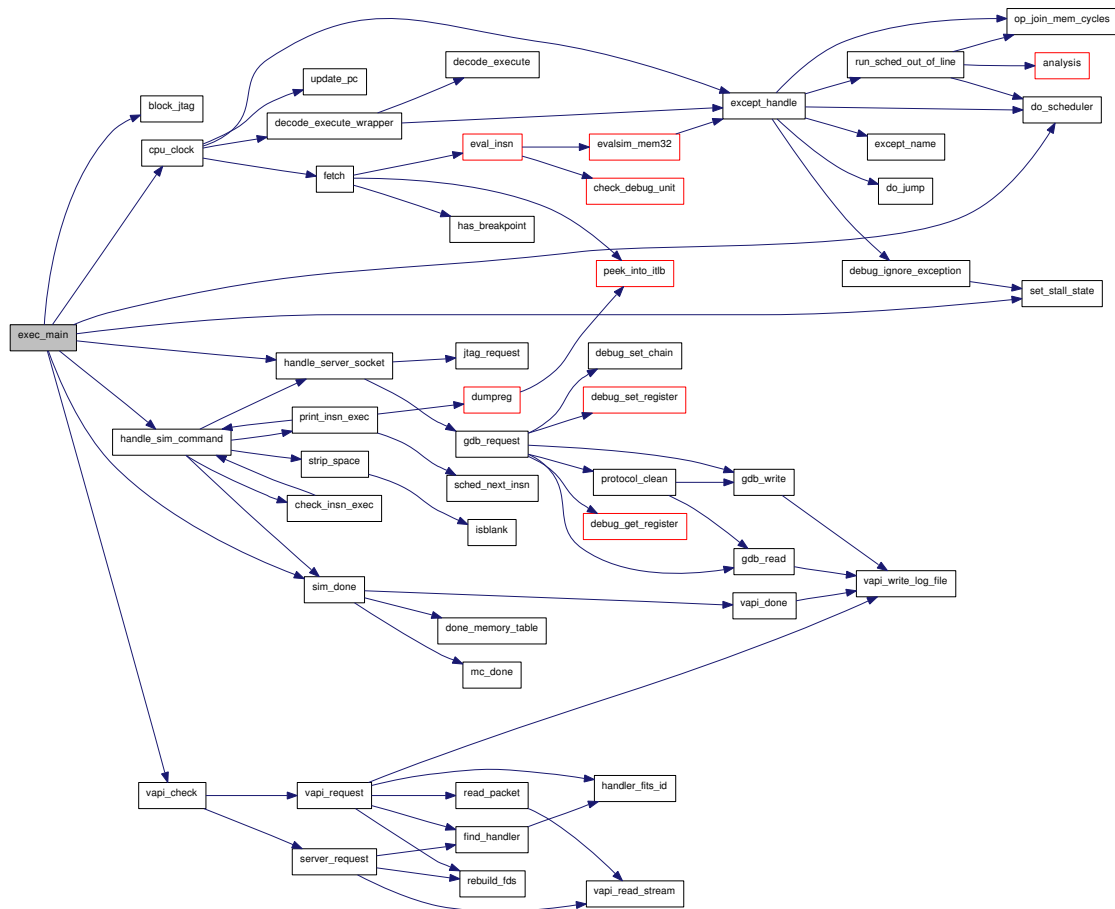
The value of the register

Here is the call graph for this function:

**5.13.2.9 void exec_main ()**

The main execution loop

Here is the call graph for this function:



5.13.2.10 void setsim_reg (unsigned int regno, uorreg_t value)

Set a specific register with value

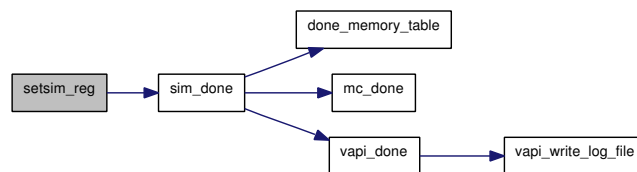
Implementation specific. Abort if we are given a duff register.

Parameters:

← **regno** The register of interest

← **value** The value to be set

Here is the call graph for this function:



5.13.3 Variable Documentation

5.13.3.1 struct `cpu_state` `cpu_state`

Current cpu state. Globally available.

5.13.3.2 int `do_stats`

Whether we are doing statistical analysis. Globally available

5.13.3.3 struct `hist_exec*` `hist_exec_tail`

History of execution. Globally available

5.13.3.4 oraddr_t `pcnext`

Temporary program counter. Globally available

5.13.3.5 int `sbuf_total_cyc`

Number of total store cycles. Globally available

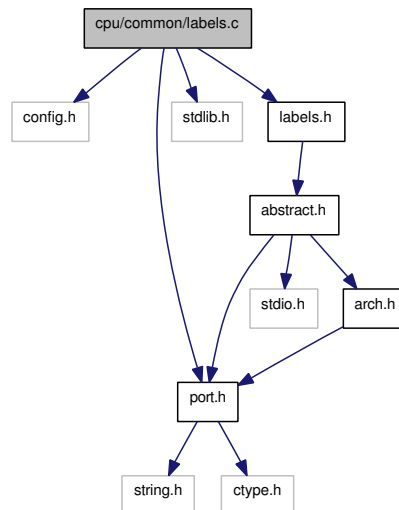
5.13.3.6 int `sbuf_wait_cyc`

Num cycles waiting for stores to complete. Globally available

5.14 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`]

5.14.1 Define Documentation

5.14.1.1 `#define LABELS_HASH_SIZE 119`

5.14.2 Function Documentation

5.14.2.1 `void add_breakpoint (oraddr_t addr)`

5.14.2.2 `void add_label (oraddr_t addr, char * name)`

5.14.2.3 `oraddr_t eval_label (char * name)`

Here is the call graph for this function:



5.14.2.4 `struct label_entry* find_label (char * name)` [read]

5.14.2.5 `struct label_entry* get_label (oraddr_t addr)` [read]

5.14.2.6 `int has_breakpoint (oraddr_t addr)`

5.14.2.7 `void init_breakpoints ()`

5.14.2.8 `void init_labels ()`

5.14.2.9 `void print_breakpoints ()`

5.14.2.10 `void remove_breakpoint (oraddr_t addr)`

5.14.3 Variable Documentation

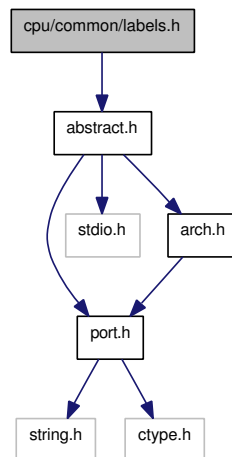
5.14.3.1 `struct breakpoint_entry* breakpoints`

5.14.3.2 `struct label_entry* label_hash[LABELS_HASH_SIZE]` [static]

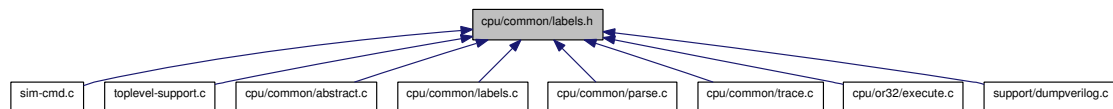
5.15 cpu/common/labels.h File Reference

```
#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](#)

5.15.1 Function Documentation

5.15.1.1 void `add_breakpoint` (`oraddr_t addr`)

5.15.1.2 void `add_label` (`oraddr_t addr`, `char * name`)

5.15.1.3 `oraddr_t eval_label` (`char * name`)

Here is the call graph for this function:



5.15.1.4 `struct label_entry* find_label` (`char * name`) [read]

5.15.1.5 `struct label_entry* get_label` (`oraddr_t addr`) [read]

5.15.1.6 int `has_breakpoint` (`oraddr_t addr`)

5.15.1.7 void `init_breakpoints` ()

5.15.1.8 void `init_labels` ()

5.15.1.9 void `print_breakpoints` ()

5.15.1.10 void `remove_breakpoint` (`oraddr_t addr`)

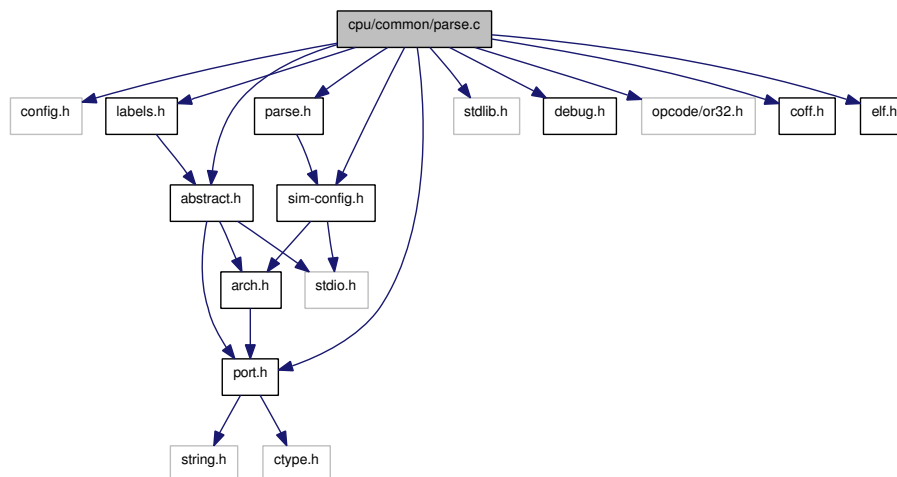
5.15.2 Variable Documentation

5.15.2.1 `struct breakpoint_entry* breakpoints`

5.16 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](#)

5.16.1 Define Documentation

5.16.1.1 #define IMM_STATS 0

Whether to do immediate statistics. This seems to be for local debugging of [parse.c](#)

5.16.1.2 #define MEMORY_LEN 0x100000000ULL

5.16.2 Function Documentation

5.16.2.1 static void addprogram ([oraddr_t](#) *address*, [uint32_t](#) *insn*, [int](#) * *breakpoint*) [static]

Add an instruction to the program

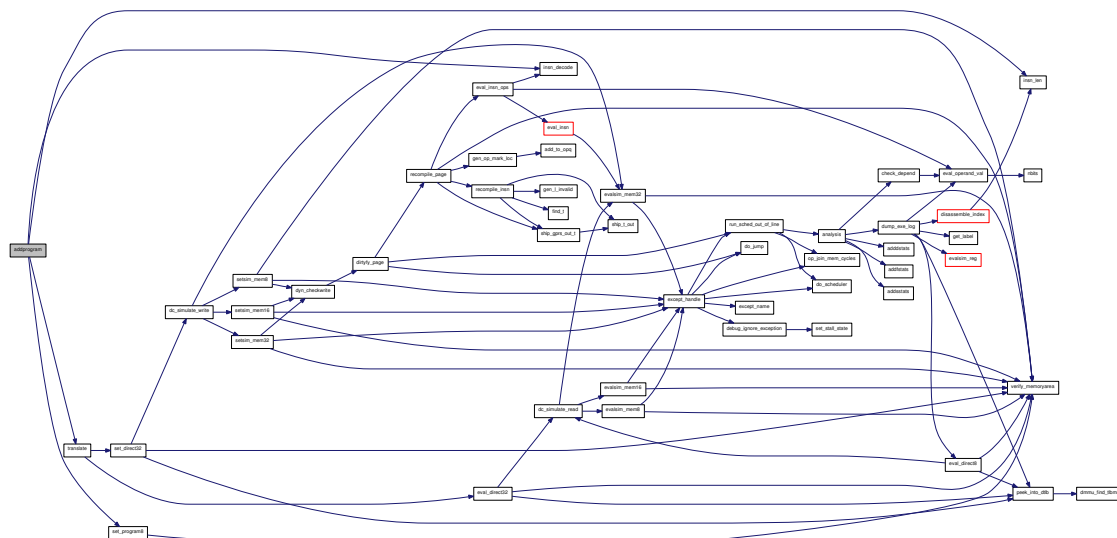
Note:

insn must be in big endian format

Parameters:

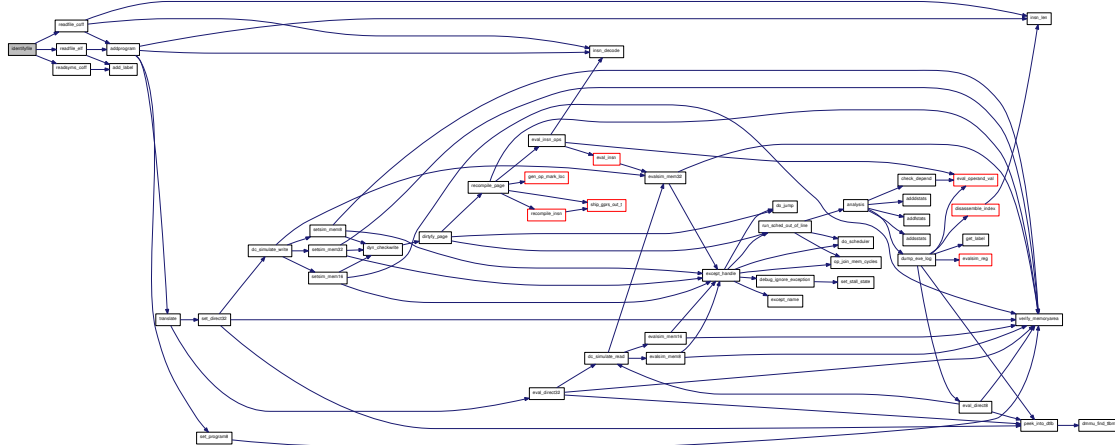
- ← *address* The address to use
- ← *insn* The instruction to add
- ← *breakpoint* Not used (it is passed to the [translate\(\)](#) function, which also does not use it.

Here is the call graph for this function:



5.16.2.2 static void identifyfile (char *filename) [static]

Here is the call graph for this function:



5.16.2.3 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

in 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)

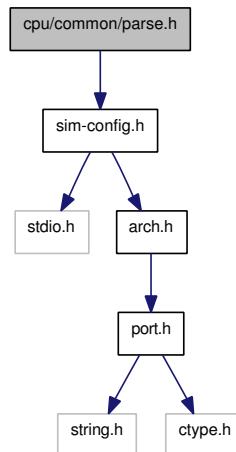
Returns:

zero on success, negative on failure.

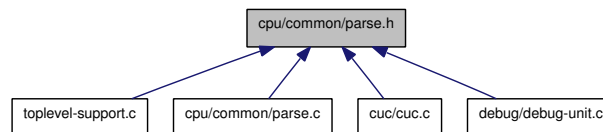
5.17 cpu/common/parse.h File Reference

```
#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`)

5.17.1 Function Documentation

5.17.1.1 `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

in `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)

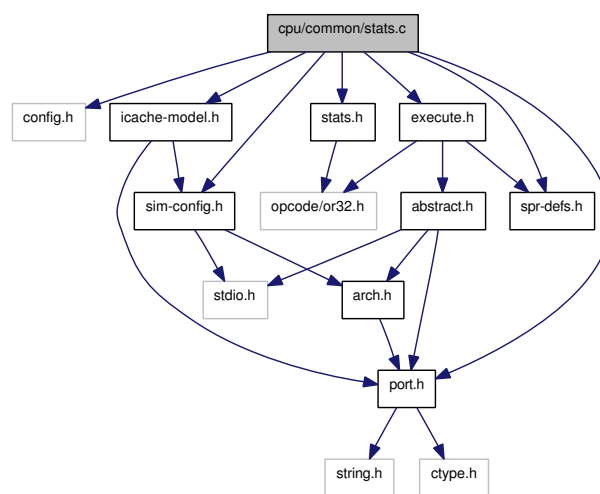
Returns:

zero on success, negative on failure.

5.18 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](#)
- struct [raw_stats](#)

Defines

- #define [DSTATS_LEN](#) 3000
- #define [SSTATS_LEN](#) 300
- #define [FSTATS_LEN](#) 200
- #define [RAW_RANGE](#) 1000
- #define [SD\(X\)](#) (X != 0 ? X : 1)

Functions

- void [addsstats](#) (int item, int cnt_dynamic)

- void [addstats](#) (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 }
- static struct [dstats_entry](#) [dstats](#) [DSTATS_LEN]
- static struct [sstats_entry](#) [sstats](#) [SSTATS_LEN]
- static struct [fstats_entry](#) [fstats](#) [FSTATS_LEN]
- static struct [raw_stats](#) [raw_stats](#)

5.18.1 Define Documentation

5.18.1.1 `#define DSTATS_LEN 3000`

5.18.1.2 `#define FSTATS_LEN 200`

5.18.1.3 `#define RAW_RANGE 1000`

5.18.1.4 `#define SD(X) (X != 0 ? X : 1)`

5.18.1.5 `#define SSTATS_LEN 300`

5.18.2 Function Documentation

5.18.2.1 `void adddstats (int item1, int item2, int cnt_dynamic, int depend)`

5.18.2.2 `void addfstats (enum insn_type item1, enum insn_type item2, int cnt_dynamic, int depend)`

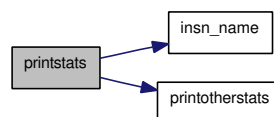
5.18.2.3 `void addsstats (int item, int cnt_dynamic)`

5.18.2.4 `void initstats ()`

5.18.2.5 `static void printotherstats (int which)` [static]

5.18.2.6 `void printstats (int which)`

Here is the call graph for this function:



5.18.3 Variable Documentation

5.18.3.1 `struct cachestats_entry dc_stats = { 0 }`

data cache stats

5.18.3.2 `struct dmmustats_entry dmmu_stats = { 0 }`

data MMU stats

5.18.3.3 `struct dstats_entry dstats[DSTATS_LEN]` [static]

dependency stats

5.18.3.4 struct fstats_entry fstats[FSTATS_LEN] [static]

func units stats

5.18.3.5 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](#)

5.18.3.6 struct cachestats_entry ic_stats = { 0 }

instruction cache stats

5.18.3.7 struct immustats_entry immu_stats = { 0 }

insn MMU stats

5.18.3.8 struct mstats_entry or1k_mstats = { 0 }

misc units stats

5.18.3.9 struct raw_stats raw_stats [static]

RAW hazard stats

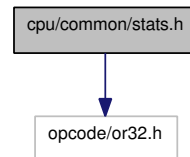
5.18.3.10 struct sstats_entry sstats[SSTATS_LEN] [static]

single stats

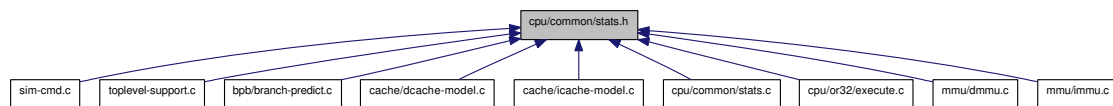
5.19 cpu/common/stats.h File Reference

```
#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](#)

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](#)

5.19.1 Function Documentation

5.19.1.1 void adddstats (int *item1*, int *item2*, int *cnt_dynamic*, int *depend*)

5.19.1.2 void addfstats (enum insn_type *item1*, enum insn_type *item2*, int *cnt_dynamic*, int *depend*)

5.19.1.3 void addsstats (int *item*, int *cnt_dynamic*)

5.19.1.4 void initstats ()

5.19.1.5 void printstats ()

5.19.2 Variable Documentation

5.19.2.1 struct cachestats_entry dc_stats

data cache stats

5.19.2.2 struct dmmustats_entry dmmu_stats

data MMU stats

5.19.2.3 struct cachestats_entry ic_stats

instruction cache stats

5.19.2.4 struct immustats_entry immu_stats

insn MMU stats

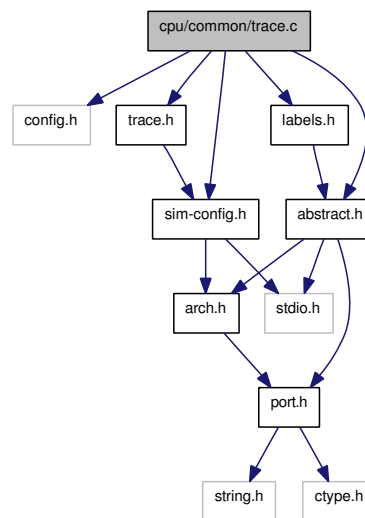
5.19.2.5 struct mstats_entry or1k_mstats

misc units stats

5.20 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*)

5.20.1 Function Documentation

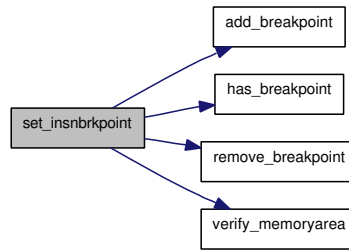
5.20.1.1 void `set_insnbrkpoint` (`oraddr_t` *addr*)

Set instruction execution breakpoint

Parameters:

← *addr* The address for the breakpoint

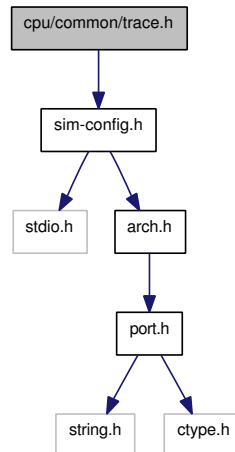
Here is the call graph for this function:



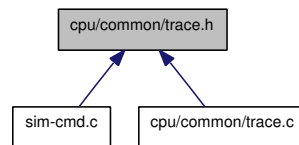
5.21 cpu/common/trace.h File Reference

```
#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*)

5.21.1 Function Documentation

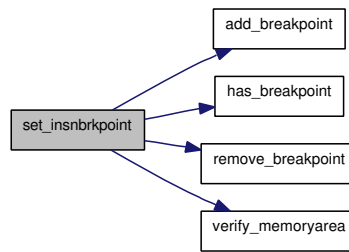
5.21.1.1 void set_insnbrkpoint (oraddr_t *addr*)

Set instruction execution breakpoint

Parameters:

← *addr* The address for the breakpoint

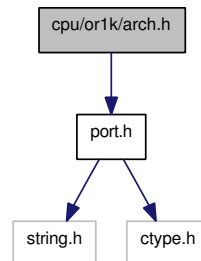
Here is the call graph for this function:



5.22 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](#)

5.22.1 Define Documentation

5.22.1.1 #define ADDR_C(c) UINT32_C(c)

5.22.1.2 #define LINK_REGNO 9

Index of the link register

5.22.1.3 #define PRIdREG PRId32

print an openrisc register in decimals

5.22.1.4 #define PRIxADDR "08" PRIx32

print an openrisc address in hex

5.22.1.5 #define PRIxREG "08" PRIx32

print an openrisc register in hex

5.22.1.6 #define REG_C(c) UINT32_C(c)**5.22.2 Typedef Documentation****5.22.2.1 typedef uint32_t oraddr_t**

Address as addressed by openrisc

5.22.2.2 typedef int32_t orreg_t

A signed register of openrisc

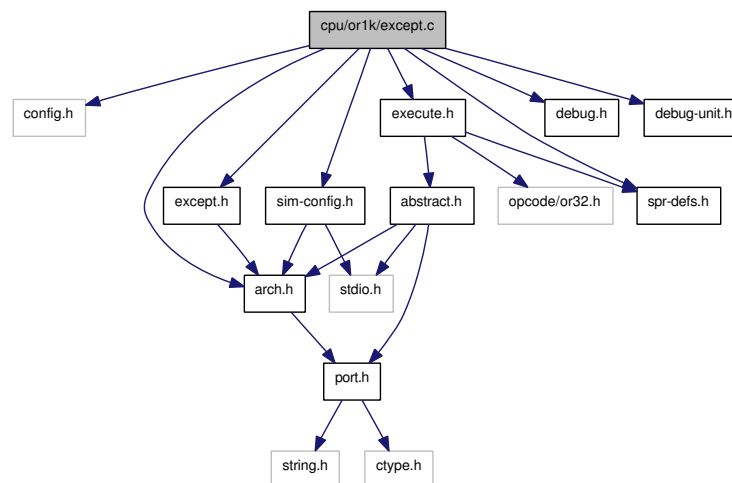
5.22.2.3 typedef uint32_t uorreg_t

An unsigned register of openrisc

5.23 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

- [DEFAULT_DEBUG_CHANNEL](#) (except)
- static const char * [except_name](#) (oraddr_t except)
- void [except_handle](#) (oraddr_t except, oraddr_t ea)

Variables

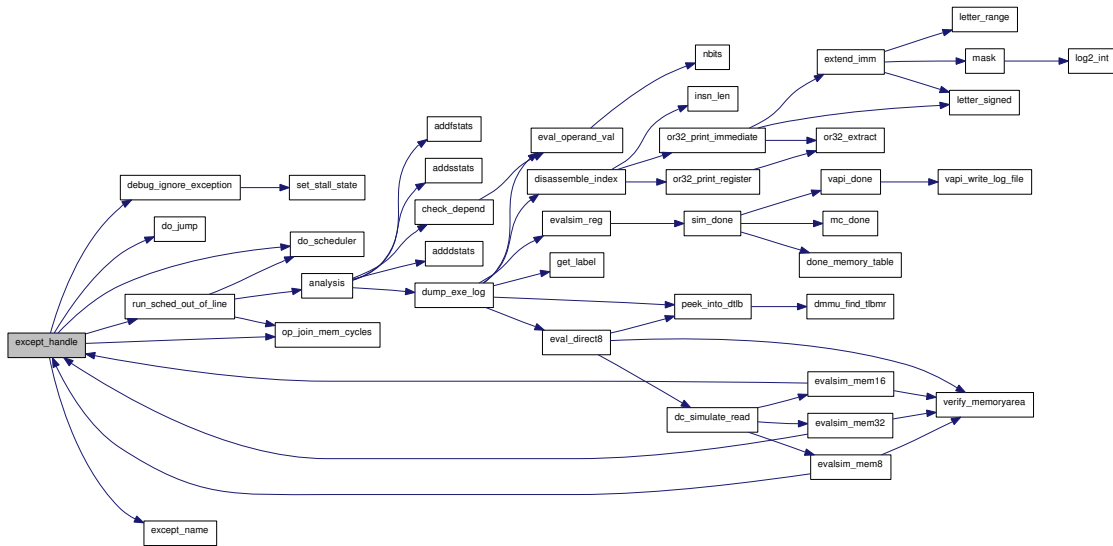
- int [except_pending](#) = 0
- static const char * [except_names](#) []

5.23.1 Function Documentation

5.23.1.1 DEFAULT_DEBUG_CHANNEL (except)

5.23.1.2 void except_handle (oraddr_t except, oraddr_t ea)

Here is the call graph for this function:



5.23.1.3 static const char* except_name (oraddr_t except) [static]

5.23.2 Variable Documentation

5.23.2.1 const char* except_names[] [static]

Initial value:

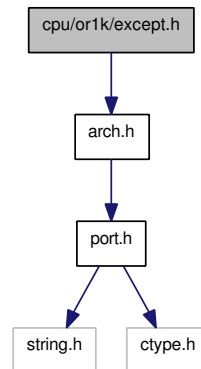
```
{
  NULL,
  "Reset",
  "Bus Error",
  "Data Page Fault",
  "Insn Page Fault",
  "Tick timer",
  "Alignment",
  "Illegal instruction",
  "Interrupt",
  "Data TLB Miss",
  "Insn TLB Miss",
  "Range",
  "System Call",
  "Floating Point",
  "Trap"
}
```

5.23.2.2 int except_pending = 0

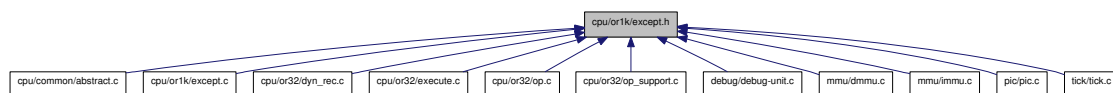
5.24 cpu/or1k/except.h File Reference

```
#include "arch.h"
```

Include dependency graph for except.h:



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



Defines

- #define `EXCEPT_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_TRAP` 0x0e00

Functions

- void `except_handle` (`oraddr_t` except, `oraddr_t` ea)

Variables

- int `except_pending`

5.24.1 Define Documentation

5.24.1.1 `#define EXCEPT_ALIGN 0x0600`

5.24.1.2 `#define EXCEPT_BUSERR 0x0200`

5.24.1.3 `#define EXCEPT_DPF 0x0300`

5.24.1.4 `#define EXCEPT_DTLBMISS 0x0900`

5.24.1.5 `#define EXCEPT_ILLEGAL 0x0700`

5.24.1.6 `#define EXCEPT_INT 0x0800`

5.24.1.7 `#define EXCEPT_IPF 0x0400`

5.24.1.8 `#define EXCEPT_ITLBMISS 0x0a00`

5.24.1.9 `#define EXCEPT_RANGE 0x0b00`

5.24.1.10 `#define EXCEPT_RESET 0x0100`

5.24.1.11 `#define EXCEPT_SYSCALL 0x0c00`

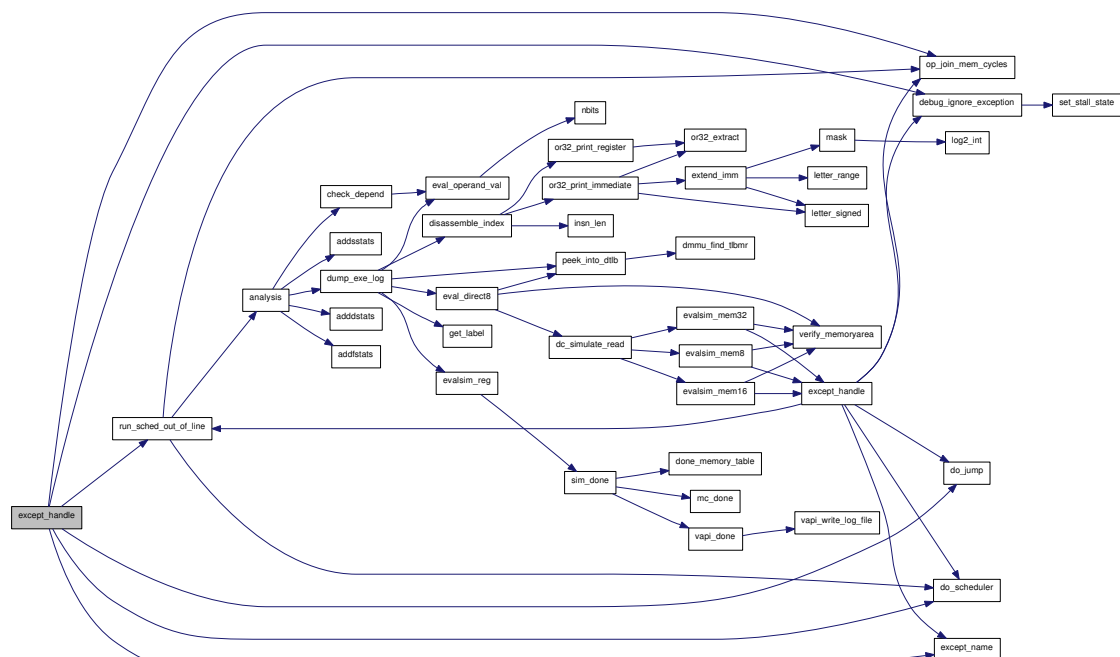
5.24.1.12 `#define EXCEPT_TICK 0x0500`

5.24.1.13 `#define EXCEPT_TRAP 0x0e00`

5.24.2 Function Documentation

5.24.2.1 `void except_handle (oraddr_t except, oraddr_t ea)`

Here is the call graph for this function:



5.24.3 Variable Documentation

5.24.3.1 `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

5.25.1 Define Documentation

5.25.1.1 #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)
```

5.25.1.2 **#define MAX_GRPS (32)**

5.25.1.3 **#define MAX_MATCHPOINTS 8**

5.25.1.4 **#define MAX_SPRS (0x10000)**

5.25.1.5 **#define MAX_SPRS_PER_GRP (1 << MAX_SPRS_PER_GRP_BITS)**

5.25.1.6 **#define MAX_SPRS_PER_GRP_BITS (11)**

5.25.1.7 **#define MAX_WATCHPOINTS (MAX_MATCHPOINTS + 2)**

5.25.1.8 **#define NOP_CNT_RESET 0x0005**

5.25.1.9 **#define NOP_EXIT 0x0001**

5.25.1.10 **#define NOP_NOP 0x0000**

5.25.1.11 **#define NOP_PRINTF 0x0003**

5.25.1.12 **#define NOP_PUTC 0x0004**

5.25.1.13 **#define NOP_REPORT 0x0002**

5.25.1.14 **#define NOP_REPORT_FIRST 0x0400**

5.25.1.15 **#define NOP_REPORT_LAST 0x03ff**

5.25.1.16 **#define SPR_CPUCFGR (SPRGROUP_SYS + 2)**

5.25.1.17 **#define SPR_CPUCFGR_CGF 0x00000010**

5.25.1.18 **#define SPR_CPUCFGR_NSGF 0x0000000f**

5.25.1.19 **#define SPR_CPUCFGR_OB32S 0x00000020**

5.25.1.20 **#define SPR_CPUCFGR_OB64S 0x00000040**

5.25.1.21 **#define SPR_CPUCFGR_OF32S 0x00000080**

5.25.1.22 **#define SPR_CPUCFGR_OF64S 0x00000100**

5.25.1.23 **#define SPR_CPUCFGR_OV64S 0x00000200**

5.25.1.24 **#define SPR_CPUCFGR_RES 0xffffc00**

5.25.1.25 **#define SPR_DCBFR (SPRGROUP_DC + 2)**

5.25.1.26 **#define SPR_DCBIR (SPRGROUP_DC + 3)**

5.25.1.27 **#define SPR_DCBLR (SPRGROUP_DC + 5)**

5.25.1.28 **#define SPR_DCBPR (SPRGROUP_DC + 1)**

5.25.1.29 **#define SPR_DCBWR (SPRGROUP_DC + 4)**

5.25.1.30 **#define SPR_DCCFGR (SPRGROUP_SYS + 5)**

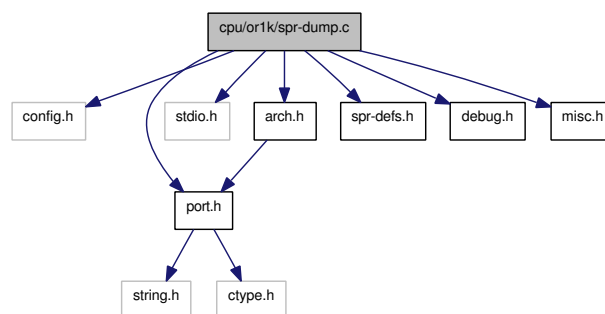
5.25.1.31 **#define SPR_DCCFGR_CBFRI 0x00002000**

5.25.1.32 **#define SPR_DCCFGR_CBIRI 0x00000400**

5.26 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

- [DEFAULT_DEBUG_CHANNEL](#) (spr)
- [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_dcfgr](#) []
- 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_ttmr []`
- static struct `spr_def spr_tt_group []`
- static struct `spr_def * spr_groups []`
- static char `ret_spr [1000]`

5.26.1 Function Documentation

5.26.1.1 DEFAULT_DEBUG_CHANNEL (spr)

5.26.1.2 char* dump_spr (uint16_t spr, uorreg_t spr_val)

Here is the call graph for this function:



5.26.2 Variable Documentation

5.26.2.1 char ret_spr[1000] [static]

5.26.2.2 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 } }
```

5.26.2.3 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 } }
```

5.26.2.4 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, 0x3fff, "SPR_DCR way 0 set %i", spr_one_val },
{ 0x400, 0x5fff, "SPR_DCR way 1 set %i", spr_one_val },
{ 0x600, 0x7fff, "SPR_DCR way 2 set %i", spr_one_val },
{ 0x800, 0x9fff, "SPR_DCR way 3 set %i", spr_one_val },
{ -1, -1, NULL, NULL } }
```

5.26.2.5 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 } }
```

5.26.2.6 struct spr_bit_def spr_dccr[] [static]**Initial value:**

```
{
{ "DCCR_EW", SPR_DCCR_EW },
{ NULL, 0 } }
```

5.26.2.7 struct spr_bit_def spr_dcfgr[] [static]**Initial value:**

```
{
{ "DCFGR_NDP", SPR_DCFGR_NDP },
{ "DCFGR_WPCI", SPR_DCFGR_WPCI },
{ NULL, 0 } }
```

5.26.2.8 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 } }
```

5.26.2.9 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 } }
```

5.26.2.10 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 } }
```

5.26.2.11 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 } }
```

5.26.2.12 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 } }
```

5.26.2.13 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 } }
```

5.26.2.14 struct spr_bit_def spr_dsr[] [static]**Initial value:**

```
{
{ "DSR_RST", SPR_DSR_RST },
{ "DSR_BUSE", SPR_DSR_BUSE },
{ "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 } }
```

5.26.2.15 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 } }
```

5.26.2.16 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 } }
```

5.26.2.17 struct spr_bit_def spr_dwcr[] [static]**Initial value:**

```
{
{ "DWCR_COUNT", SPR_DWCR_COUNT },
{ "DWCR_MATCH", SPR_DWCR_MATCH },
{ NULL, 0 } }
```

5.26.2.18 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 }
```

5.26.2.19 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 } }
```

5.26.2.20 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 } }
```

5.26.2.21 struct spr_bit_def spr_iccr[] [static]**Initial value:**

```
{
  { "ICCR_EW", SPR_ICCR_EW },
  { NULL, 0 } }
```

5.26.2.22 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 } }
```

5.26.2.23 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 } }
```

5.26.2.24 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 } }
```

5.26.2.25 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 } }
```

5.26.2.26 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 } }
```

5.26.2.27 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 } }
```

5.26.2.28 struct spr_bit_def spr_one_val[] [static]**Initial value:**

```
{
  { "", 0xffffffff },
  { NULL, 0 } }
```

5.26.2.29 struct spr_def spr_pc_group[] [static]**Initial value:**

```
{
  { 0x00, 0x07, "PCCR", spr_one_val },
  { 0x08, 0x0f, "PCMR", spr_pcmr },
  { -1, -1, NULL, NULL } }
```

5.26.2.30 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 } }

```

5.26.2.31 struct spr_def spr_pic_group[] [static]

Initial value:

```

{
{ 0, 0, "PICMR", spr_picmr },
{ 2, 2, "PICSR", spr_one_val },
{ -1, -1, NULL, NULL } }

```

5.26.2.32 struct spr_bit_def spr_picmr[] [static]

Initial value:

```

{
{ "PICMR_IUM", SPR_PICMR_IUM },
{ NULL, 0 } }

```

5.26.2.33 struct spr_def spr_pm_group[] [static]

Initial value:

```

{
{ 0x0, 0x0, "SPR_PMR", spr_pmr },
{ -1, -1, NULL, NULL } }

```

5.26.2.34 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 } }

```


5.26.2.35 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              } }

```

5.26.2.36 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      } }

```

5.26.2.37 struct spr_def spr_tt_group[] [static]**Initial value:**

```

{
{ 0, 0, "TTMR", spr_ttmr },
{ 0, 0, "TTCR", spr_one_val },
{ -1, -1, NULL, NULL } }

```

5.26.2.38 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 } }
```

5.26.2.39 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 } }
```

5.26.2.40 struct spr_bit_def spr_vr[] [static]**Initial value:**

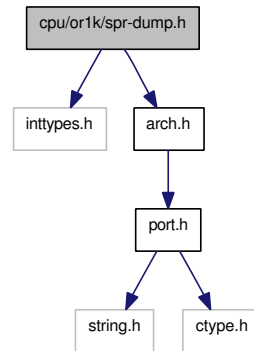
```
{
{ "VR_VER", SPR_VR_VER },
{ "VR_REV", SPR_VR_REV },
{ NULL, 0 } }
```

5.27 cpu/or1k/spr-dump.h File Reference

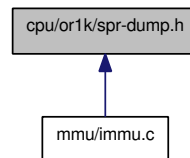
```
#include <inttypes.h>
```

```
#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)`

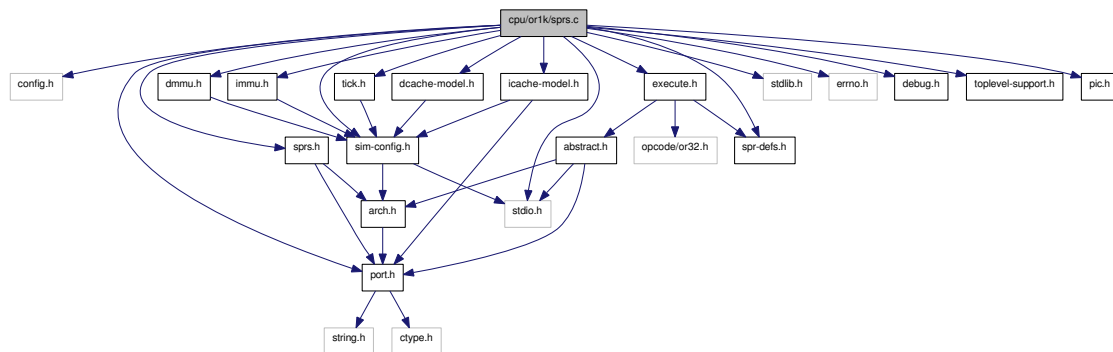
5.27.1 Function Documentation

5.27.1.1 `char* dump_spr (uint16_t spr, uorreg_t spr_val)`

5.28 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

- [DEFAULT_DEBUG_CHANNEL](#) (spr)
- [DECLARE_DEBUG_CHANNEL](#) (immu)
- void [mfspr](#) (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

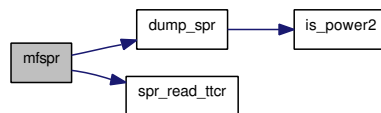
5.28.1 Function Documentation

5.28.1.1 DECLARE_DEBUG_CHANNEL (immu)

5.28.1.2 DEFAULT_DEBUG_CHANNEL (spr)

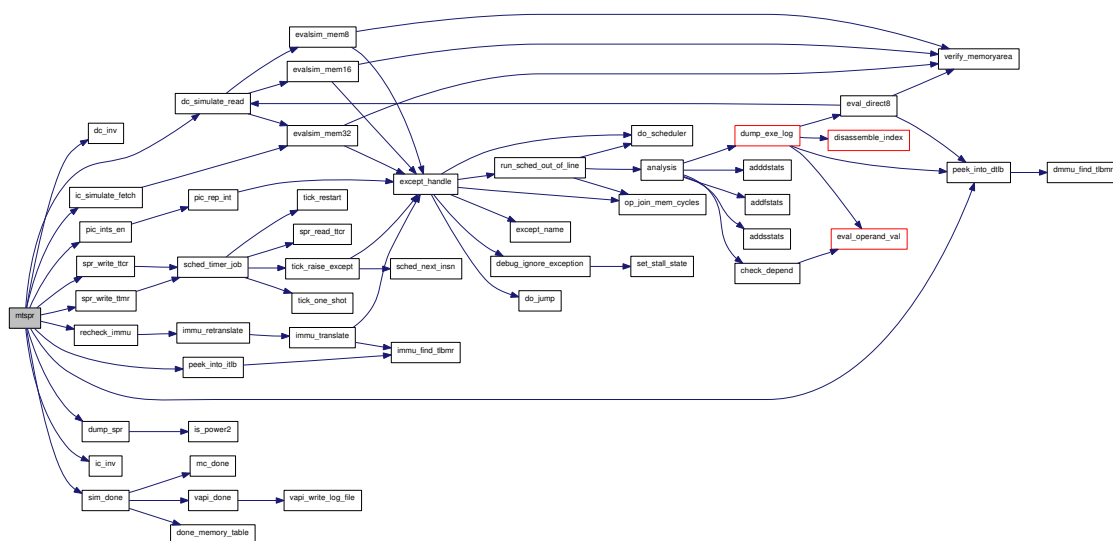
5.28.1.3 uorreg_t mfspr (const uint16_t regno)

Here is the call graph for this function:



5.28.1.4 void mtspr (uint16_t regno, const uorreg_t value)

Here is the call graph for this function:



5.28.1.5 void sprs_status (void)

5.28.2 Variable Documentation

5.28.2.1 int audio_cnt = 0 [static]

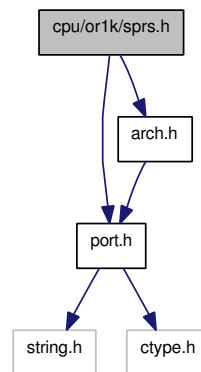
5.28.2.2 FILE* fo = 0 [static]

5.29 cpu/or1k/sprs.h File Reference

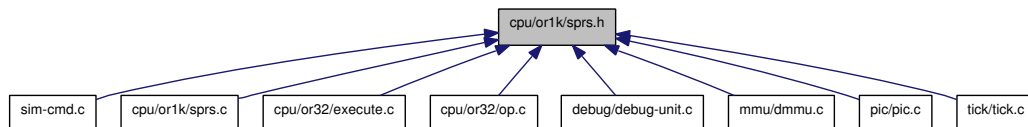
```
#include "port.h"
```

```
#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 `sprrs_status` ()
- char * `dump_spr` (uint16_t spr, `uorreg_t` spr_val)

5.29.1 Function Documentation

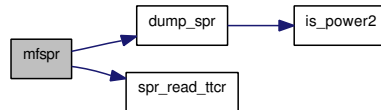
5.29.1.1 char* `dump_spr` (uint16_t *spr*, `uorreg_t` *spr_val*)

Here is the call graph for this function:



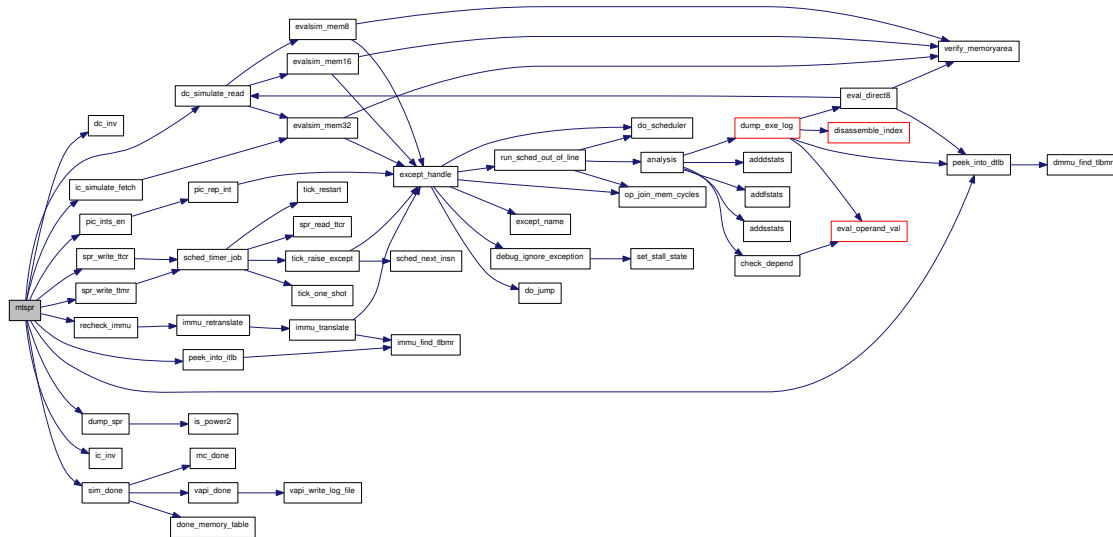
5.29.1.2 uorreg_t mfspr (const uint16_t regno)

Here is the call graph for this function:



5.29.1.3 void mtspr (uint16_t regno, const uorreg_t value)

Here is the call graph for this function:



5.29.1.4 void sprs_status ()

5.30 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`

5.30.1 Function Documentation

5.30.1.1 `static oraddr_t get_pc (void)` [`static`]

5.30.1.2 `static void set_pc (oraddr_t pc)` [`static`]

5.30.1.3 `static void upd_sim_cycles (void)` [`static`]

5.30.2 Variable Documentation

5.30.2.1 `uint32_t high32`

5.30.2.2 `uint32_t low32`

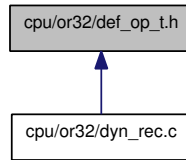
5.30.2.3 `union { ... } useless_x86` [`static`]

5.30.2.4 `struct { ... } val3232`

5.30.2.5 `uint64_t val64`

5.31 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)

5.31.1 Define Documentation

5.31.1.1 #define DEF_1T_OP(type, name, op_name)

Value:

```
static const type name[NUM_T_REGS] = \
    OP_ROW(op_name)
```

5.31.1.2 #define DEF_2T_OP(type, name, op_name)

Value:

```
static const type name[NUM_T_REGS][NUM_T_REGS] = \
    OP_ROW_COL(op_name)
```

5.31.1.3 #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)
```

5.31.1.4 #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)
```

5.31.1.5 #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)
```

5.31.1.6 #define DEF_GPR_OP(type, name, op_name)

Value:

```
static const generic_gen_op name[NUM_T_REGS][32] = \
    GPR_ROW_COL(op_name)
```

5.31.1.7 #define GPR_T(op_name, temp)

5.32 cpu/or32/dyn32_defs.h File Reference

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)

5.32.1 Function Documentation

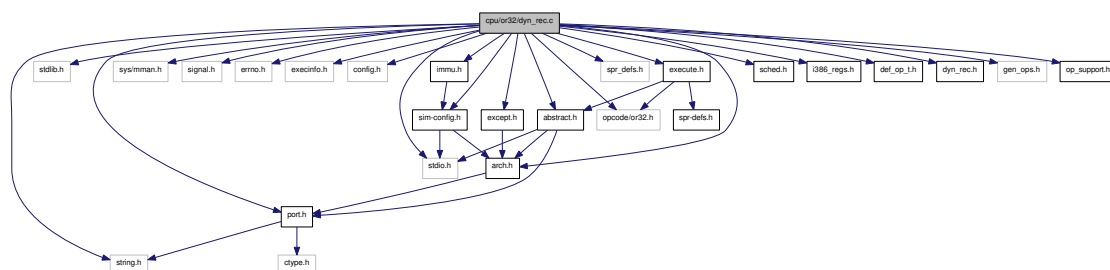
5.32.1.1 void [l_none](#) (struct [op_queue](#) * *opq*, int * *param_t*, int *delay_slot*)

5.32.1.2 void [gen_lf_sub_s](#) PARAMS ((struct [op_queue](#) *, int *, int))

5.33 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 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_IT_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_IT_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_excpt, gen_op_check_null_except)
- DEF_1T_OP (generic_gen_op, check_null_excpt_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_sftu_null_t_table`, `gen_op_sftu_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_sftu_imm_t_table`, `gen_op_sftu_imm`)
- `DEF_2T_OP` (`generic_gen_op`, `l_sftu_t_table`, `gen_op_sftu`)
- void `gen_l_sftu` (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]

5.33.1 Define Documentation

- 5.33.1.1 `#define OPS_ENLARGE_BY 5`
- 5.33.1.2 `#define RECED_PAGE_ENLARGE_BY 51200`
- 5.33.1.3 `#define T_NONE (-1)`
- 5.33.1.4 `#define TFLAG_DST 2`
- 5.33.1.5 `#define TFLAG_SAVED 4`
- 5.33.1.6 `#define TFLAG_SOURCED 8`
- 5.33.1.7 `#define TFLAG_SRC 1`

5.33.2 Typedef Documentation

- 5.33.2.1 `typedef void(* generic_gen_op)(struct op_queue *opq, int end)`
- 5.33.2.2 `typedef void(* imm_gen_op)(struct op_queue *opq, int end, uorreg_t imm)`

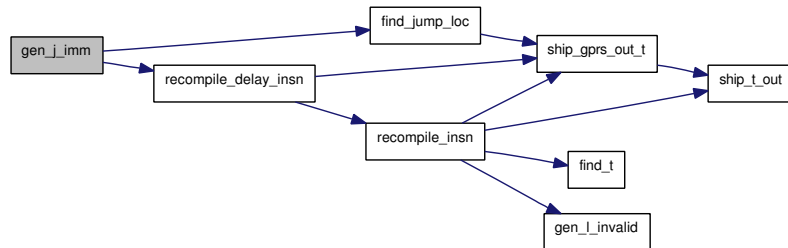
5.33.3 Function Documentation

- 5.33.3.1 `void add_to_op_params (struct op_queue * opq, int end, unsigned long param)`
- 5.33.3.2 `void add_to_opq (struct op_queue * opq, int end, int op)`
- 5.33.3.3 `DEF_1T_OP (imm_gen_op, l_sfne_imm_t_table, gen_op_sfne_imm)`
- 5.33.3.4 `DEF_1T_OP (generic_gen_op, l_sfne_null_t_table, gen_op_sfne_null)`
- 5.33.3.5 `DEF_1T_OP (imm_gen_op, l_sftu_imm_t_table, gen_op_sftu_imm)`
- 5.33.3.6 `DEF_1T_OP (imm_gen_op, l_sfts_imm_t_table, gen_op_sfts_imm)`
- 5.33.3.7 `DEF_1T_OP (imm_gen_op, l_sfleu_imm_t_table, gen_op_sfleu_imm)`
- 5.33.3.8 `DEF_1T_OP (imm_gen_op, l_sfles_imm_t_table, gen_op_sfles_imm)`
- 5.33.3.9 `DEF_1T_OP (imm_gen_op, l_sfgtu_imm_t_table, gen_op_sfgtu_imm)`
- 5.33.3.10 `DEF_1T_OP (generic_gen_op, l_sftu_null_t_table, gen_op_sftu_null)`
- 5.33.3.11 `DEF_1T_OP (generic_gen_op, l_sfgtu_null_t_table, gen_op_sfgtu_null)`
- 5.33.3.12 `DEF_1T_OP (imm_gen_op, l_sfgts_imm_t_table, gen_op_sfgts_imm)`
- 5.33.3.13 `DEF_1T_OP (generic_gen_op, l_sfts_null_t_table, gen_op_sfts_null)`
- 5.33.3.14 `DEF_1T_OP (generic_gen_op, l_sfgts_null_t_table, gen_op_sfgts_null)`
- 5.33.3.15 `DEF_1T_OP (imm_gen_op, l_sfgeu_imm_t_table, gen_op_sfgeu_imm)`
- 5.33.3.16 `DEF_1T_OP (generic_gen_op, l_sfneu_imm_t_table, gen_op_sfneu_imm)`
- 5.33.3.17 `DEF_1T_OP (generic_gen_op, l_sfgeu_null_t_table, gen_op_sfgeu_null)`
- 5.33.3.18 `DEF_1T_OP (imm_gen_op, l_sfges_imm_t_table, gen_op_sfges_imm)`
- 5.33.3.19 `DEF_1T_OP (generic_gen_op, l_sfges_null_t_table, gen_op_sfges_null)`

5.33.3.107 `static int find_t (struct op_queue * opq, unsigned int reg)` [static]

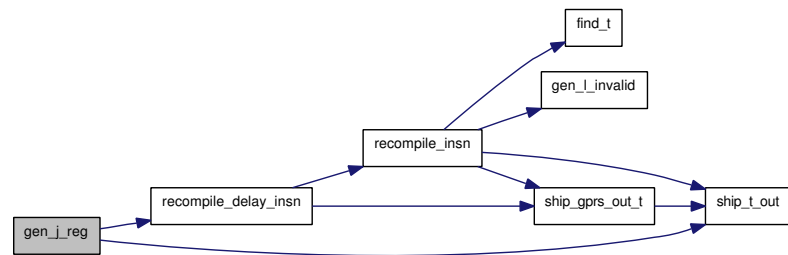
5.33.3.108 `static void gen_j_imm (struct op_queue * opq, oraddr_t off)` [static]

Here is the call graph for this function:



5.33.3.109 `static void gen_j_reg (struct op_queue * opq, unsigned int gpr)` [static]

Here is the call graph for this function:



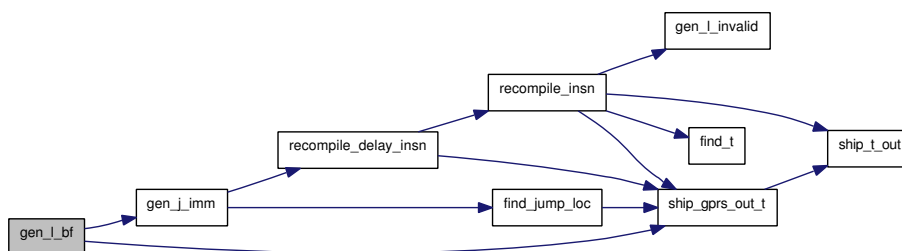
5.33.3.110 `void gen_l_add (struct op_queue * opq, int param_t[3], int delay_slot)`

5.33.3.111 `void gen_l_addc (struct op_queue * opq, int param_t[3], int delay_slot)`

5.33.3.112 `void gen_l_and (struct op_queue * opq, int param_t[3], int delay_slot)`

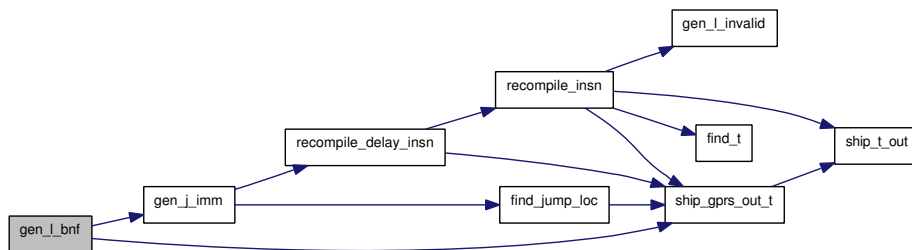
5.33.3.113 `void gen_l_bf (struct op_queue * opq, int param_t[3], int delay_slot)`

Here is the call graph for this function:



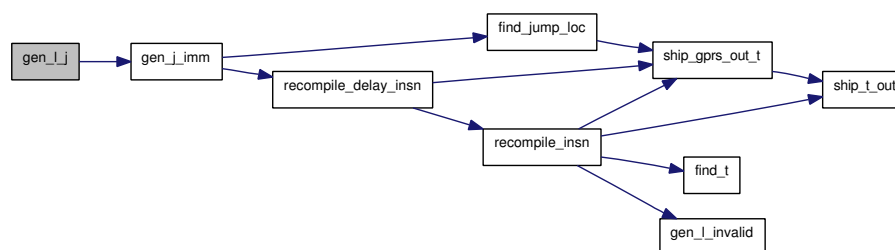
5.33.3.114 void gen_l_bnf (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:



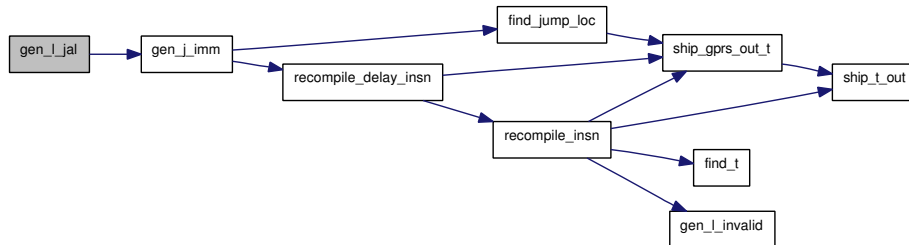
- 5.33.3.115 void `gen_l_cmov` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.116 void `gen_l_cust1` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.117 void `gen_l_cust2` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.118 void `gen_l_cust3` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.119 void `gen_l_cust4` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.120 void `gen_l_cust5` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.121 void `gen_l_cust6` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.122 void `gen_l_cust7` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.123 void `gen_l_cust8` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.124 void `gen_l_div` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.125 void `gen_l_divu` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.126 void `gen_l_extbs` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.127 void `gen_l_extbz` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.128 void `gen_l_exths` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.129 void `gen_l_exthz` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.130 void `gen_l_extws` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.131 void `gen_l_extwz` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.132 void `gen_l_ff1` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.133 void `gen_l_invalid` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)
- 5.33.3.134 void `gen_l_j` (struct `op_queue` * *opq*, int *param_t*[3], int *delay_slot*)

Here is the call graph for this function:



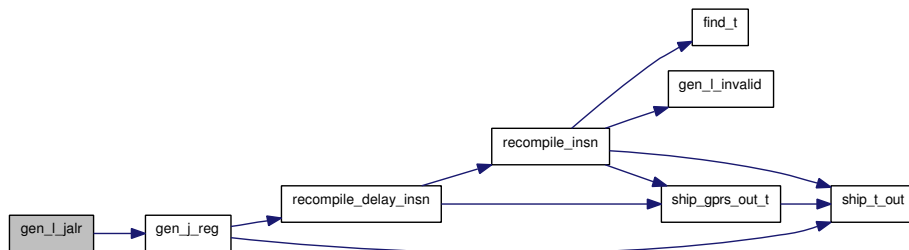
5.33.3.135 void gen_l_jal (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:



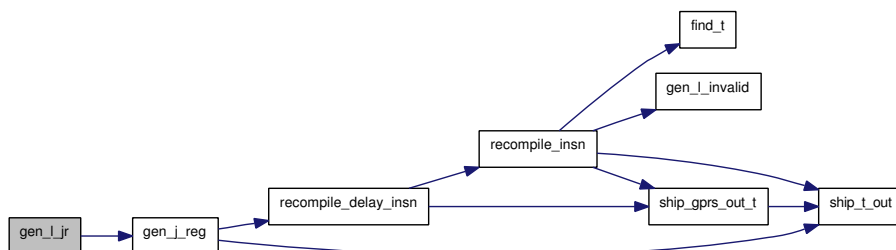
5.33.3.136 void gen_l_jalr (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:



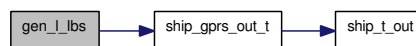
5.33.3.137 void gen_l_jr (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:



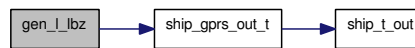
5.33.3.138 void gen_l_lbs (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:

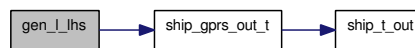


5.33.3.139 void gen_l_lbz (struct op_queue * opq, int param_t[3], int delay_slot)

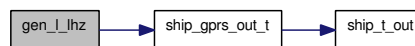
Here is the call graph for this function:

**5.33.3.140 void gen_l_lhs (struct op_queue * opq, int param_t[3], int delay_slot)**

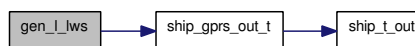
Here is the call graph for this function:

**5.33.3.141 void gen_l_lhz (struct op_queue * opq, int param_t[3], int delay_slot)**

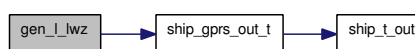
Here is the call graph for this function:

**5.33.3.142 void gen_l_lws (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**5.33.3.143 void gen_l_lwz (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:



5.33.3.144 void `gen_l_mac` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

5.33.3.145 void `gen_l_macrc` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

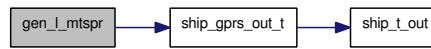
5.33.3.146 void `gen_l_mfspr` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

5.33.3.147 void `gen_l_movhi` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

5.33.3.148 void `gen_l_msb` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

5.33.3.149 void `gen_l_mtspr` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:

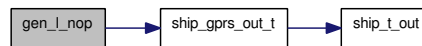


5.33.3.150 void `gen_l_mul` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

5.33.3.151 void `gen_l_mulu` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

5.33.3.152 void `gen_l_nop` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:

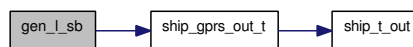


5.33.3.153 void `gen_l_or` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

5.33.3.154 void `gen_l_rfe` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

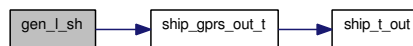
5.33.3.155 void `gen_l_sb` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



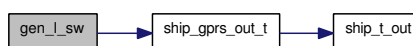
- 5.33.3.156 void `gen_l_sfeq` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.157 void `gen_l_sfges` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.158 void `gen_l_sfgeu` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.159 void `gen_l_sfgts` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.160 void `gen_l_sfgtu` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.161 void `gen_l_sfles` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.162 void `gen_l_sfleu` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.163 void `gen_l_sfits` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.164 void `gen_l_sftu` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.165 void `gen_l_sfne` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.166 void `gen_l_sh` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



- 5.33.3.167 void `gen_l_sll` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.168 void `gen_l_sra` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.169 void `gen_l_srl` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.170 void `gen_l_sub` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)
- 5.33.3.171 void `gen_l_sw` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



5.33.3.172 void `gen_l_sys` (struct `op_queue` * `opq`, int `param_t`[3], int `delay_slot`)

5.33.3.173 void `gen_l_trap` (struct `op_queue` * `opq`, int `param_t`[3], int `delay_slot`)

5.33.3.174 void `gen_l_xor` (struct `op_queue` * `opq`, int `param_t`[3], int `delay_slot`)

5.33.3.175 void `gen_lf_add_s` (struct `op_queue` * `opq`, int `param_t`[3], int `delay_slot`)

Here is the call graph for this function:



5.33.3.176 void `gen_lf_div_s` (struct `op_queue` * `opq`, int `param_t`[3], int `delay_slot`)

Here is the call graph for this function:



5.33.3.177 void `gen_lf_ftoi_s` (struct `op_queue` * `opq`, int `param_t`[3], int `delay_slot`)

Here is the call graph for this function:



5.33.3.178 void `gen_lf_itof_s` (struct `op_queue` * `opq`, int `param_t`[3], int `delay_slot`)

Here is the call graph for this function:



5.33.3.179 void `gen_lf_madd_s` (struct `op_queue` * `opq`, int `param_t`[3], int `delay_slot`)

Here is the call graph for this function:



5.33.3.180 void gen_lf_mul_s (struct op_queue * opq, int param_t[3], int delay_slot)

Here is the call graph for this function:

**5.33.3.181 void gen_lf_rem_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**5.33.3.182 void gen_lf_sfeq_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**5.33.3.183 void gen_lf_sfge_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**5.33.3.184 void gen_lf_sfgt_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:

**5.33.3.185 void gen_lf_sfle_s (struct op_queue * opq, int param_t[3], int delay_slot)**

Here is the call graph for this function:



5.33.3.186 void `gen_lf_sflt_s` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



5.33.3.187 void `gen_lf_sfne_s` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



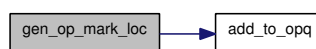
5.33.3.188 void `gen_lf_sub_s` (`struct op_queue * opq`, `int param_t[3]`, `int delay_slot`)

Here is the call graph for this function:



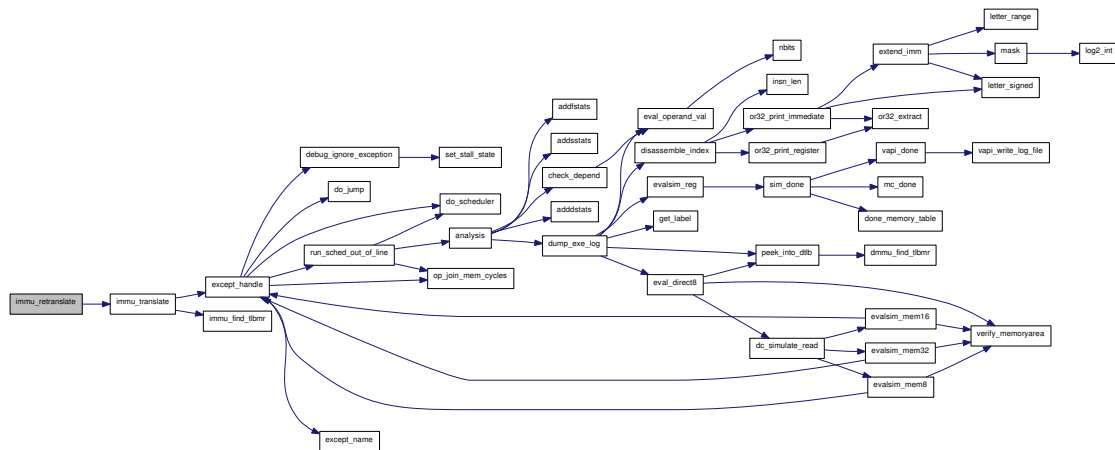
5.33.3.189 static void `gen_op_mark_loc` (`struct op_queue * opq`, `int end`) [static]

Here is the call graph for this function:



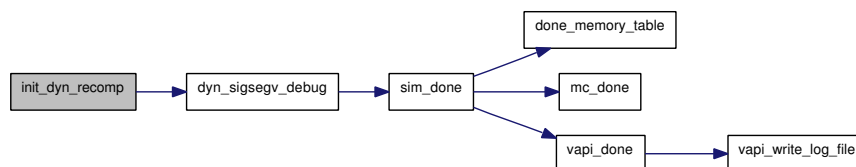
5.33.3.190 static void immu_retranslate (void * dat) [static]

Here is the call graph for this function:



5.33.3.191 void init_dyn_recomp (void)

Here is the call graph for this function:



5.33.3.199 `static void ship_t_out (struct op_queue * opq, unsigned int t)` [static]

5.33.4 Variable Documentation

5.33.4.1 `uorreg_t __op_param1`

5.33.4.2 `uorreg_t __op_param2`

5.33.4.3 `uorreg_t __op_param3`

5.33.4.4 `int do_stats`

Whether we are doing statistical analysis. Globally available

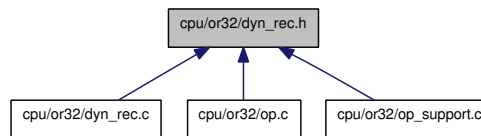
5.33.4.5 `const generic_gen_op set_pc_delay_gpr[32]` [static]

5.33.4.6 `void* sigsegv_addr = NULL` [static]

5.33.4.7 `int sigsegv_state = 0` [static]

5.34 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

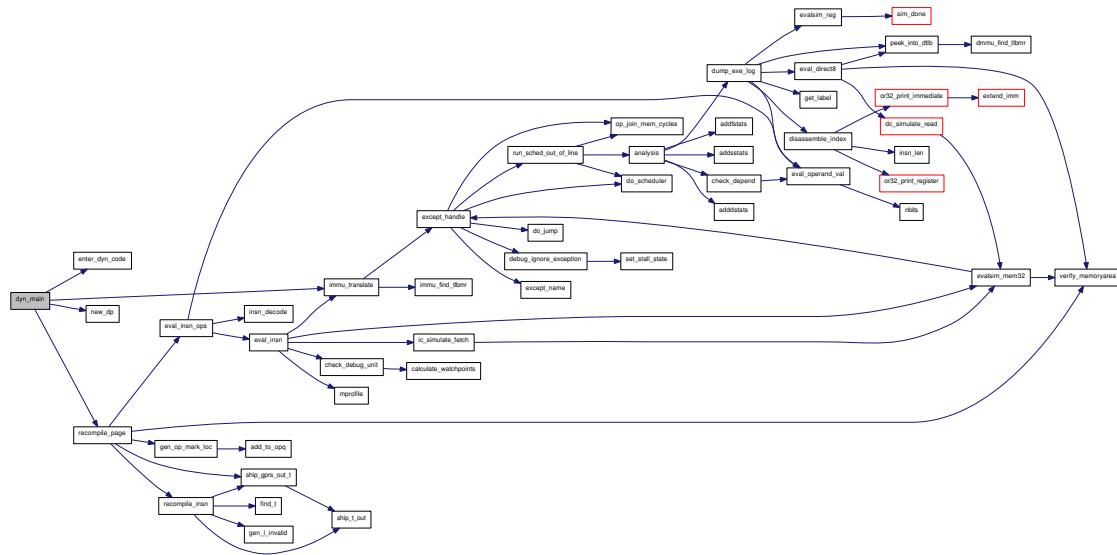
- void [recompile_page](#) (struct [dyn_page](#) *dyn)
- struct [dyn_page](#) * [new_dp](#) (oraddr_t page)
- void [add_to_opq](#) (struct [op_queue](#) *opq, int end, int op)
- void [add_to_op_params](#) (struct [op_queue](#) *opq, int end, unsigned long param)
- void * [enough_host_page](#) (struct [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](#)

5.34.3.4 void dyn_main (void)

Here is the call graph for this function:

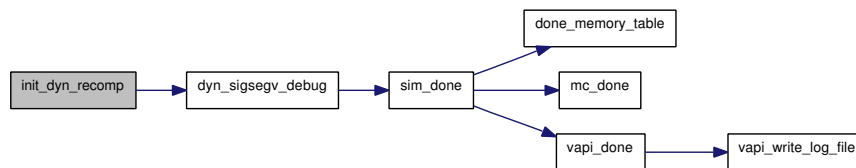


5.34.3.5 void* enough_host_page (struct dyn_page * dp, void * cur, unsigned int * len, unsigned int amount)

5.34.3.6 void enter_dyn_code (oraddr_t addr, struct dyn_page * dp)

5.34.3.7 void init_dyn_recomp (void)

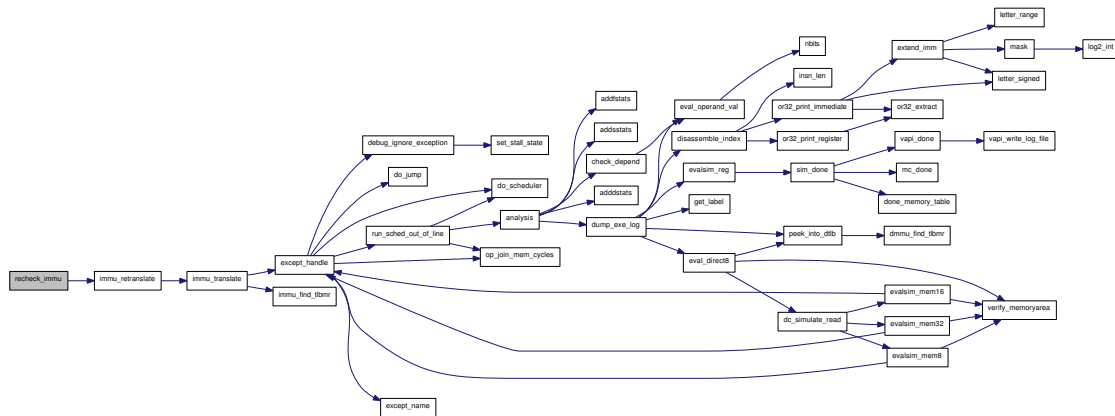
Here is the call graph for this function:



5.34.3.8 struct dyn_page* new_dp (oraddr_t page) [read]

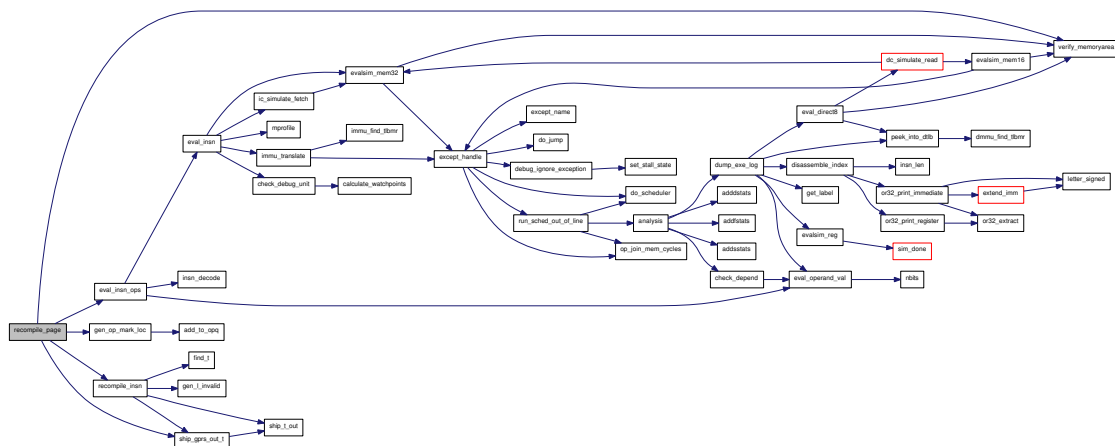
5.34.3.9 void recheck_immu (int got_en_dis)

Here is the call graph for this function:



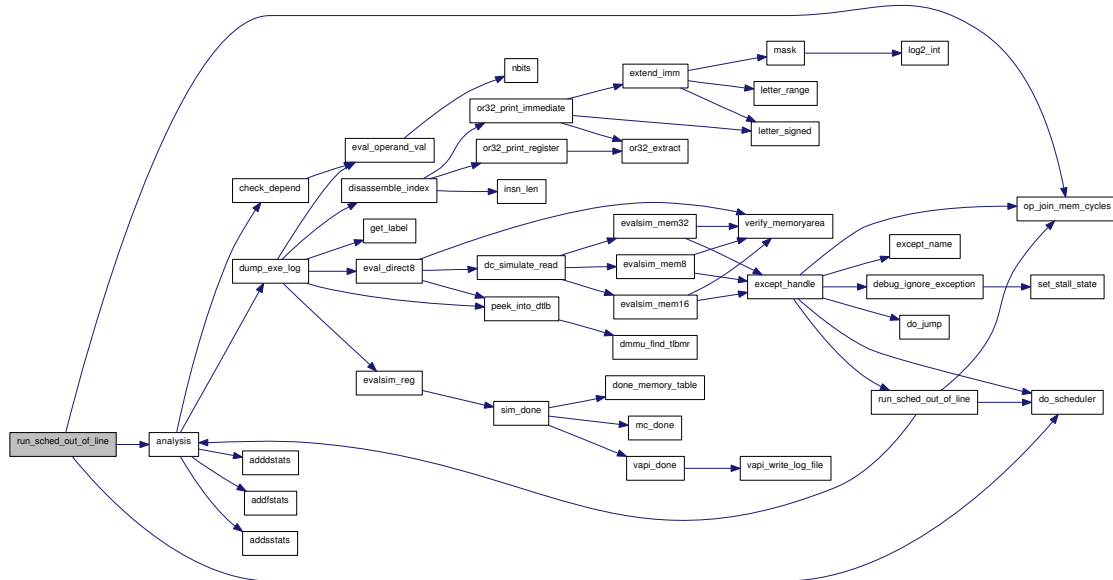
5.34.3.10 void recompile_page (struct dyn_page * dyn)

Here is the call graph for this function:



5.34.3.11 void run_sched_out_of_line (void)

Here is the call graph for this function:



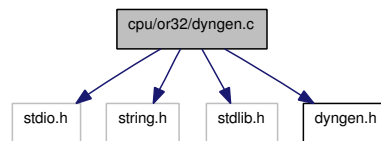
5.34.4 Variable Documentation

5.34.4.1 void* rec_stack_base

5.35 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`

5.35.1 Define Documentation

5.35.1.1 `#define MAX_PARAMS (3 + 1)`

5.35.1.2 `#define OP_FUNC_PARAM_PREFIX "__op_param"`

5.35.1.3 `#define OP_FUNC_PREFIX "op_"`

5.35.2 Function Documentation

5.35.2.1 `static void gen_func_proto (FILE *f, const char *name, int *params)` [static]

5.35.2.2 `int main (int argc, char **argv)`

Here is the call graph for this function:



5.35.3 Variable Documentation

5.35.3.1 `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"

```

5.35.3.2 `const char* c_rel_file_head` [static]

5.35.3.3 `const char* c_rel_file_tail` [static]

Initial value:

```

    }\n"
  }\n"
  opq = opq->next;\n"
}\n"

```

5.35.3.4 `const char* c_sw_file_head` [static]

5.35.3.5 `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"
```

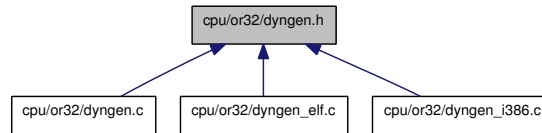
5.35.3.6 `const char* gen_code_proto` [static]

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"
```

5.36 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](#)

5.36.1 Variable Documentation

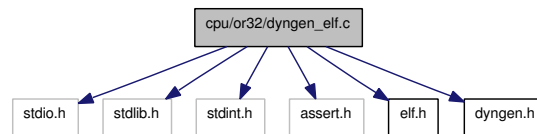
5.36.1.1 struct archf archfs

5.36.1.2 struct bff bffs

5.37 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 bffs](#)

5.37.1 Function Documentation

5.37.1.1 void [elf_close_obj](#) (void * *e_obj*)

5.37.1.2 static [Elf32_Sym](#)* [elf_find_func](#) (struct [elf_obj](#) * *obj*, unsigned int *func*) [static]

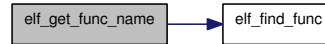
5.37.1.3 unsigned int [elf_get_func_len](#) (void * *e_obj*, unsigned int *func*)

Here is the call graph for this function:

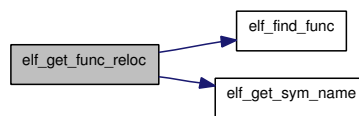


5.37.1.4 `char* elf_get_func_name (void * e_obj, unsigned int func)`

Here is the call graph for this function:

**5.37.1.5** `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:

**5.37.1.6** `void* elf_get_func_start (void * e_obj, unsigned int func)`

Here is the call graph for this function:

**5.37.1.7** `static char* elf_get_sym_name (struct elf_obj * obj, unsigned int sym)` [static]**5.37.1.8** `void* elf_open_obj (const char * file)`**5.37.2 Variable Documentation****5.37.2.1 struct bff bffs**

Initial value:

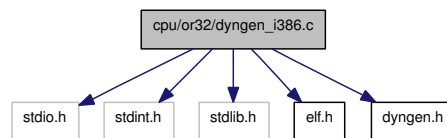
```

{
  elf_open_obj,
  elf_close_obj,
  elf_get_func_name,
  elf_get_func_start,
  elf_get_func_len,
  elf_get_func_reloc }
  
```

5.38 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

5.38.1 Define Documentation

5.38.1.1 #define RET_OPCODE 0xc3

5.38.2 Function Documentation

5.38.2.1 void i386_gen_func_reloc (FILE *f, struct reloc *reloc)

5.38.2.2 void i386_gen_reloc (FILE *f, struct reloc *reloc, unsigned int param)

5.38.2.3 unsigned int i386_get_real_func_len (void *f_start, unsigned int f_len, char *name)

5.38.3 Variable Documentation

5.38.3.1 struct archf archfs

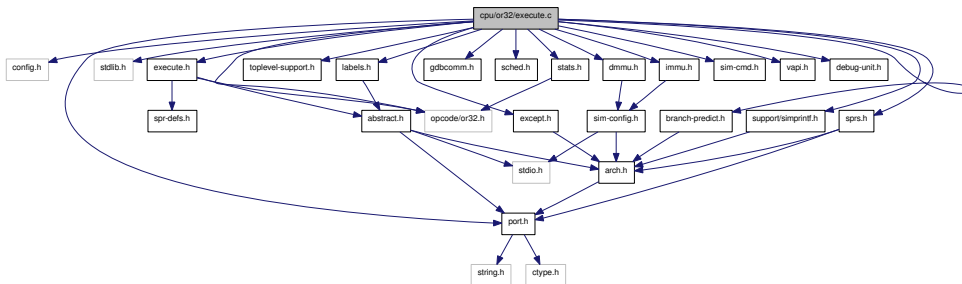
Initial value:


```
{
  i386_get_real_func_len,
  i386_gen_reloc,
  i386_gen_func_reloc
}
```

5.39 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 dependency graph for execute.c:



Defines

- #define [SET_OV_FLAG_FN\(value\)](#)

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 `next_delay_insn`
- static int `breakpoint`

5.39.1 Define Documentation

5.39.1.1 `#define SET_OV_FLAG_FN(value)`

5.39.2 Function Documentation

5.39.2.1 void `analysis` (`struct iqueue_entry * current`)

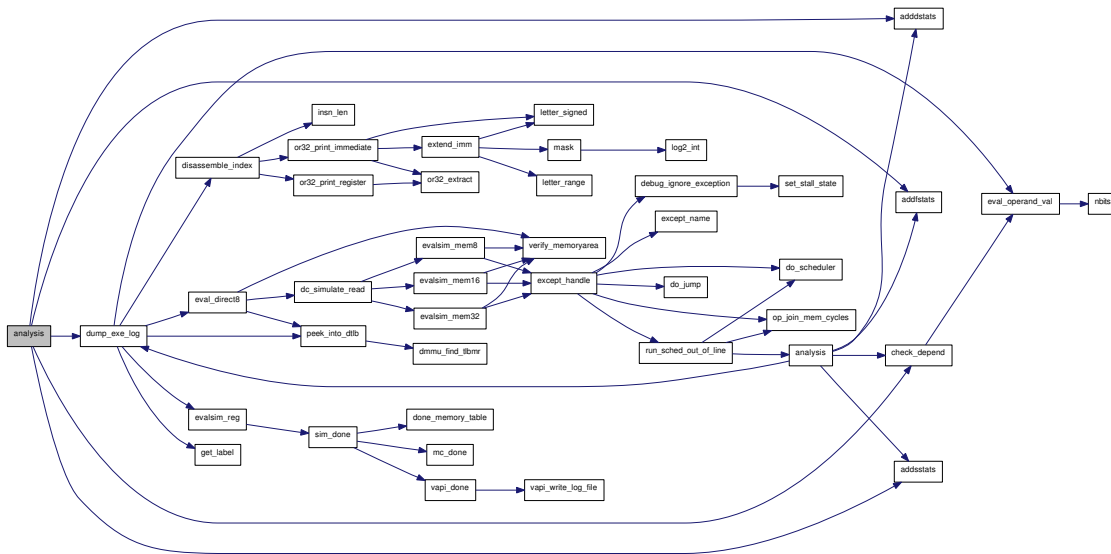
Perform analysis of the instruction being executed

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:



5.39.2.2 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:

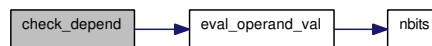
← *prev* Previous instruction

← *next* Next instruction

Returns:

Non-zero if yes.

Here is the call graph for this function:



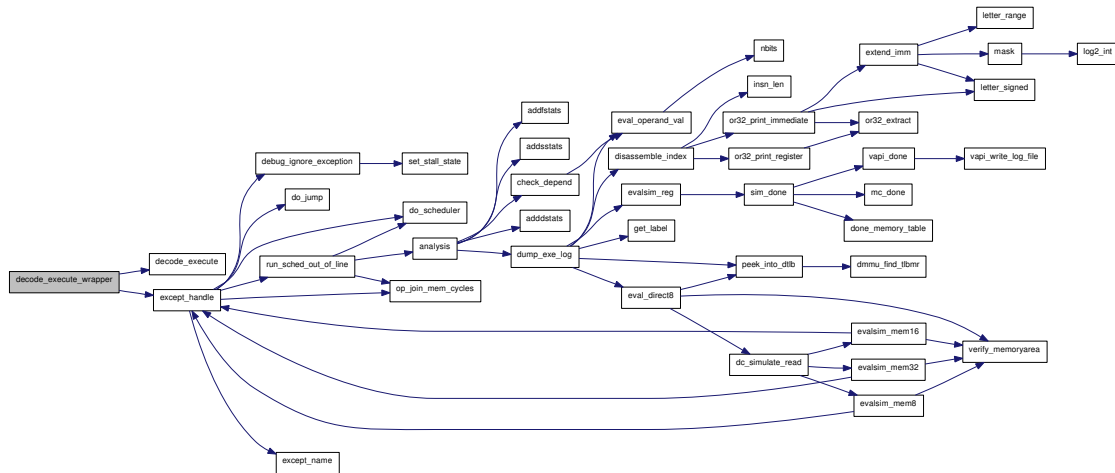
5.39.2.3 int cpu_clock ()

Simulates one CPU clock cycle

Returns:

non-zero if a breakpoint is hit, zero otherwise.

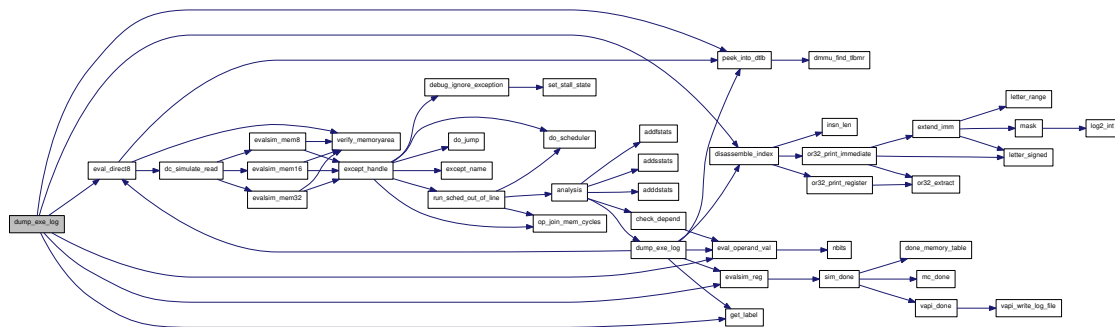
Here is the call graph for this function:



5.39.2.7 void dump_exe_log ()

Outputs disassembled instruction

Here is the call graph for this function:

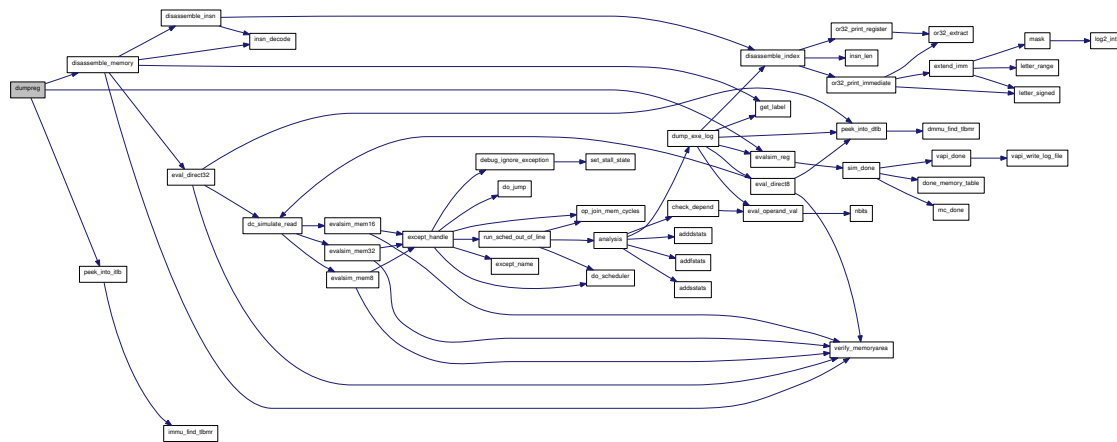


5.39.2.8 void dumpreg ()

Dump registers

Supports the CLI 'r' and 't' commands

Here is the call graph for this function:



5.39.2.9 uorreg_t eval_operand_val (uint32_t insn, struct insn_op_struct * opd)

Evaluates source operand

Implementation specific. Declared global, although this is only actually required for DYNAMIC_EXECUTION,

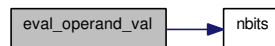
Parameters:

- ← *insn* The instruction
- ← *opd* The operand

Returns:

The value of the source operand

Here is the call graph for this function:



5.39.2.10 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\(\)](#).

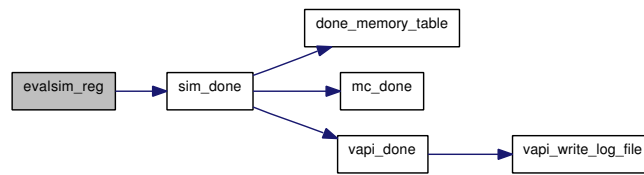
Parameters:

- ← *regno* The register of interest

Returns:

The value of the register

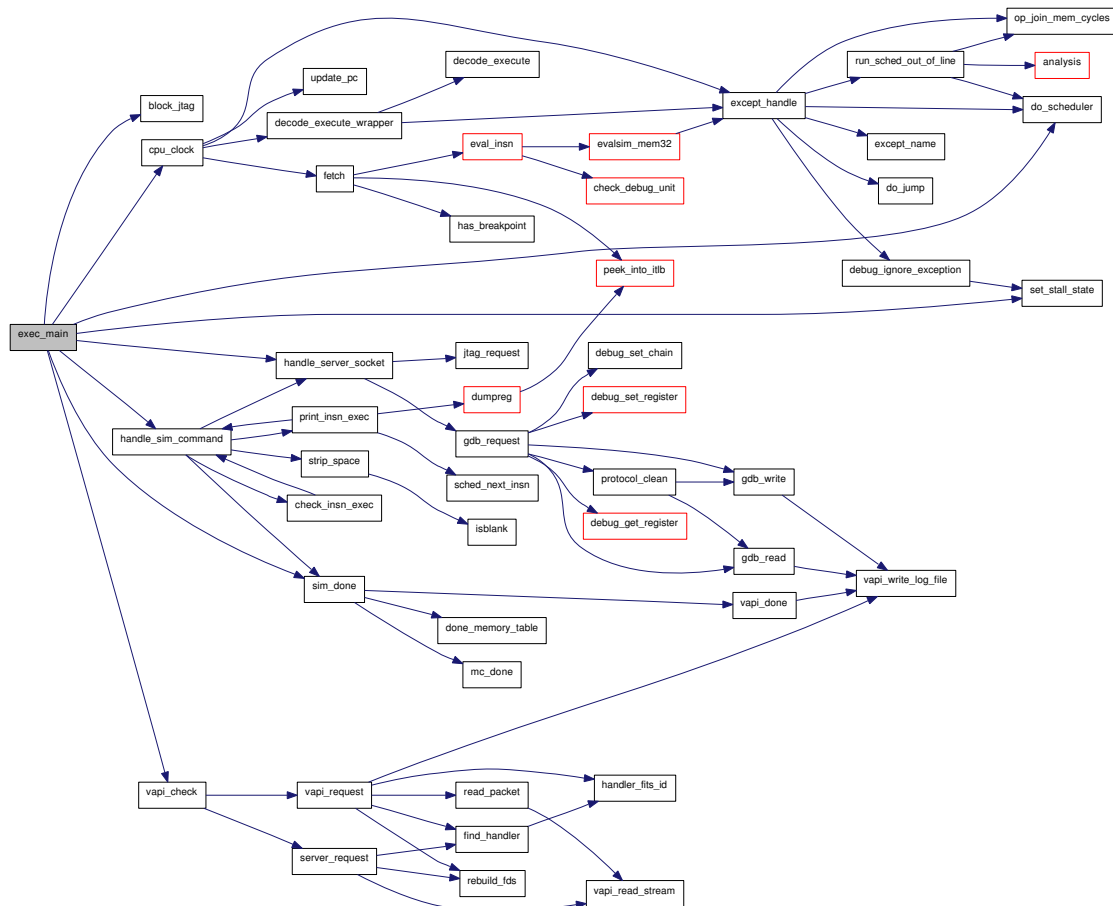
Here is the call graph for this function:



5.39.2.11 void exec_main ()

The main execution loop

Here is the call graph for this function:



5.39.2.12 static int fetch () [static]

Should instruction NOT be executed?

Modified by CZ 26/05/01 for new mode execution.

5.39.2.15 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

5.39.2.16 void setsim_reg (unsigned int regno, uorreg_t value)

Set a specific register with value

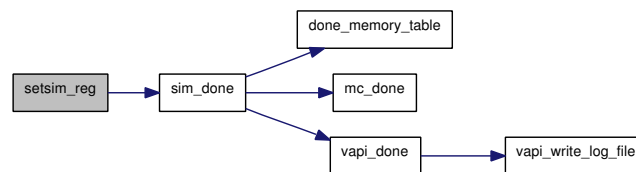
Implementation specific. Abort if we are given a duff register.

Parameters:

← *regno* The register of interest

← *value* The value to be set

Here is the call graph for this function:



5.39.2.17 static void update_pc () [static]

This code actually updates the PC value

5.39.3 Variable Documentation

5.39.3.1 int breakpoint [static]

5.39.3.2 struct cpu_state cpu_state

Current cpu state. Globally available.

5.39.3.3 int do_stats = 0

Whether we are doing statistical analysis. Globally available

5.39.3.4 struct hist_exec* hist_exec_tail = NULL

History of execution. Globally available

5.39.3.5 `int issued_per_cycle = 4` [static]

5.39.3.6 `int multissue[20]` [static]

5.39.3.7 `int next_delay_insn` [static]

5.39.3.8 `oraddr_t pnext`

Temporary program counter. Globally available

5.39.3.9 `int sbuf_buf[MAX_SBUF_LEN] = { 0 }` [static]

5.39.3.10 `int sbuf_count = 0` [static]

5.39.3.11 `int sbuf_head = 0` [static]

5.39.3.12 `int sbuf_prev_cycles = 0` [static]

5.39.3.13 `int sbuf_tail = 0` [static]

5.39.3.14 `int sbuf_total_cyc = 0`

Number of total store cycles. Globally available

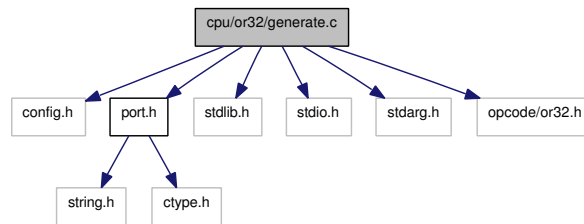
5.39.3.15 `int sbuf_wait_cyc = 0`

Num cycles waiting for stores to complete. Globally available

5.40 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[])

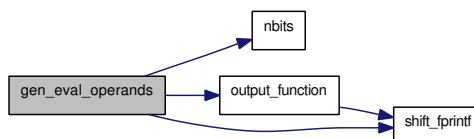
Variables

- static char * [in_file](#)
- static char * [out_file](#)
- static int [write_to_reg](#)
- static int [out_lines](#) = 0

5.40.1 Function Documentation

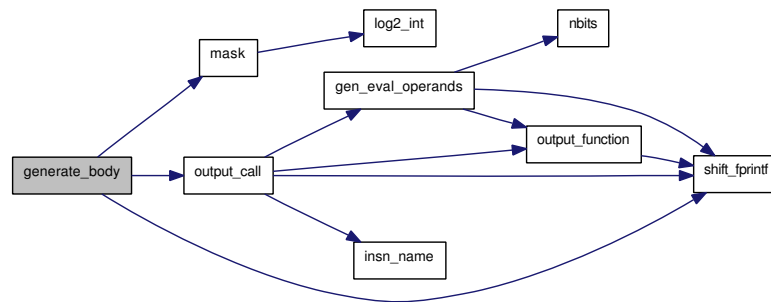
5.40.1.1 static int [gen_eval_operands](#) (FILE *fo, int *insn_index*, int *level*) [static]

Here is the call graph for this function:



5.40.1.2 static int generate_body (FILE *fo, unsigned long *a, unsigned long cur_mask, int level) [static]

Here is the call graph for this function:

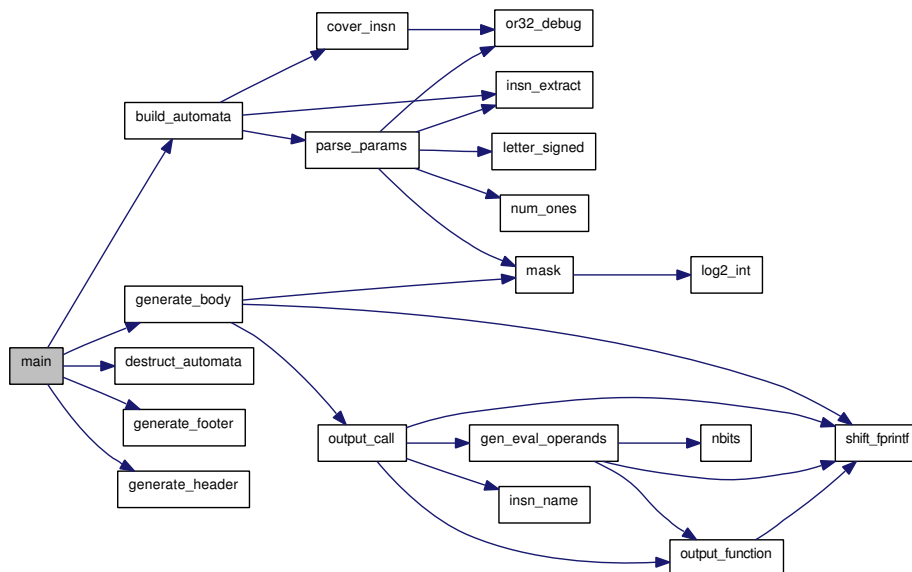


5.40.1.3 static int generate_footer (FILE *fo) [static]

5.40.1.4 static int generate_header (FILE *fo) [static]

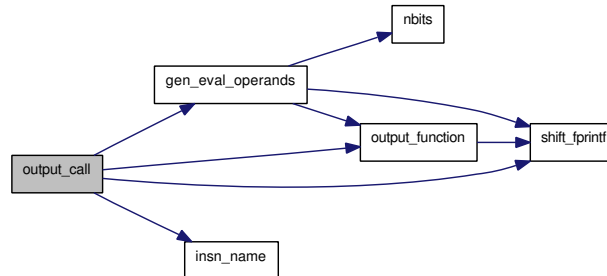
5.40.1.5 int main (int argc, char *argv[])

Here is the call graph for this function:



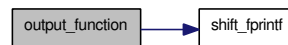
5.40.1.6 `static int output_call (FILE *fo, int index, int level)` [static]

Here is the call graph for this function:



5.40.1.7 `int output_function (FILE *fo, const char *func_name, int level)`

Here is the call graph for this function:



5.40.1.8 `static int shift_fprintf (int level, FILE *f, const char *fmt, ...)` [static]

5.40.2 Variable Documentation

5.40.2.1 `char* in_file` [static]

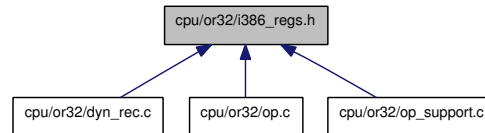
5.40.2.2 `char* out_file` [static]

5.40.2.3 `int out_lines = 0` [static]

5.40.2.4 `int write_to_reg` [static]

5.41 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

5.41.1 Define Documentation

5.41.1.1 #define CPU_STATE_REG "ebp"

5.41.1.2 #define NUM_T_REGS 3

5.41.1.3 #define T0_REG "ebx"

5.41.1.4 #define T1_REG "esi"

5.41.1.5 #define T2_REG "edi"

5.42 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)

5.42.1 Function Documentation

5.42.1.1 INSTRUCTION (l_cust4)

5.42.1.2 INSTRUCTION (l_cust3)

5.42.1.3 INSTRUCTION (l_cust2)

5.42.1.4 INSTRUCTION (l_cust1)

5.42.1.5 INSTRUCTION (lf_sub_s)

5.42.1.6 INSTRUCTION (lf_sfne_s)

5.42.1.7 INSTRUCTION (lf_sflt_s)

5.42.1.8 INSTRUCTION (lf_sfle_s)

5.42.1.9 INSTRUCTION (lf_sfgt_s)

5.42.1.10 INSTRUCTION (lf_sfge_s)

5.42.1.11 INSTRUCTION (lf_sfeq_s)

5.42.1.12 INSTRUCTION (lf_rem_s)

5.42.1.13 INSTRUCTION (lf_mul_s)

5.42.1.14 INSTRUCTION (lf_madd_s)

5.42.1.15 INSTRUCTION (lf_itof_s)

5.42.1.16 INSTRUCTION (lf_ftoi_s)

5.42.1.17 INSTRUCTION (lf_div_s)

5.42.1.18 INSTRUCTION (lf_add_s)

5.42.1.19 INSTRUCTION (l_ff1)

5.42.1.20 INSTRUCTION (l_cmov)

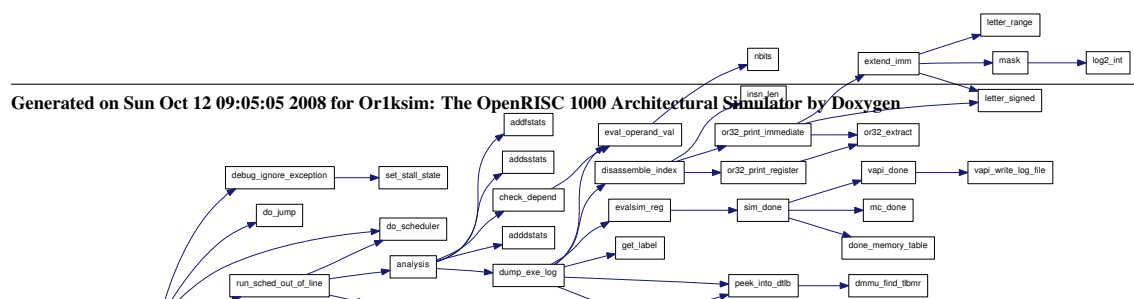
5.42.1.21 INSTRUCTION (l_macrc)

5.42.1.22 INSTRUCTION (l_msb)

5.42.1.23 INSTRUCTION (l_mac)

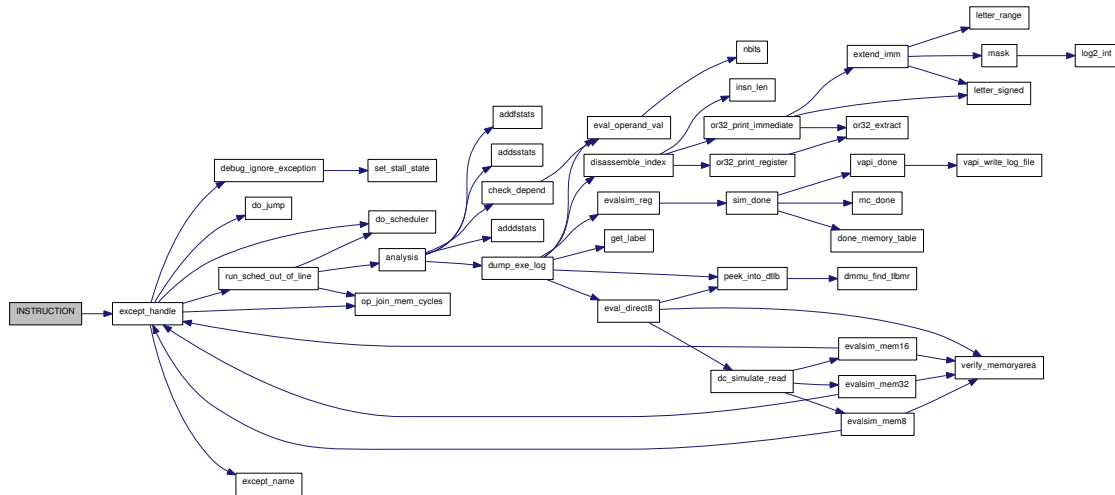
5.42.1.24 INSTRUCTION (l_trap)

Here is the call graph for this function:



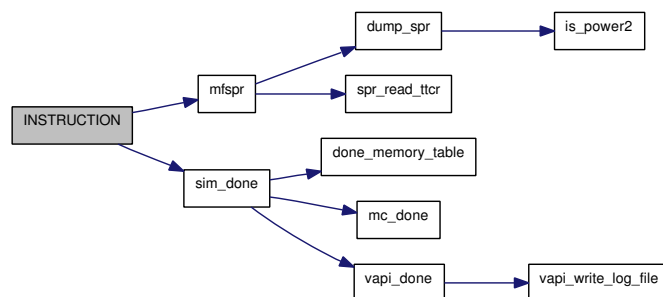
5.42.1.25 INSTRUCTION (l_sys)

Here is the call graph for this function:



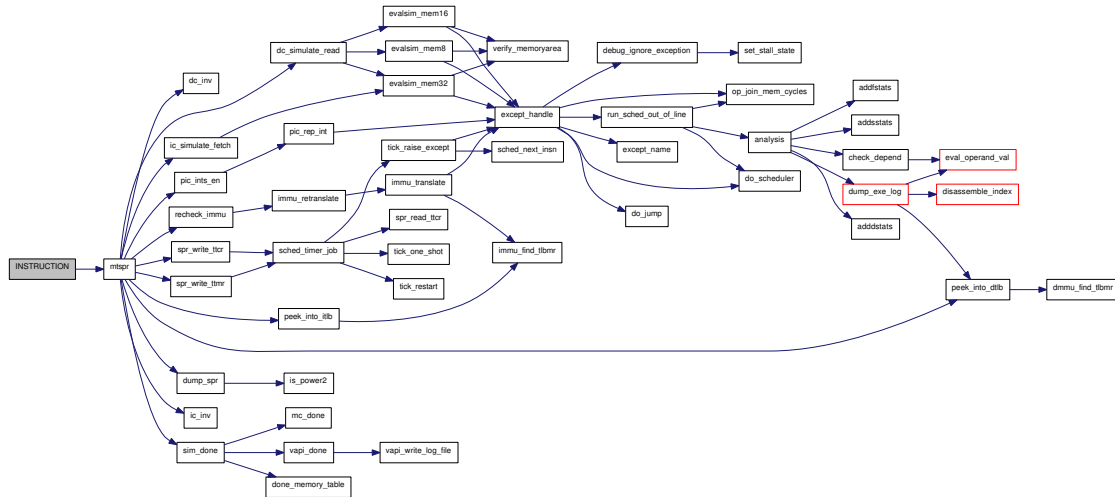
5.42.1.26 INSTRUCTION (l_mfspr)

Here is the call graph for this function:



5.42.1.45 INSTRUCTION (l_rfe)

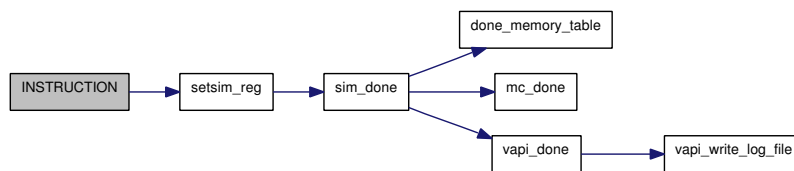
Here is the call graph for this function:



5.42.1.46 INSTRUCTION (l_jr)

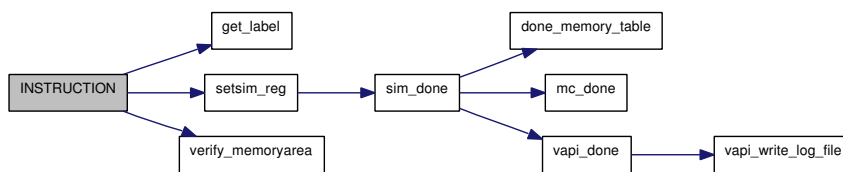
5.42.1.47 INSTRUCTION (l_jalr)

Here is the call graph for this function:



5.42.1.48 INSTRUCTION (l_jal)

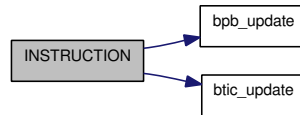
Here is the call graph for this function:



5.42.1.49 INSTRUCTION (l_j)

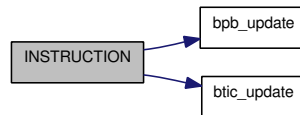
5.42.1.50 INSTRUCTION (l_bnf)

Here is the call graph for this function:



5.42.1.51 INSTRUCTION (l_bf)

Here is the call graph for this function:



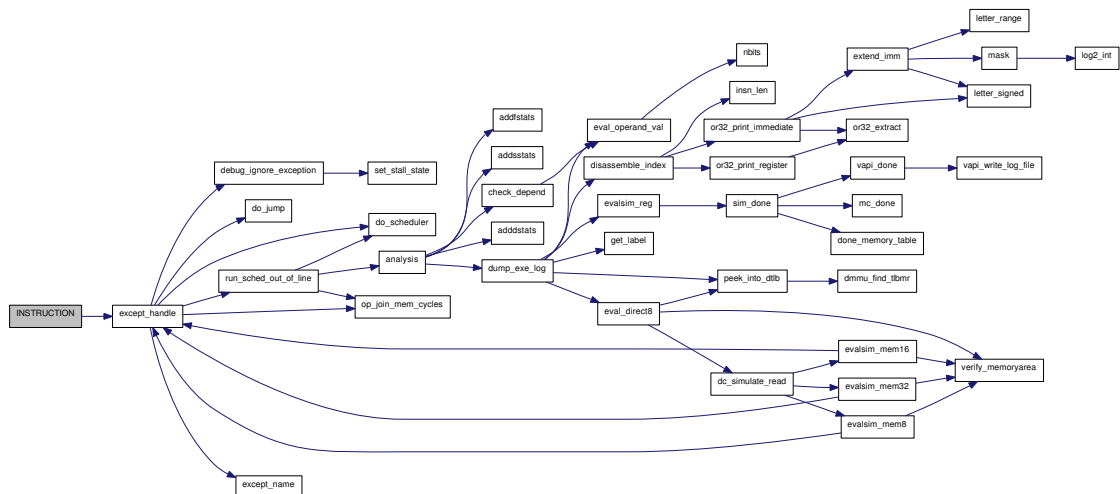
5.42.1.52 INSTRUCTION (l_srI)

5.42.1.53 INSTRUCTION (l_sra)

5.42.1.54 INSTRUCTION (l_slI)

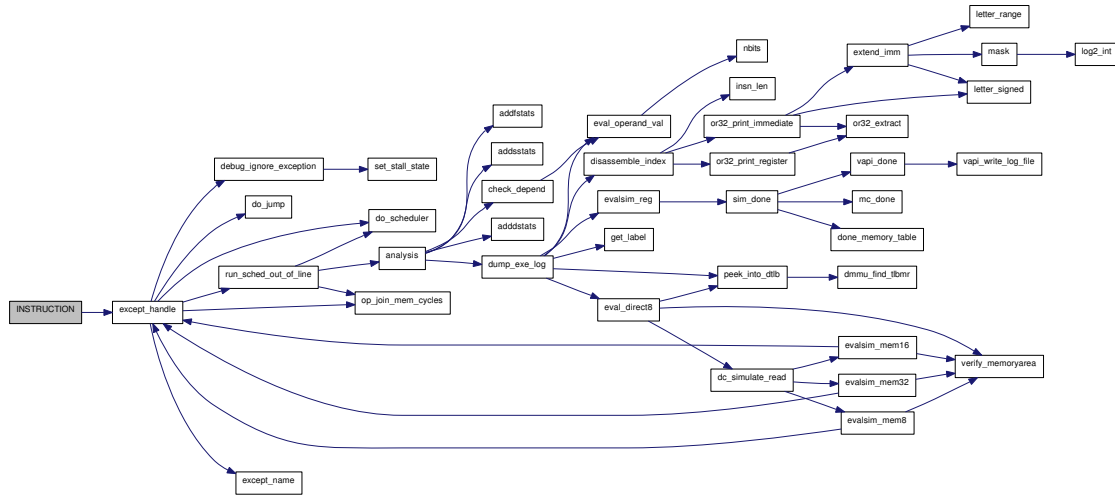
5.42.1.55 INSTRUCTION (l_divu)

Here is the call graph for this function:



5.42.1.56 INSTRUCTION (l_div)

Here is the call graph for this function:



5.42.1.57 INSTRUCTION (l_mul)

5.42.1.58 INSTRUCTION (l_sub)

5.42.1.59 INSTRUCTION (l_xor)

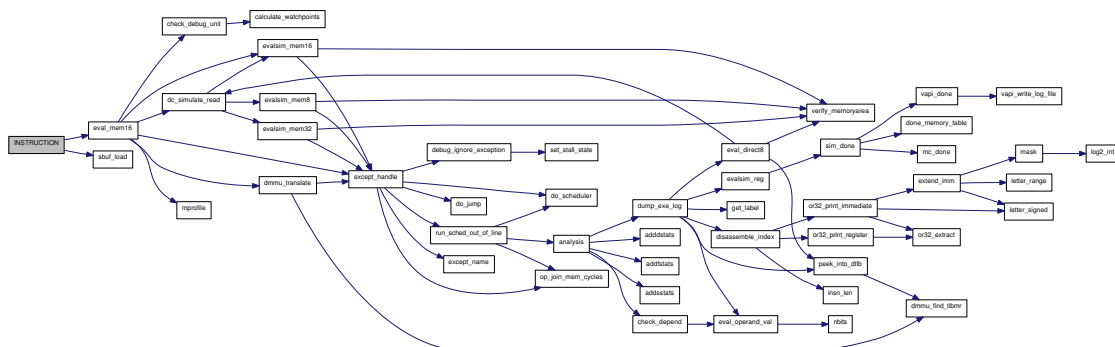
5.42.1.60 INSTRUCTION (l_or)

5.42.1.61 INSTRUCTION (l_and)

5.42.1.62 INSTRUCTION (l_movhi)

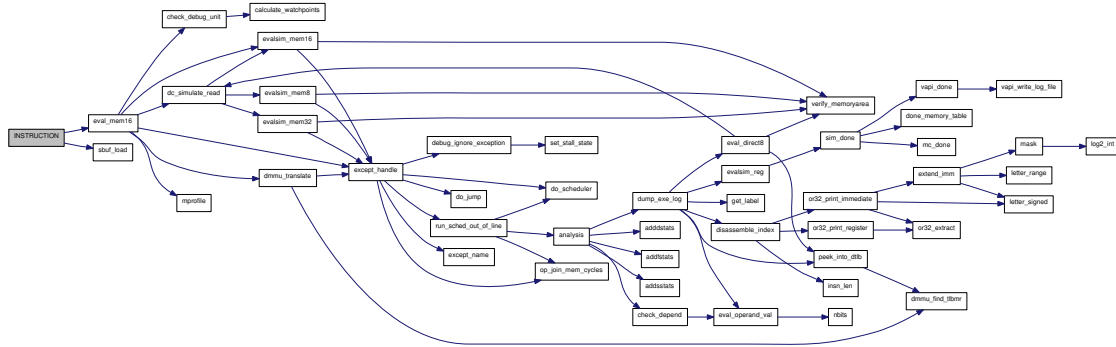
5.42.1.63 INSTRUCTION (l_lhz)

Here is the call graph for this function:



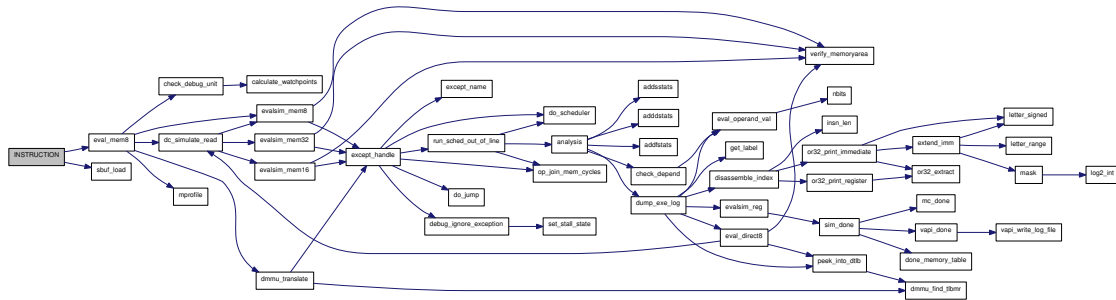
5.42.1.64 INSTRUCTION (l_lhs)

Here is the call graph for this function:



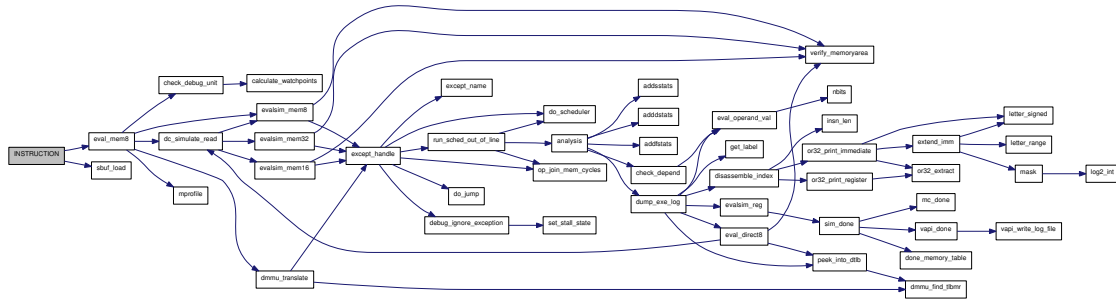
5.42.1.65 INSTRUCTION (l_ibz)

Here is the call graph for this function:



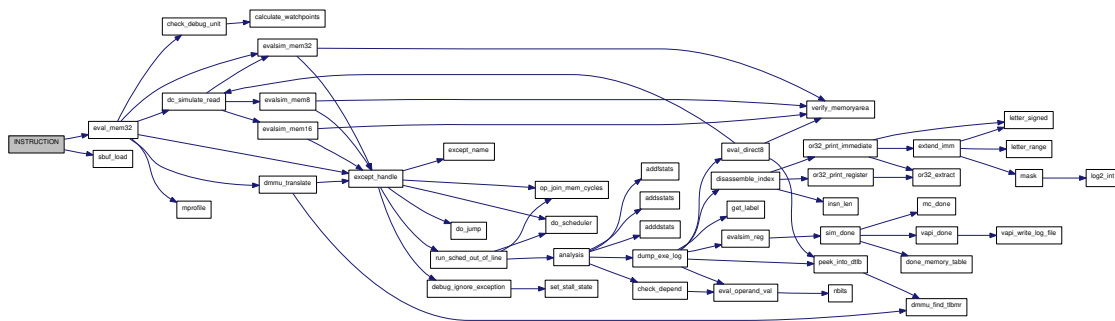
5.42.1.66 INSTRUCTION (l_lbs)

Here is the call graph for this function:



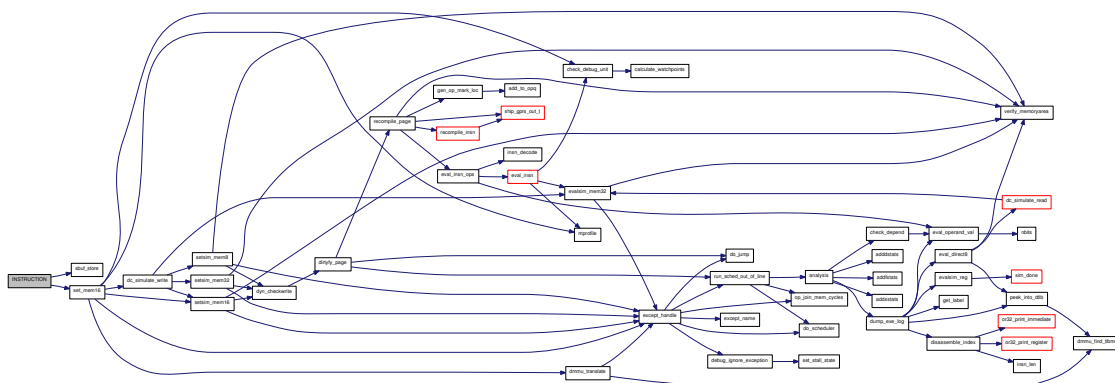
5.42.1.67 INSTRUCTION (l_lwz)

Here is the call graph for this function:



5.42.1.68 INSTRUCTION (l_sh)

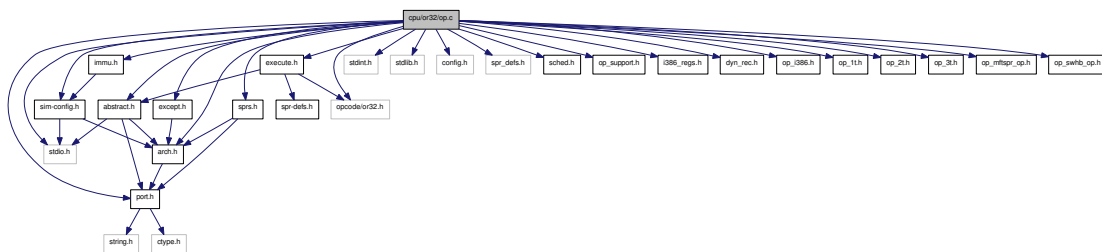
Here is the call graph for this function:



5.43 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` `eprc_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](#)

5.43.1 Define Documentation

5.43.1.1 `#define __or_dynop`

5.43.1.2 `#define COMP <=`

5.43.1.3 `#define COMP <`

5.43.1.4 `#define COMP >=`

5.43.1.5 `#define COMP >`

5.43.1.6 `#define COMP <=`

5.43.1.7 `#define COMP <`

5.43.1.8 `#define COMP >=`

5.43.1.9 `#define COMP >`

5.43.1.10 `#define COMP !=`

5.43.1.11 `#define COMP ==`

5.43.1.12 `#define COMP_CAST(x) (orreg_t)(x)`

5.43.1.13 `#define COMP_CAST(x) (orreg_t)(x)`

5.43.1.14 `#define COMP_CAST(x) (orreg_t)(x)`

5.43.1.15 `#define COMP_CAST(x) (orreg_t)(x)`

5.43.1.16 `#define COMP_CAST(x) (x)`

5.43.1.17 `#define COMP_CAST(x) (x)`

5.43.1.18 `#define COMP_CAST(x) (x)`

5.43.1.19 `#define COMP_CAST(x) (x)`

5.43.1.20 `#define COMP_CAST(x) (x)`

5.43.1.21 `#define COMP_CAST(x) (x)`

5.43.1.22 `#define COMP_NAME sfles`

5.43.1.23 `#define COMP_NAME sflts`

5.43.1.24 `#define COMP_NAME sfges`

5.43.1.25 `#define COMP_NAME sfgts`

5.43.1.26 `#define COMP_NAME sfleu`

5.43.1.27 `#define COMP_NAME sfltu`

5.43.1.28 `#define COMP_NAME sfgeu`

5.43.1.29 `#define COMP_NAME sfgtu`

5.43.1.30 `#define COMP_NAME sfne`

5.43.2.6 void do_sched_wrap_delay (void)

Here is the call graph for this function:

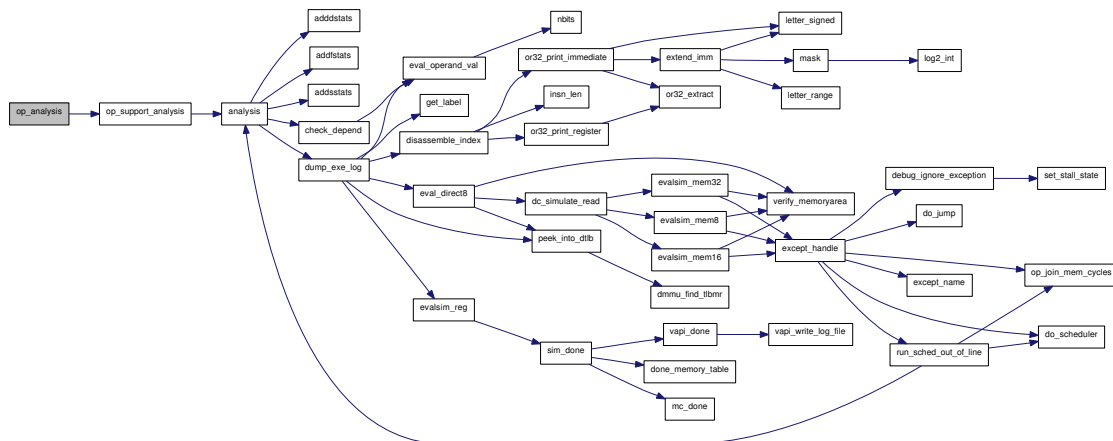


5.43.2.7 void enter_dyn_code (oraddr_t addr, struct dyn_page * dp)

5.43.2.8 __or_dynop void op_add_pc (void)

5.43.2.9 __or_dynop void op_analysis (void)

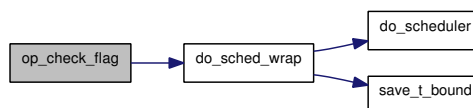
Here is the call graph for this function:



5.43.2.10 __or_dynop void op_check_delay_slot (void)

5.43.2.11 __or_dynop void op_check_flag (void)

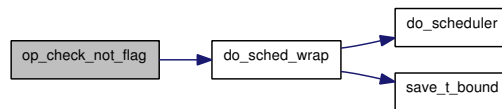
Here is the call graph for this function:



5.43.2.12 `__or_dynop void op_check_flag_delay (void)`

5.43.2.13 `__or_dynop void op_check_not_flag (void)`

Here is the call graph for this function:



5.43.2.14 `__or_dynop void op_check_not_flag_delay (void)`

5.43.2.15 `__or_dynop void op_clear_delay_insn (void)`

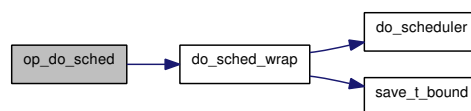
5.43.2.16 `__or_dynop void op_clear_flag (void)`

5.43.2.17 `__or_dynop void op_clear_pc_delay (void)`

5.43.2.18 `__or_dynop void op_do_jump_delay (void)`

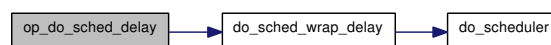
5.43.2.19 `__or_dynop void op_do_sched (void)`

Here is the call graph for this function:



5.43.2.20 `__or_dynop void op_do_sched_delay (void)`

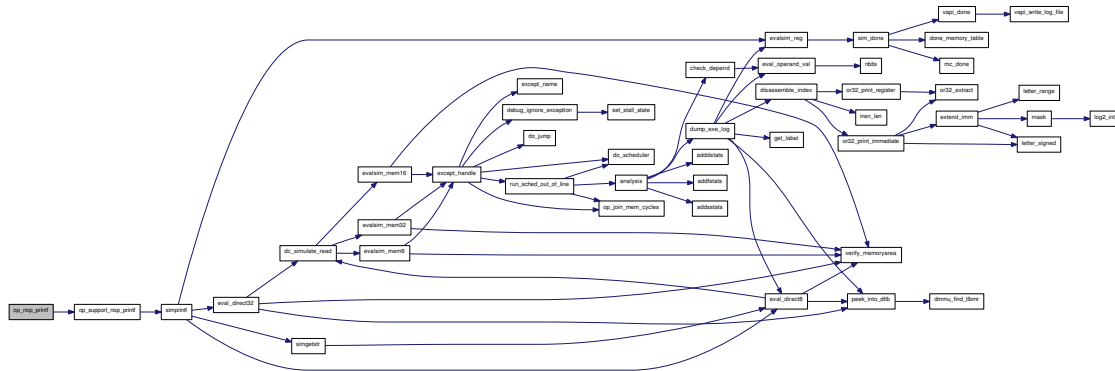
Here is the call graph for this function:



- 5.43.2.21 `__or_dynop void op_illegal (void)`
- 5.43.2.22 `__or_dynop void op_illegal_delay (void)`
- 5.43.2.23 `__or_dynop void op_jump_imm (void)`
- 5.43.2.24 `__or_dynop void op_join_mem_cycles (void)`
- 5.43.2.25 `__or_dynop void op_macc (void)`
- 5.43.2.26 `__or_dynop void op_move_gpr10_pc_delay (void)`
- 5.43.2.27 `__or_dynop void op_move_gpr11_pc_delay (void)`
- 5.43.2.28 `__or_dynop void op_move_gpr12_pc_delay (void)`
- 5.43.2.29 `__or_dynop void op_move_gpr13_pc_delay (void)`
- 5.43.2.30 `__or_dynop void op_move_gpr14_pc_delay (void)`
- 5.43.2.31 `__or_dynop void op_move_gpr15_pc_delay (void)`
- 5.43.2.32 `__or_dynop void op_move_gpr16_pc_delay (void)`
- 5.43.2.33 `__or_dynop void op_move_gpr17_pc_delay (void)`
- 5.43.2.34 `__or_dynop void op_move_gpr18_pc_delay (void)`
- 5.43.2.35 `__or_dynop void op_move_gpr19_pc_delay (void)`
- 5.43.2.36 `__or_dynop void op_move_gpr1_pc_delay (void)`
- 5.43.2.37 `__or_dynop void op_move_gpr20_pc_delay (void)`
- 5.43.2.38 `__or_dynop void op_move_gpr21_pc_delay (void)`
- 5.43.2.39 `__or_dynop void op_move_gpr22_pc_delay (void)`
- 5.43.2.40 `__or_dynop void op_move_gpr23_pc_delay (void)`
- 5.43.2.41 `__or_dynop void op_move_gpr24_pc_delay (void)`
- 5.43.2.42 `__or_dynop void op_move_gpr25_pc_delay (void)`
- 5.43.2.43 `__or_dynop void op_move_gpr26_pc_delay (void)`
- 5.43.2.44 `__or_dynop void op_move_gpr27_pc_delay (void)`
- 5.43.2.45 `__or_dynop void op_move_gpr28_pc_delay (void)`
- 5.43.2.46 `__or_dynop void op_move_gpr29_pc_delay (void)`
- 5.43.2.47 `__or_dynop void op_move_gpr2_pc_delay (void)`
- 5.43.2.48 `__or_dynop void op_move_gpr30_pc_delay (void)`
- 5.43.2.49 `__or_dynop void op_move_gpr31_pc_delay (void)`
- 5.43.2.50 `__or_dynop void op_move_gpr3_pc_delay (void)`
- 5.43.2.51 `__or_dynop void op_move_gpr4_pc_delay (void)`

5.43.2.58 __or_dynop void op_nop_printf (void)

Here is the call graph for this function:



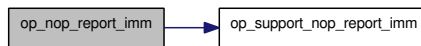
5.43.2.59 __or_dynop void op_nop_report (void)

Here is the call graph for this function:



5.43.2.60 __or_dynop void op_nop_report_imm (void)

Here is the call graph for this function:



5.43.2.61 __or_dynop void op_nop_reset (void)

Here is the call graph for this function:



5.43.2.62 `__or_dynop void op_prep_rfe (void)`**5.43.2.63** `__or_dynop void op_prep_sys (void)`

Here is the call graph for this function:

**5.43.2.64** `__or_dynop void op_prep_sys_delay (void)`

Here is the call graph for this function:

**5.43.2.65** `__or_dynop void op_prep_trap (void)`

Here is the call graph for this function:

**5.43.2.66** `__or_dynop void op_prep_trap_delay (void)`

Here is the call graph for this function:



5.43.2.67 `__or_dynop void op_set_delay_insn (void)`

5.43.2.68 `__or_dynop void op_set_flag (void)`

5.43.2.69 `__or_dynop void op_set_pc_delay_imm (void)`

5.43.2.70 `__or_dynop void op_set_pc_delay_pc (void)`

5.43.2.71 `__or_dynop void op_set_pc_pc_delay (void)`

5.43.2.72 `__or_dynop void op_store_insn_ea (void)`

5.43.2.73 `__or_dynop void op_store_link_addr_gpr (void)`

5.43.2.74 `static void prep_except (oraddr_t epcr_base) [inline, static]`

5.43.2.75 `static void save_t_bound (oraddr_t pc) [inline, static]`

5.43.3 Variable Documentation

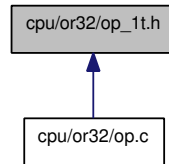
5.43.3.1 `uorreg_t __op_param1`

5.43.3.2 `uorreg_t __op_param2`

5.43.3.3 `uorreg_t __op_param3`

5.44 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)

5.44.1 Define Documentation

5.44.1.1 #define OP_1T

5.44.1.2 #define T glue(_, T0)

5.45 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)`

5.45.1 Function Documentation

5.45.1.1 `__or_dynop void glue (op_mac_imm, T)`

5.45.1.2 `__or_dynop void glue (op_macrc, T)`

5.45.1.3 `__or_dynop void glue (op_calc_insn_ea, T)`

5.45.1.4 `__or_dynop void glue (op_check_null_except, T)`

Here is the call graph for this function:



5.45.1.5 `__or_dynop void glue (op_check_null_except_delay, T)`

Here is the call graph for this function:

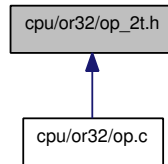


5.45.1.6 `__or_dynop void glue (op_clear, T)`

5.45.1.7 `__or_dynop void glue (op_imm, T)`

5.46 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)

5.46.1 Define Documentation

5.46.1.1 #define OP_2T

5.46.1.2 #define T glue(glue(glue(_, T0), _), T1)

5.47 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)`

5.47.1 Function Documentation

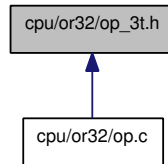
5.47.1.1 `__or_dynop void glue (op_neg, T)`

5.47.1.2 `__or_dynop void glue (op_ff1, T)`

5.47.1.3 `__or_dynop void glue (op_move, T)`

5.48 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)

5.48.1 Define Documentation

5.48.1.1 #define OP_3T

5.48.1.2 #define **T** glue(glue(glue(glue(glue(_, T0), _), T1), _), T2)

5.49 cpu/or32/op_3t_op.h File Reference

Functions

- `__or_dynop void glue (op_cmov, T)(void)`

5.49.1 Function Documentation

5.49.1.1 `__or_dynop void glue (op_cmov, T)`

5.50 `cpu/or32/op_arith_op.h` File Reference

5.51 cpu/or32/op_comp_op.h File Reference

5.52 cpu/or32/op_extend_op.h File Reference

Functions

- void [glue](#) (glue(op_, EXT_NAME), T)(void)

5.52.1 Function Documentation

5.52.1.1 void glue (glue(op_, EXT_NAME), T)

5.53 cpu/or32/op_ff1_op.h File Reference

Functions

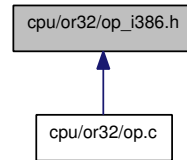
- `__or_dynop void glue(glue(glue(op_ff1_, DST_T), _), SRC_T)(void)`

5.53.1 Function Documentation

5.53.1.1 `__or_dynop void glue(glue(glue(op_ff1_, DST_T), _), SRC_T)`

5.54 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")`

5.54.1 Define Documentation

5.54.1.1 #define **FORCE_RET** `asm volatile ("")`

5.54.1.2 #define **OP_JUMP**(x) `asm("jmp *%0" : : "rm" (x))`

5.54.1.3 #define **SPEEDY_CALL**(func) `asm("call "#func"\n")`

5.54.2 Function Documentation

5.54.2.1 `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")`

5.55 `cpu/or32/op_lwhb_op.h` File Reference

5.56 cpu/or32/op_mac_op.h File Reference

Functions

- `__or_dynop void glue (glue(op_, OP_NAME), T)(void)`

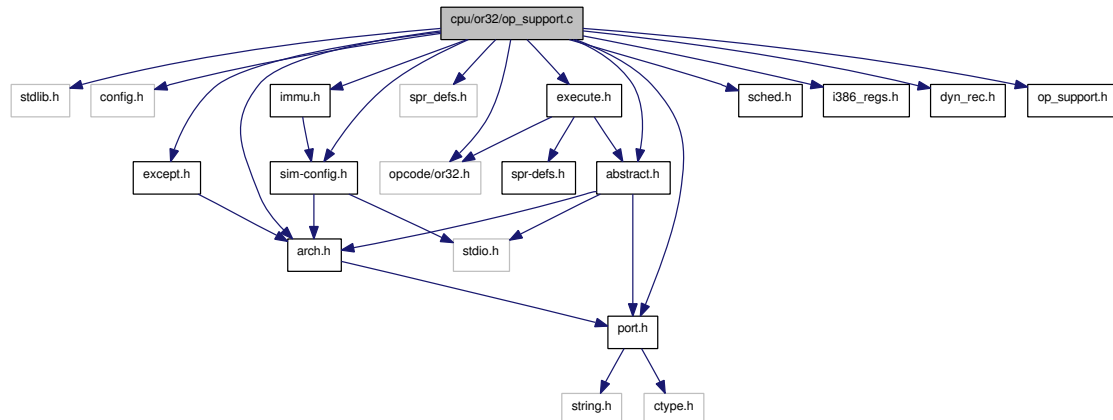
5.56.1 Function Documentation

5.56.1.1 `__or_dynop void glue (glue(op_, OP_NAME), T)`

5.58 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 dependency graph for op_support.c:



Functions

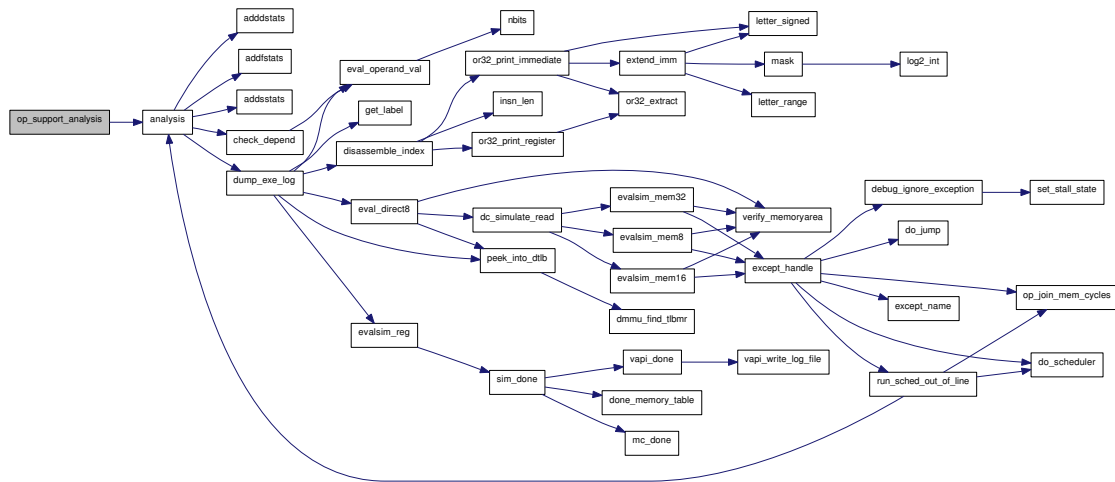
- void [op_support_nop_exit](#) (void)
- void [op_support_nop_reset](#) (void)
- void [op_support_nop_printf](#) (void)
- void [op_support_nop_report](#) (void)
- void [op_support_nop_report_imm](#) (int imm)
- void [do_jump](#) (oraddr_t addr)
- void [op_support_analysis](#) (void)

5.58.1 Function Documentation

5.58.1.1 void do_jump (oraddr_t addr)

5.58.1.2 void op_support_analysis (void)

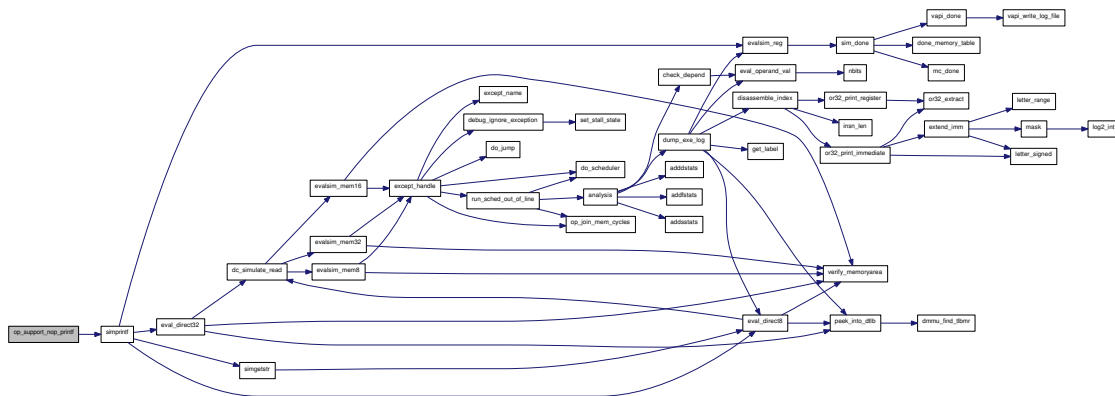
Here is the call graph for this function:



5.58.1.3 void op_support_nop_exit (void)

5.58.1.4 void op_support_nop_printf (void)

Here is the call graph for this function:



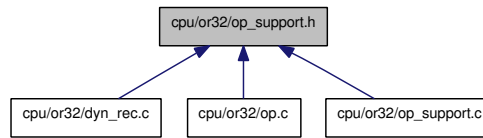
5.58.1.5 void `op_support_nop_report` (void)

5.58.1.6 void `op_support_nop_report_imm` (int *imm*)

5.58.1.7 void `op_support_nop_reset` (void)

5.59 cpu/or32/op_support.h File Reference

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



Functions

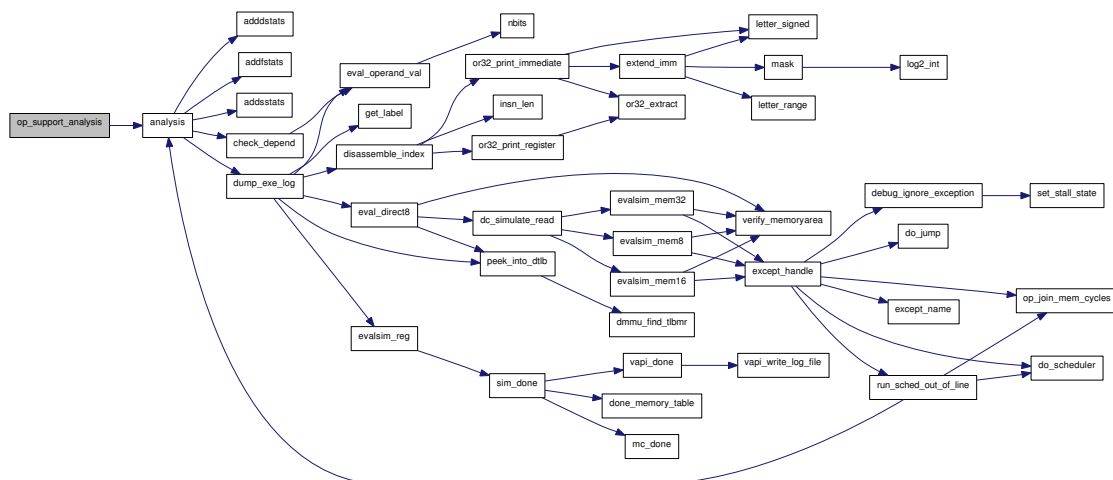
- void [op_support_nop_exit](#) (void)
- void [op_support_nop_reset](#) (void)
- void [op_support_nop_printf](#) (void)
- void [op_support_nop_report](#) (void)
- void [op_support_nop_report_imm](#) (int imm)
- void [op_support_analysis](#) (void)
- void [do_jump](#) (oraddr_t addr)
- void [upd_reg_from_t](#) (oraddr_t pc, int bound)

5.59.1 Function Documentation

5.59.1.1 void do_jump (oraddr_t addr)

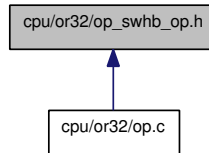
5.59.1.2 void op_support_analysis (void)

Here is the call graph for this function:



5.60 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)`

5.60.1 Function Documentation

5.60.1.1 `__or_dynop void glue (glue(op_, S_OP_NAME), _clear_imm)`

5.61 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)`

5.61.1 Function Documentation

5.61.1.1 `__or_dynop void glue (op_ttg_gpr31, T)`

5.61.1.2 `__or_dynop void glue (op_ttg_gpr30, T)`

5.61.1.3 `__or_dynop void glue (op_ttg_gpr29, T)`

5.61.1.4 `__or_dynop void glue (op_ttg_gpr28, T)`

5.61.1.5 `__or_dynop void glue (op_ttg_gpr27, T)`

5.61.1.6 `__or_dynop void glue (op_ttg_gpr26, T)`

5.61.1.7 `__or_dynop void glue (op_ttg_gpr25, T)`

5.61.1.8 `__or_dynop void glue (op_ttg_gpr24, T)`

5.61.1.9 `__or_dynop void glue (op_ttg_gpr23, T)`

5.61.1.10 `__or_dynop void glue (op_ttg_gpr22, T)`

5.61.1.11 `__or_dynop void glue (op_ttg_gpr21, T)`

5.61.1.12 `__or_dynop void glue (op_ttg_gpr20, T)`

5.61.1.13 `__or_dynop void glue (op_ttg_gpr19, T)`

5.61.1.14 `__or_dynop void glue (op_ttg_gpr18, T)`

5.61.1.15 `__or_dynop void glue (op_ttg_gpr17, T)`

5.61.1.16 `__or_dynop void glue (op_ttg_gpr16, T)`

5.61.1.17 `__or_dynop void glue (op_ttg_gpr15, T)`

5.61.1.18 `__or_dynop void glue (op_ttg_gpr14, T)`

5.61.1.19 `__or_dynop void glue (op_ttg_gpr13, T)`

5.61.1.20 `__or_dynop void glue (op_ttg_gpr12, T)`

5.61.1.21 `__or_dynop void glue (op_ttg_gpr11, T)`

5.61.1.22 `__or_dynop void glue (op_ttg_gpr10, T)`

5.61.1.23 `__or_dynop void glue (op_ttg_gpr9, T)`

5.61.1.24 `__or_dynop void glue (op_ttg_gpr8, T)`

5.61.1.25 `__or_dynop void glue (op_ttg_gpr7, T)`

5.61.1.26 `__or_dynop void glue (op_ttg_gpr6, T)`

5.61.1.27 `__or_dynop void glue (op_ttg_gpr5, T)`

5.61.1.28 `__or_dynop void glue (op_ttg_gpr4, T)`

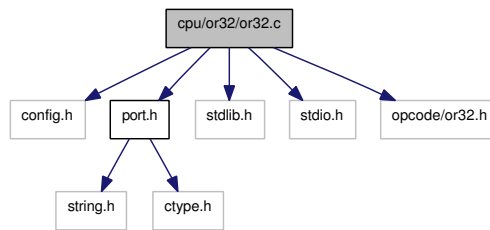
5.61.1.29 `__or_dynop void glue (op_ttg_gpr3, T)`

5.61.1.30 `__or_dynop void glue (op_ttg_gpr2, T)`

5.62 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]

5.62.1 Define Documentation

5.62.1.1 `#define EF(func) EFN`

5.62.1.2 `#define EFI EFN`

5.62.1.3 `#define EFN &l_none`

5.62.1.4 `#define MAX_AUTOMATA_SIZE (1200)`

5.62.1.5 `#define MAX_LEN (8)`

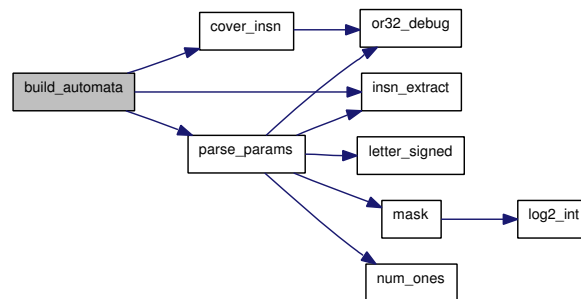
5.62.1.6 `#define MAX_OP_TABLE_SIZE (1200)`

5.62.1.7 `#define MIN(x, y) ((x) < (y) ? (x) : (y))`

5.62.2 Function Documentation

5.62.2.1 `void build_automata ()`

Here is the call graph for this function:



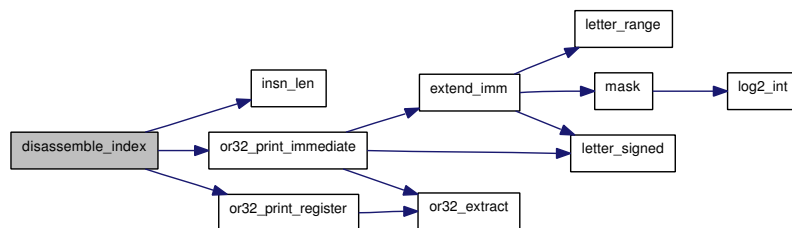
5.62.2.2 `static unsigned long* cover_insn (unsigned long * cur, int pass, unsigned int mask)`
`[static]`

Here is the call graph for this function:

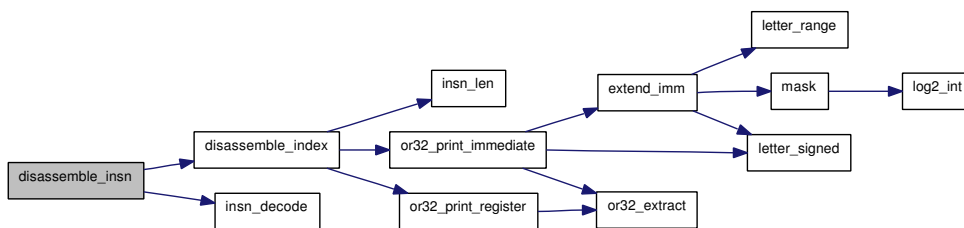


5.62.2.3 void destruct_automata ()**5.62.2.4 int disassemble_index (unsigned long *insn*, int *index*)**

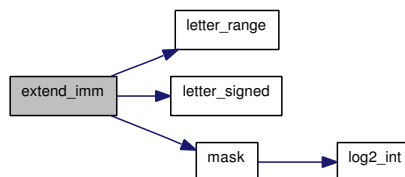
Here is the call graph for this function:

**5.62.2.5 int disassemble_insn (unsigned long *insn*)**

Here is the call graph for this function:

**5.62.2.6 unsigned long extend_imm (unsigned long *imm*, char *l*)**

Here is the call graph for this function:



5.62.2.7 int `insn_decode` (unsigned int *insn*)

5.62.2.8 unsigned long `insn_extract` (char *param_ch*, char * *enc_initial*)

5.62.2.9 int `insn_index` (char * *insn*)

5.62.2.10 int `insn_len` (int *insn_index*)

5.62.2.11 CONST char* `insn_name` (int *index*)

5.62.2.12 void `l_none` ()

5.62.2.13 int `letter_range` (char *l*)

5.62.2.14 int `letter_signed` (char *l*)

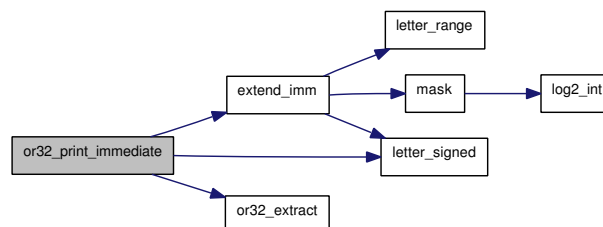
5.62.2.15 static int `num_ones` (unsigned long *value*) [static]

5.62.2.16 static void `or32_debug` (int *level*, const char * *format*, ...) [static]

5.62.2.17 unsigned long `or32_extract` (char *param_ch*, char * *enc_initial*, unsigned long *insn*)

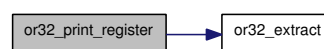
5.62.2.18 static char* `or32_print_immediate` (char * *dest*, char *param_ch*, char * *encoding*, unsigned long *insn*) [static]

Here is the call graph for this function:



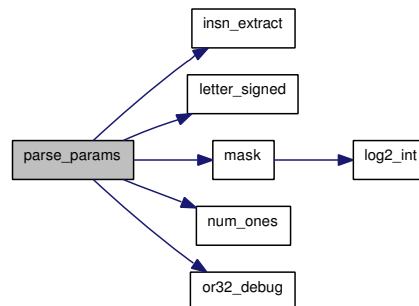
5.62.2.19 static char* `or32_print_register` (char * *dest*, char *param_ch*, char * *encoding*, unsigned long *insn*) [static]

Here is the call graph for this function:



5.62.2.20 `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:



5.62.3 Variable Documentation

5.62.3.1 `unsigned long* automata`

5.62.3.2 `int curpass = 0`

5.62.3.3 `char* disassembled = &disassembled_str[0]`

5.62.3.4 `char disassembled_str[50]` [static]

5.62.3.5 `CONST int num_opcodes`

Initial value:

```
((sizeof (or32_opcodes)) / (sizeof (struct or32_opcode))) - 1
```

5.62.3.6 `int nuncovered`

5.62.3.7 `struct insn_op_struct* op_data`

5.62.3.8 `struct insn_op_struct ** op_start`

5.62.3.9 `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}
}
```

5.62.3.10 `CONST struct or32_opcode or32_opcodes[]`

5.62.3.11 `int range_cache[256] = { 0 } [static]`

5.62.3.12 `struct temp_insn_struct* ti`

5.63 cpu/or32/rec_i386.h File Reference

Functions

- static void * [get_sp](#) (void)

5.63.1 Function Documentation

5.63.1.1 static void* [get_sp](#) (void) [static]

5.64 `cpu/or32/sched_i386.h` File Reference

Functions

- static void `set_sched_cycle` (`int32_t job_time`)

5.64.1 Function Documentation

5.64.1.1 `static void set_sched_cycle (int32_t job_time)` `[static]`

5.65 cpu/or32/simpl32_defs.h File Reference

Functions

- void l_invalid PARAMS ((struct iqueue_entry *))
- void l_sfgeu PARAMS ()(struct iqueue_entry *)

5.65.1 Function Documentation

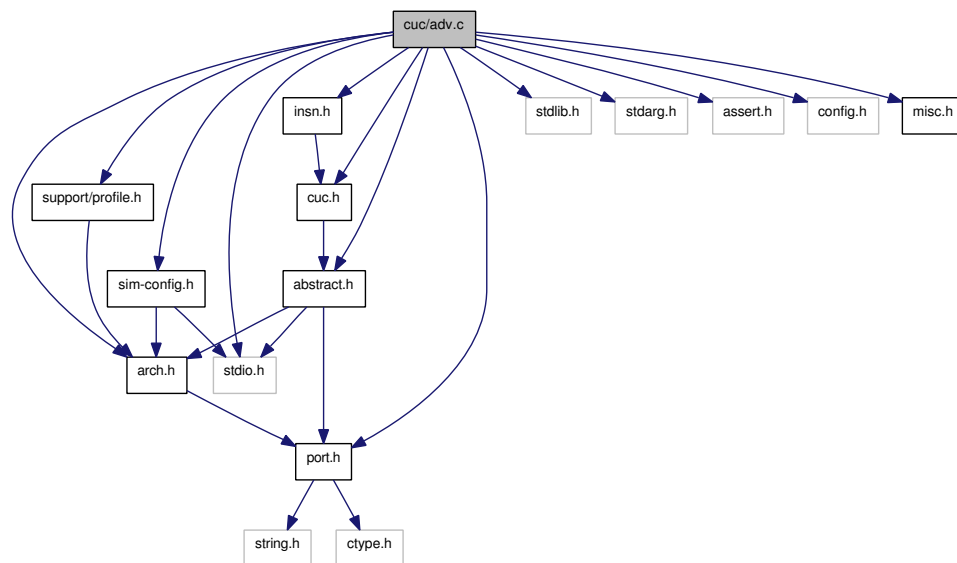
5.65.1.1 void l_sfgeu PARAMS ()

5.65.1.2 void l_none PARAMS ((struct iqueue_entry *))

5.66 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)

5.66.1 Function Documentation

5.66.1.1 static unsigned long calc_max (cuc_func *f, int ref) [static]

Here is the call graph for this function:



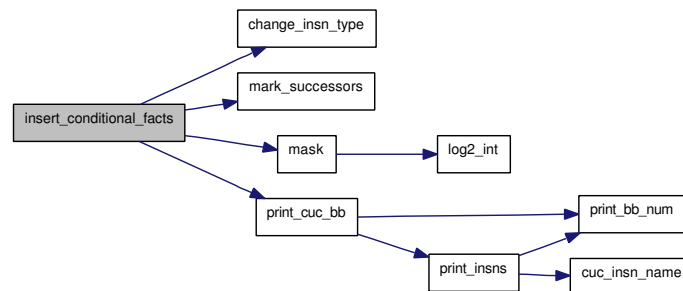
5.66.1.2 void detect_max_values (cuc_func *f)

Here is the call graph for this function:



5.66.1.3 void insert_conditional_facts (cuc_func *f)

Here is the call graph for this function:



5.66.1.4 static void mark_successors (cuc_func *f, int b, int m, int stopb) [static]

5.66.1.5 static unsigned long mask (unsigned long c) [static]

Here is the call graph for this function:

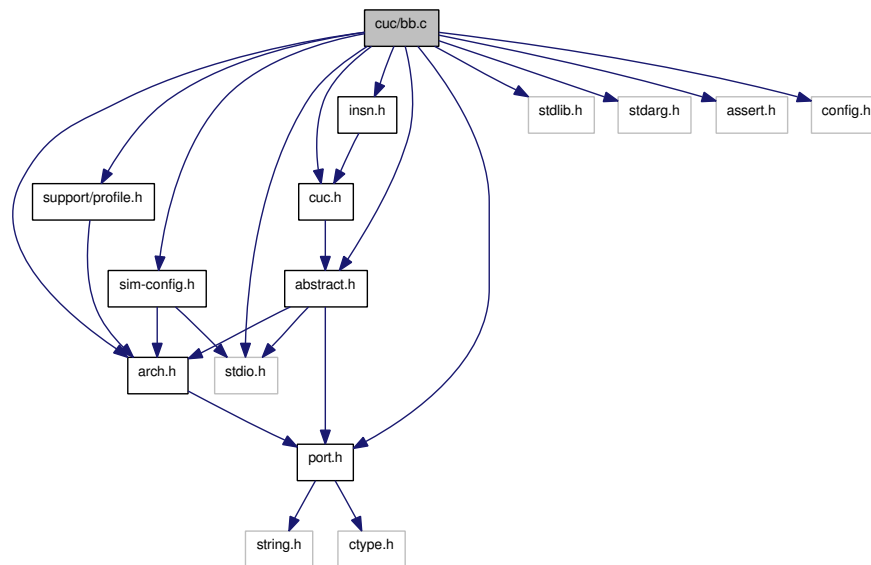


5.66.1.6 static unsigned long max_op (cuc_func *f, int ref, int o) [static]

5.67 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

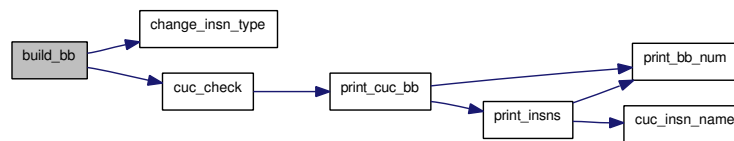
- void [print_bb_num](#) (int num)
- void [print_cuc_bb](#) (cuc_func *f, char *s)
- void [cpy_bb](#) (cuc_bb *dest, cuc_bb *src)
- cuc_func * [dup_func](#) (cuc_func *f)
- void [free_func](#) (cuc_func *f)
- void [recalc_last_used_reg](#) (cuc_func *f, int b)
- void [detect_bb](#) (cuc_func *f)
- void [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)

5.67.1 Function Documentation

5.67.1.1 void build_bb (cuc_func * f)

Here is the call graph for this function:



5.67.1.2 void count_bb_seq (cuc_func * f, int b, char * bb_filename, int * counts, int preroll, int unroll)

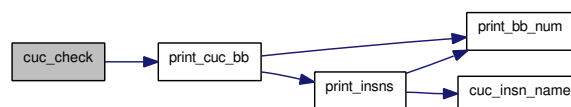
5.67.1.3 void cpy_bb (cuc_bb * dest, cuc_bb * src)

Here is the call graph for this function:



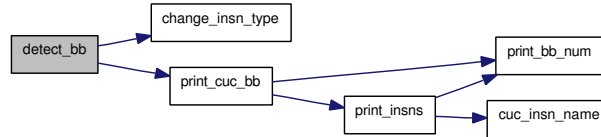
5.67.1.4 void cuc_check (cuc_func * f)

Here is the call graph for this function:



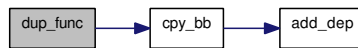
5.67.1.5 void detect_bb (cuc_func *f)

Here is the call graph for this function:



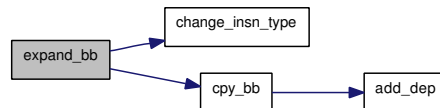
5.67.1.6 cuc_func* dup_func (cuc_func *f)

Here is the call graph for this function:



5.67.1.7 void expand_bb (cuc_func *f, int b)

Here is the call graph for this function:



5.67.1.8 void free_func (cuc_func *f)

Here is the call graph for this function:



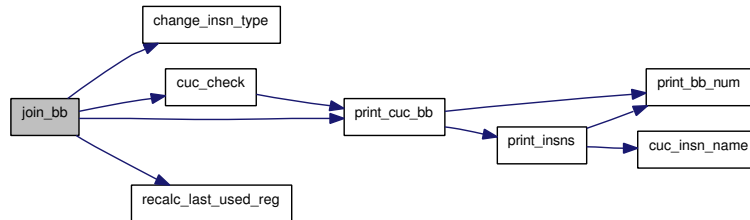
5.67.1.9 void generate_bb_seq (cuc_func *f, char *mp_filename, char *bb_filename)

Here is the call graph for this function:



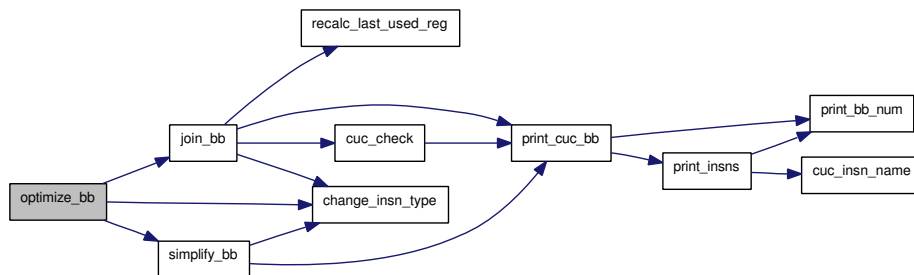
5.67.1.10 static void join_bb (cuc_func *f, int pred, int succ, int type) [static]

Here is the call graph for this function:



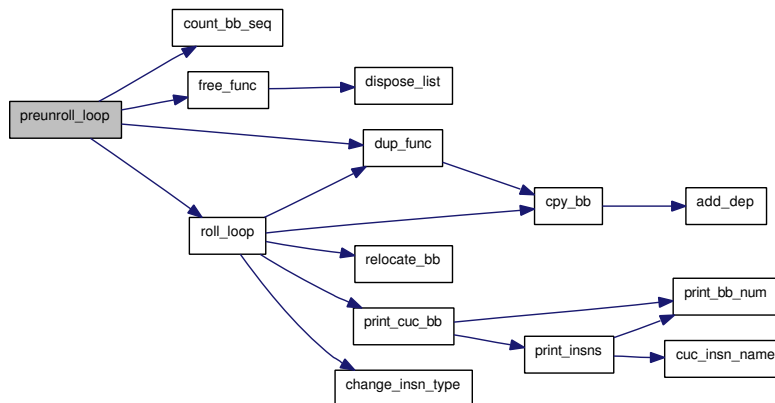
5.67.1.11 int optimize_bb (cuc_func *f)

Here is the call graph for this function:



5.67.1.12 cuc_func* preunroll_loop (cuc_func *f, int b, int preroll, int unroll, char *bb_filename)

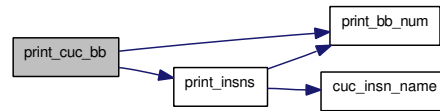
Here is the call graph for this function:



5.67.1.13 void print_bb_num (int num)

5.67.1.14 void print_cuc_bb (cuc_func *f, char *s)

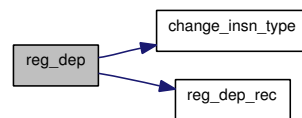
Here is the call graph for this function:



5.67.1.15 void recalc_last_used_reg (cuc_func *f, int b)

5.67.1.16 void reg_dep (cuc_func *f)

Here is the call graph for this function:



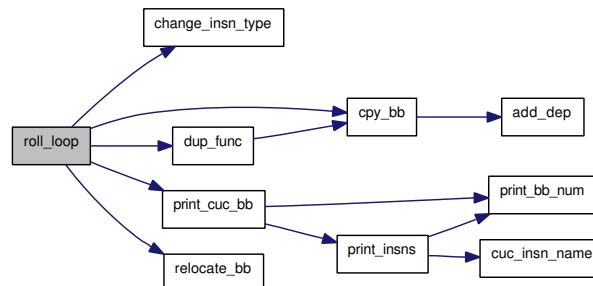
5.67.1.17 static void reg_dep_rec (cuc_func *f, int cur) [static]

5.67.1.18 static void relocate_bb (cuc_bb *bb, int b, int back, int fwd) [static]

5.67.1.19 int remove_dead_bb (cuc_func *f)

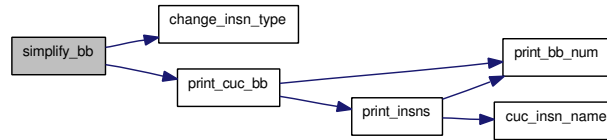
5.67.1.20 static cuc_func* roll_loop (cuc_func *f, int b, int ntimes, int type) [static]

Here is the call graph for this function:



5.67.1.21 `static void simplify_bb (cuc_func *f, int pred, int s1, int s2, int neg)` [static]

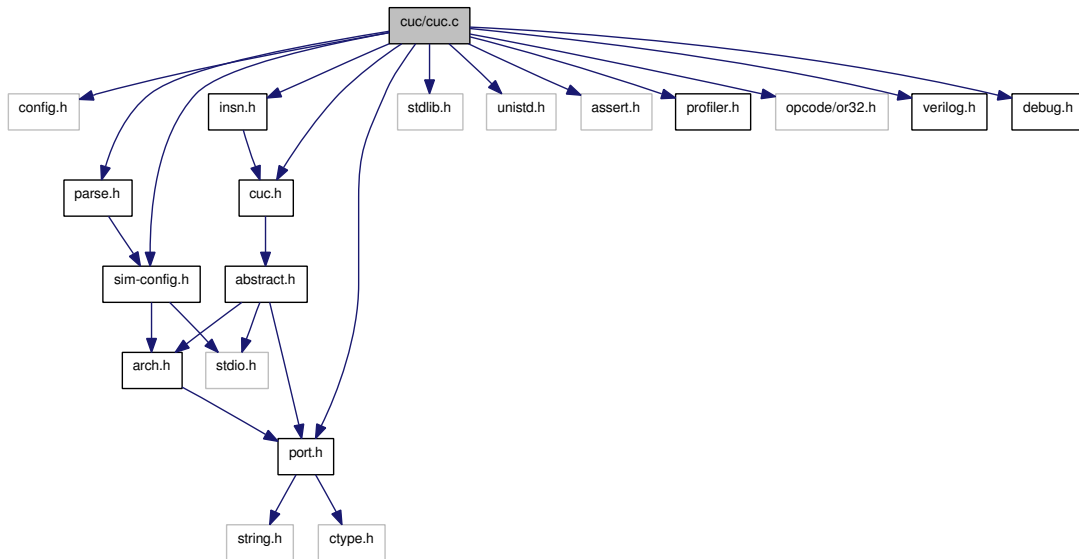
Here is the call graph for this function:



5.68 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](#) ()

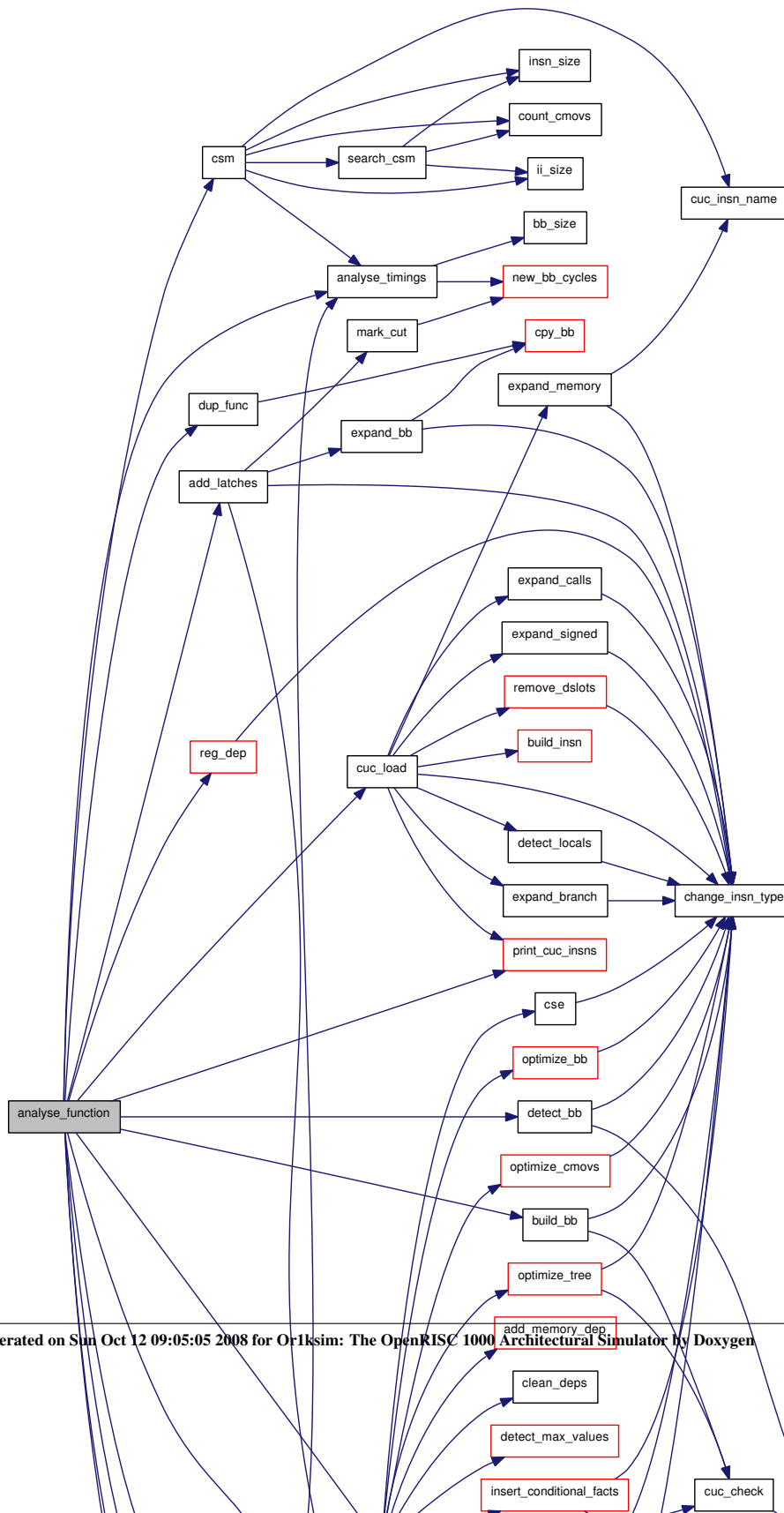
Variables

- 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]

5.68.1 Function Documentation

5.68.1.1 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:



5.68.1.2 `int calc_cycles (cuc_func *f)`

5.68.1.3 `double calc_size (cuc_func *f)`

5.68.1.4 `static void cuc_calling_conv (union param_val val, void *dat)` [static]

5.68.1.5 `static void cuc_enable_bursts (union param_val val, void *dat)` [static]

5.68.1.6 `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:

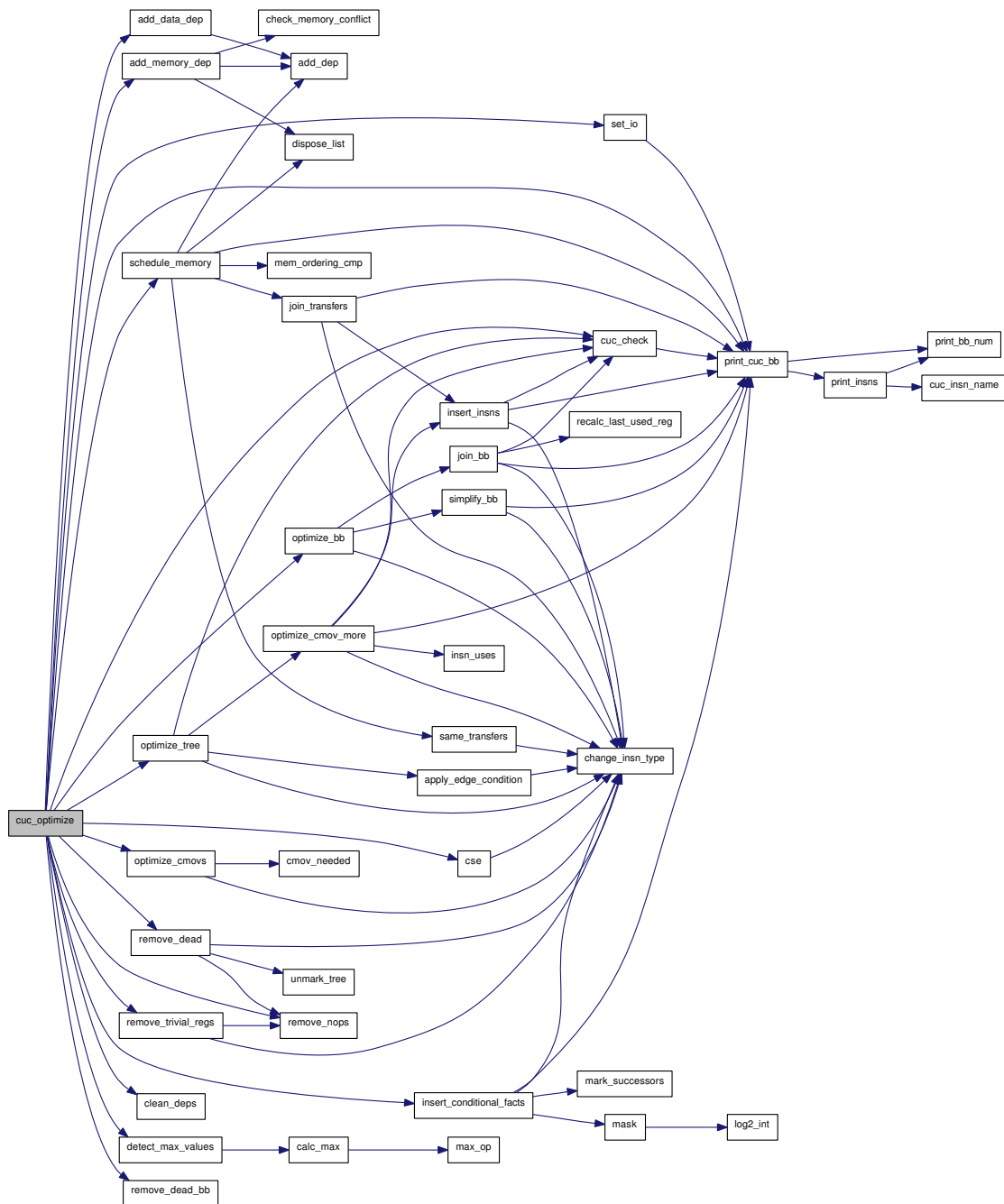
← *val* The value to use

← *dat* The [config](#) data structure (not used here)

5.68.1.7 `static void cuc_no_multicycle (union param_val val, void * dat) [static]`

5.68.1.8 `void cuc_optimize (cuc_func * func)`

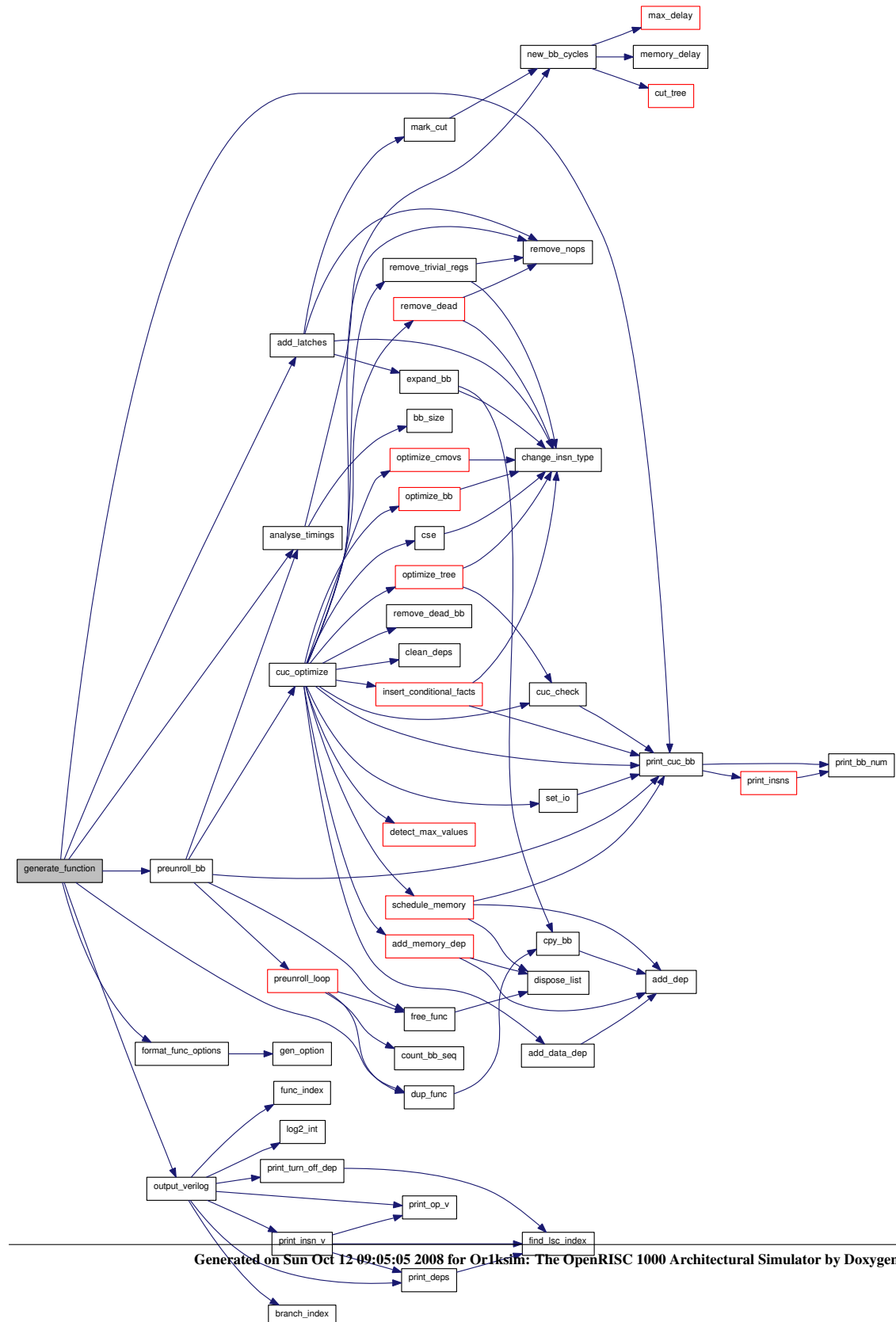
Here is the call graph for this function:



5.68.1.12 `char* gen_option (char * s, int bb_no, int f_opt)`

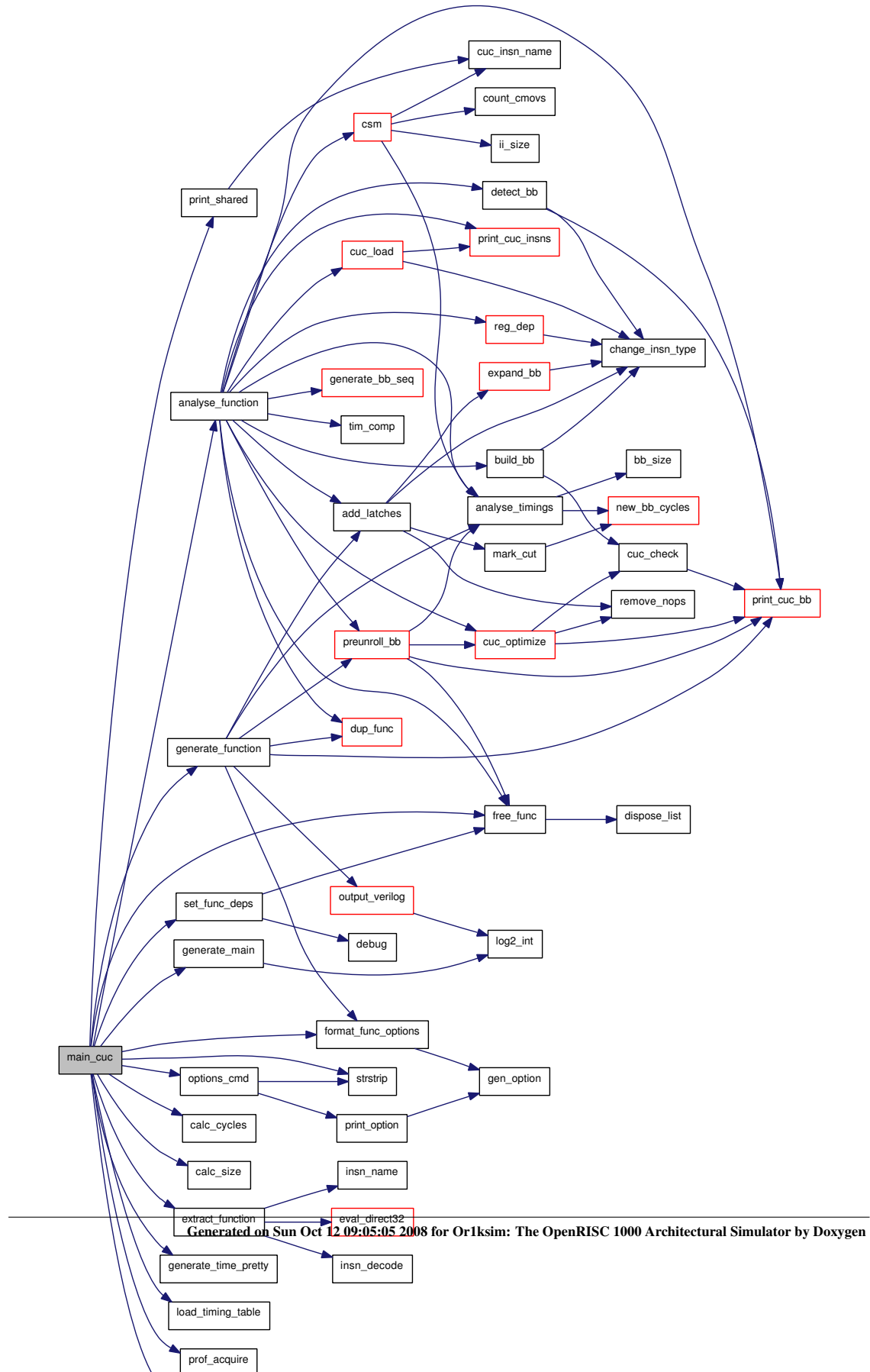
5.68.1.13 `cuc_func* generate_function (cuc_func * rf, char * name, char * cut_filename)`

Here is the call graph for this function:



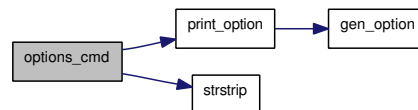
5.68.1.14 void main_cuc (char * filename)

Here is the call graph for this function:



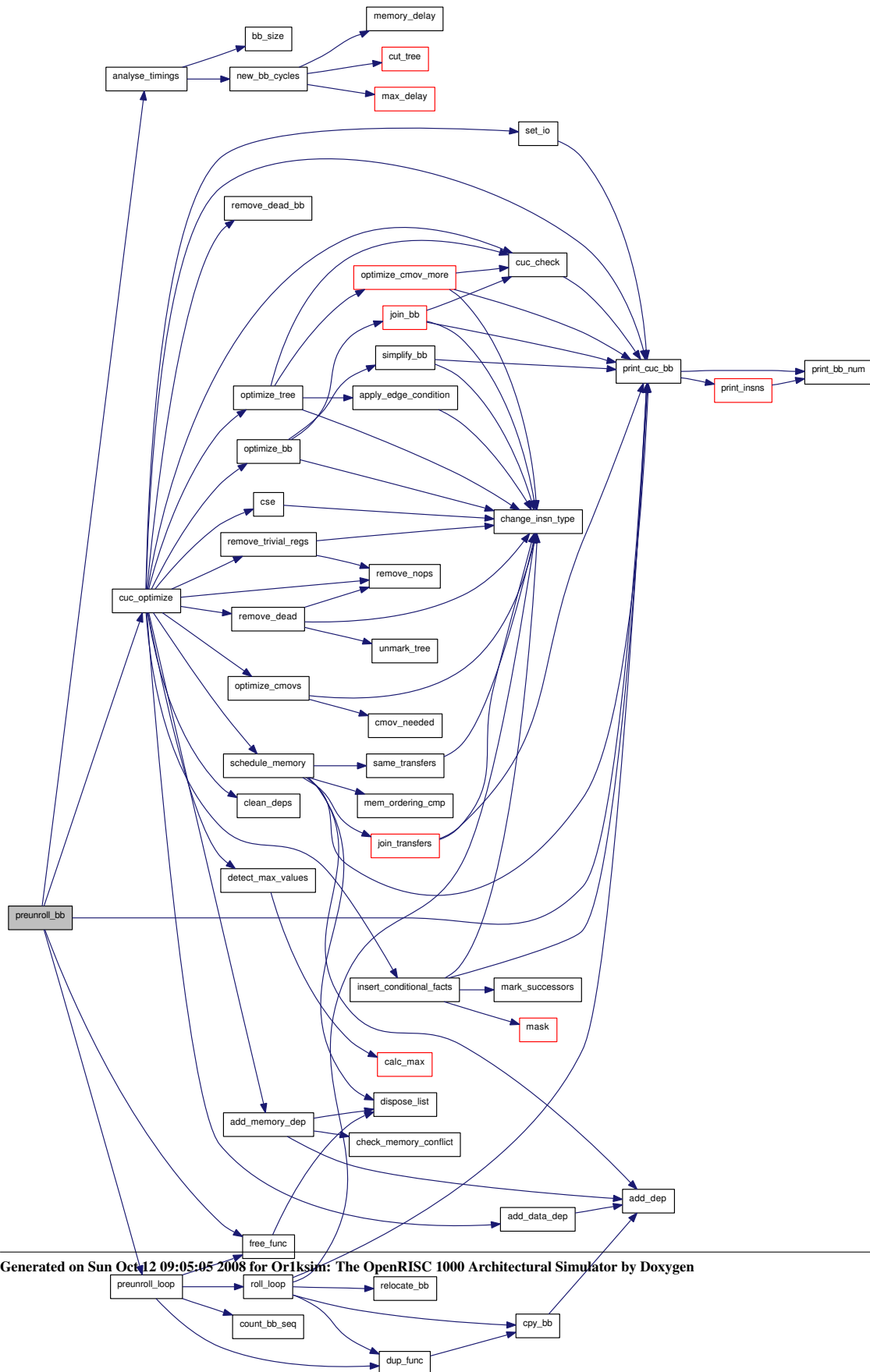
5.68.1.15 `static void options_cmd (int func_no, cuc_func *f)` [static]

Here is the call graph for this function:



5.68.1.16 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:



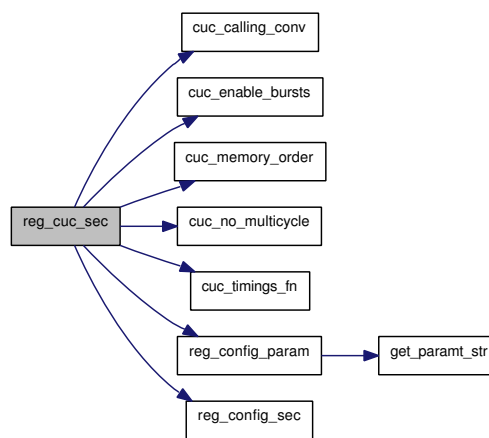
5.68.1.17 void print_option (int bb_no, int f_opt)

Here is the call graph for this function:



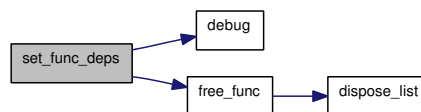
5.68.1.18 void reg_cuc_sec ()

Here is the call graph for this function:



5.68.1.19 static void set_func_deps () [static]

Here is the call graph for this function:



5.68.1.20 int tim_comp (cuc_timings * a, cuc_timings * b)

5.68.2 Variable Documentation

5.68.2.1 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  
}
```

5.68.2.2 `int cuc_debug = 0`

5.68.2.3 `FILE* flog`

5.68.2.4 `cuc_func* func[MAX_FUNCS]` [static]

5.68.2.5 `int func_v[MAX_FUNCS]` [static]

5.68.2.6 `const char* option_char` [static]

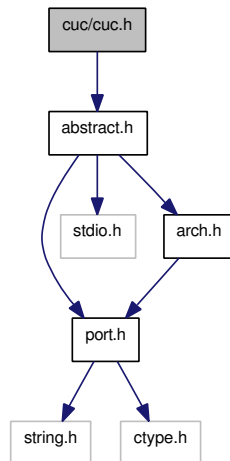
Initial value:

```
"?abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

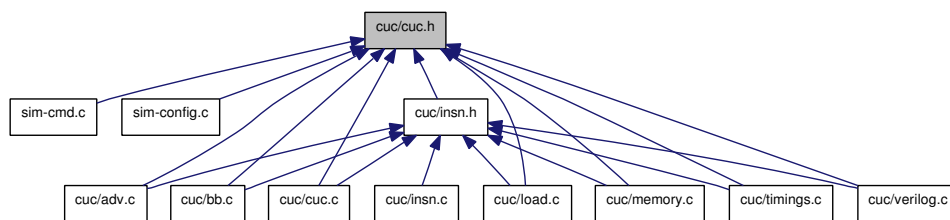
5.69 cuc/cuc.h File Reference

```
#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 *insns, 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](#)

5.69.1 Define Documentation

5.69.1.1 `#define BB_DEAD 0x08`

5.69.1.2 `#define BB_INLOOP 0x01`

5.69.1.3 `#define BB_OPTIONAL 0x02`

5.69.1.4 `#define BBID_END (MAX_BB + 1)`

5.69.1.5 `#define BBID_START MAX_BB`

5.69.1.6 `#define CUC_MAX_STACK 0x1000`

5.69.1.7 `#define CUC_WIDTH_ITERATIONS 256`

5.69.1.8 `#define cucdebug(x, s...) {if ((x) <= cuc_debug) PRINTF (s);}`

5.69.1.9 `#define FLAG_REG (MAX_REGS - 2)`

5.69.1.10 `#define INSN(ref) bb[REF_BB(ref)].insn[REF_I(ref)]`

5.69.1.11 `#define IT_BBEND 0x0008`

5.69.1.12 `#define IT_BBSTART 0x0004`

5.69.1.13 `#define IT_BRANCH 0x0001`

5.69.1.14 `#define IT_COND 0x1000`

5.69.1.15 `#define IT_CUT 0x4000`

5.69.1.16 `#define IT_FLAG1 0x0100`

5.69.1.17 `#define IT_FLAG2 0x0200`

5.69.1.18 `#define IT_INDELAY 0x0002`

5.69.1.19 `#define IT_LATCHED 0x2000`

5.69.1.20 `#define IT_MEMADD 0x0800`

5.69.1.21 `#define IT_MEMORY 0x0040`

5.69.1.22 `#define IT_OUTPUT 0x0010`

5.69.1.23 `#define IT_SIGNED 0x0020`

5.69.1.24 `#define IT_UNUSED 0x0080`

5.69.1.25 `#define IT_VOLATILE 0x0400`

5.69.1.26 `#define log(x...) {fprintf (flog, x); fflush (flog); }`

5.69.1.27 `#define LRB_REG (MAX_REGS - 1)`
Generated on Sun Oct 12 09:46:53 2008 for OpenMim: The OpenMIPS 1000 Architectural Simulator by Doxygen

5.69.1.28 `#define MAX(x, y) ((x) > (y) ? (x) : (y))`

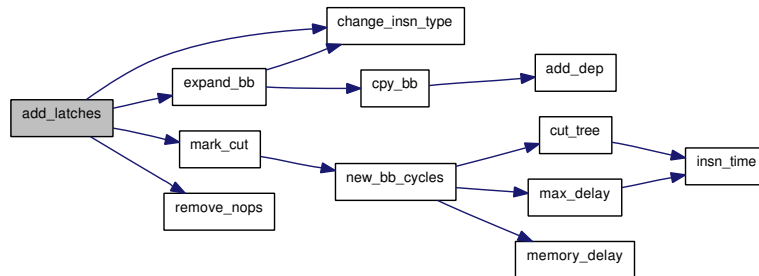
5.69.1.29 `#define MAX_BB 0x1000`

5.69.1.30 `#define MAX_INSN 0x10000`

5.69.3.2 void add_dep (dep_list ** list, int dep)

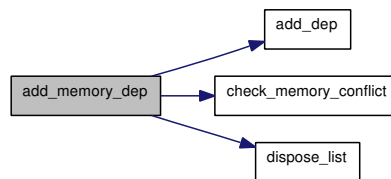
5.69.3.3 void add_latches (cuc_func * f)

Here is the call graph for this function:



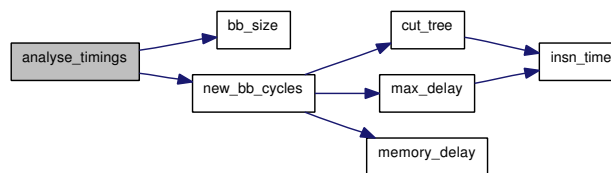
5.69.3.4 void add_memory_dep (cuc_func * f, int otype)

Here is the call graph for this function:



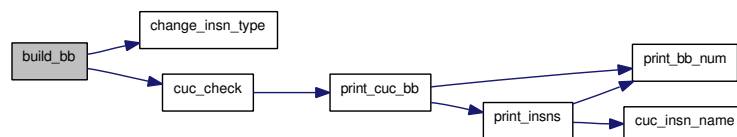
5.69.3.5 void analyse_timings (cuc_func * func, cuc_timings * timings)

Here is the call graph for this function:



5.69.3.6 void build_bb (cuc_func * f)

Here is the call graph for this function:

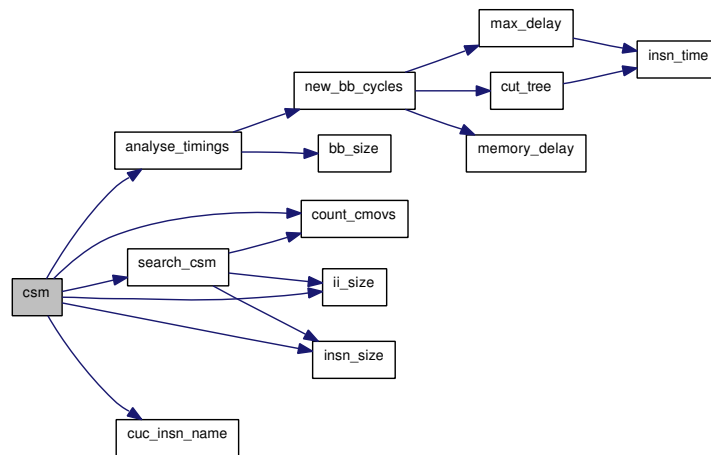


5.69.3.7 void clean_deps (cuc_func *func)**5.69.3.8 int cse (cuc_func *f)**

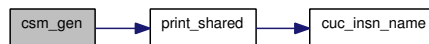
Here is the call graph for this function:

**5.69.3.9 void csm (cuc_func *f)**

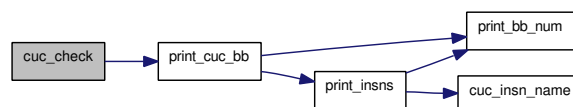
Here is the call graph for this function:

**5.69.3.10 void csm_gen (cuc_func *f, cuc_func *rf, cuc_shared_item *shared, int nshared)**

Here is the call graph for this function:

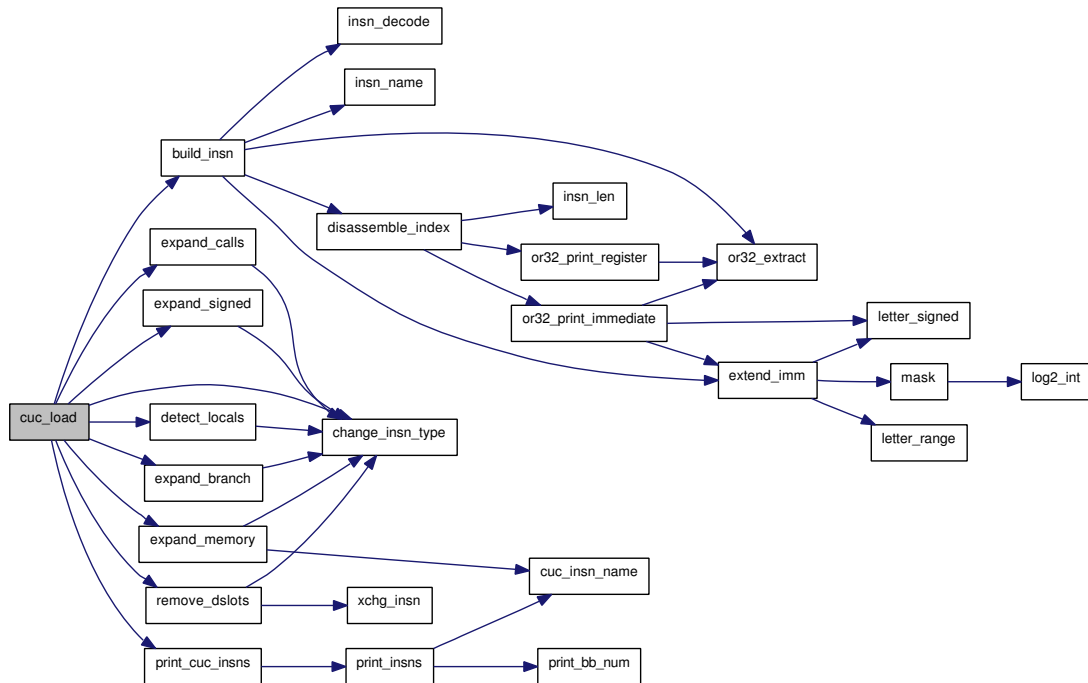
**5.69.3.11 void cuc_check (cuc_func *f)**

Here is the call graph for this function:



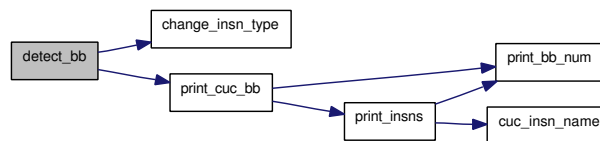
5.69.3.12 int cuc_load (char * in_fn)

Here is the call graph for this function:



5.69.3.13 void detect_bb (cuc_func * func)

Here is the call graph for this function:



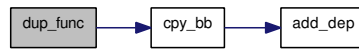
5.69.3.14 void detect_max_values (cuc_func * f)

Here is the call graph for this function:

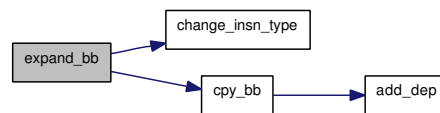


5.69.3.15 void dispose_list (dep_list ** list)**5.69.3.16 cuc_func* dup_func (cuc_func * f)**

Here is the call graph for this function:

**5.69.3.17 void expand_bb (cuc_func * f, int b)**

Here is the call graph for this function:

**5.69.3.18 void free_func (cuc_func * f)**

Here is the call graph for this function:

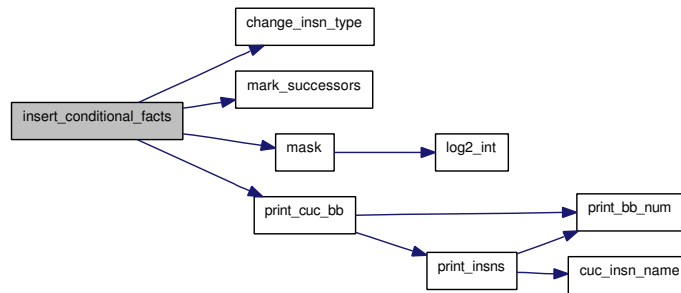
**5.69.3.19 void generate_bb_seq (cuc_func * f, char * mp_filename, char * bb_filename)**

Here is the call graph for this function:



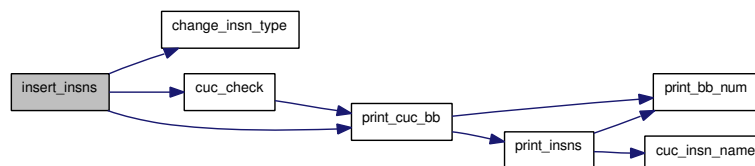
5.69.3.20 void insert_conditional_facts (cuc_func * func)

Here is the call graph for this function:



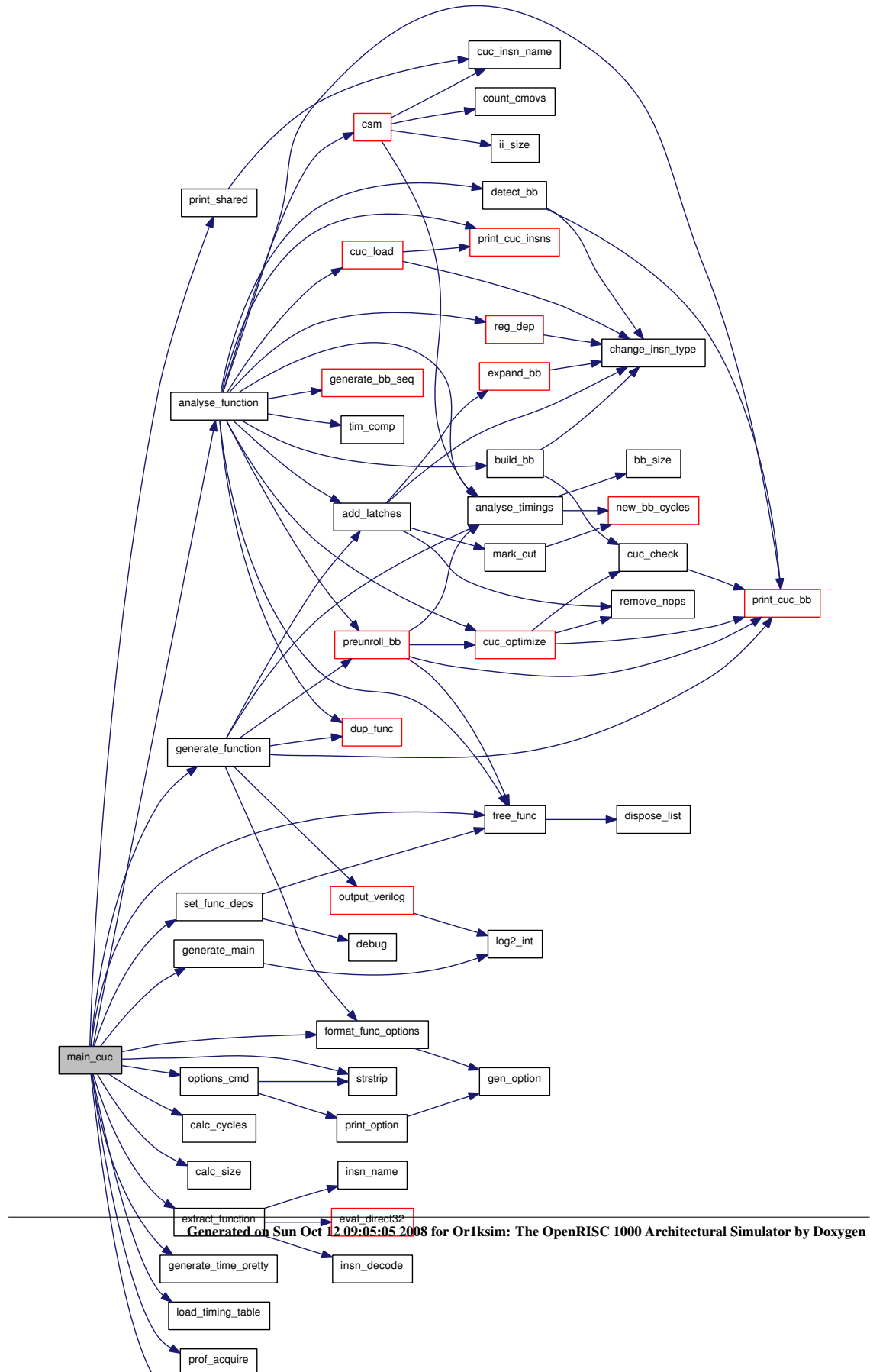
5.69.3.21 void insert_insns (cuc_func * f, int ref, int n)

Here is the call graph for this function:



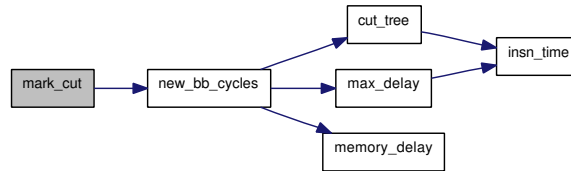
5.69.3.22 void main_cuc (char * filename)

Here is the call graph for this function:



5.69.3.23 void mark_cut (cuc_func * f)

Here is the call graph for this function:



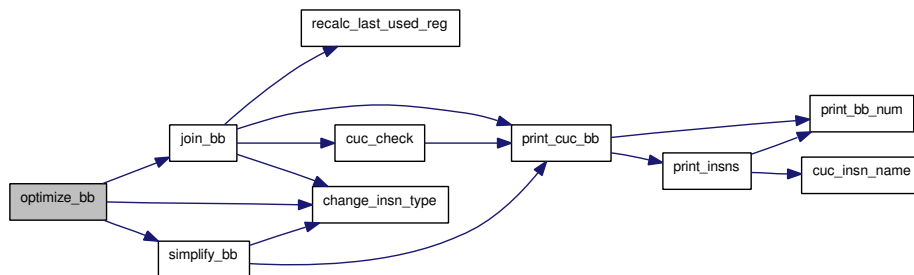
5.69.3.24 void negate_conditional (cuc_insn * ii)

Here is the call graph for this function:



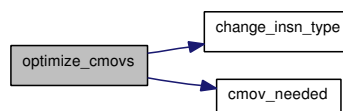
5.69.3.25 int optimize_bb (cuc_func * func)

Here is the call graph for this function:



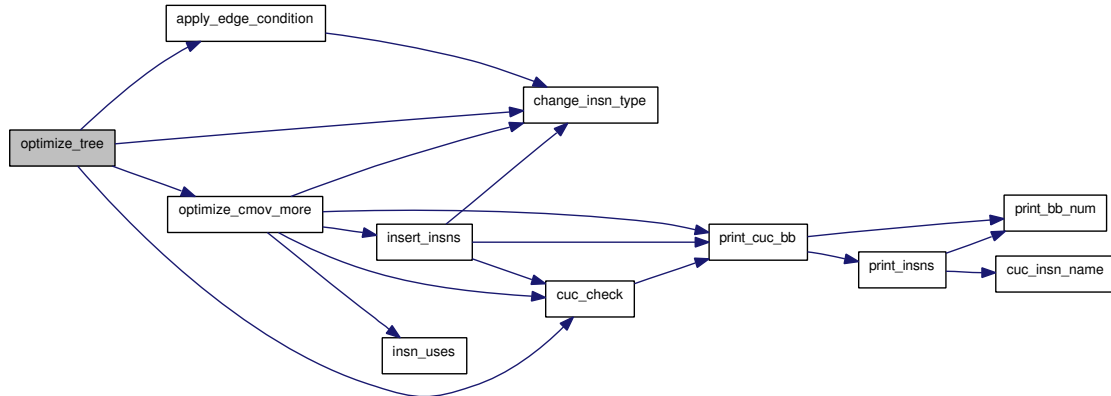
5.69.3.26 int optimize_cmovs (cuc_func * func)

Here is the call graph for this function:



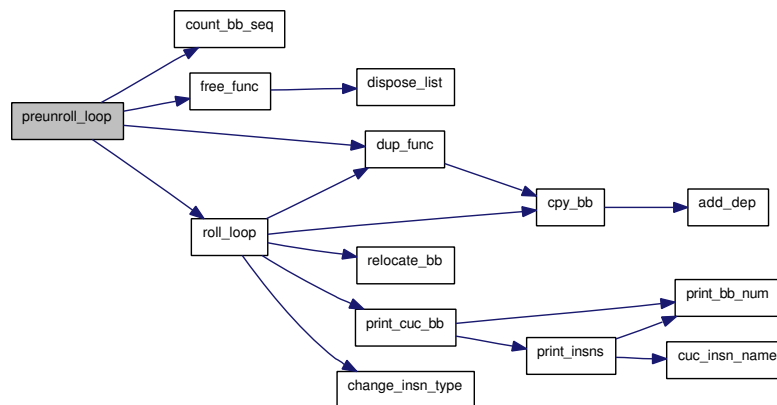
5.69.3.27 int optimize_tree (cuc_func * func)

Here is the call graph for this function:



5.69.3.28 cuc_func* preunroll_loop (cuc_func * func, int b, int preroll, int unroll, char * bb_filename)

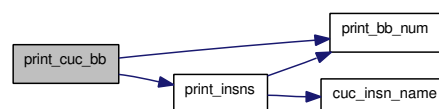
Here is the call graph for this function:



5.69.3.29 void print_bb_num (int num)

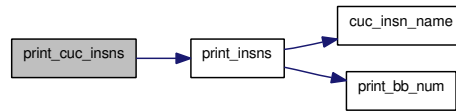
5.69.3.30 void print_cuc_bb (cuc_func * func, char * s)

Here is the call graph for this function:

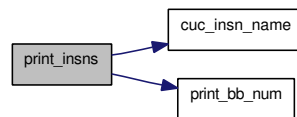


5.69.3.31 void print_cuc_insns (char * s, int verbose)

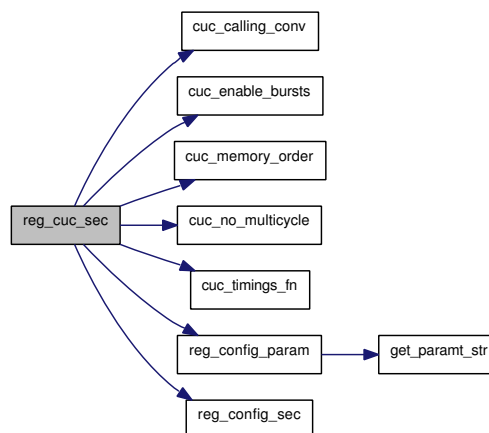
Here is the call graph for this function:

**5.69.3.32 void print_insns (int bb, cuc_insn * insn, int size, int verbose)**

Here is the call graph for this function:

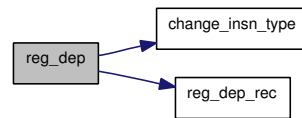
**5.69.3.33 void recalc_cnts (cuc_func * f, char * bb_filename)****5.69.3.34 void reg_cuc_sec ()**

Here is the call graph for this function:

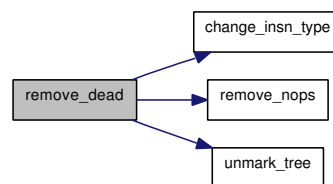


5.69.3.35 void reg_dep (cuc_func *func)

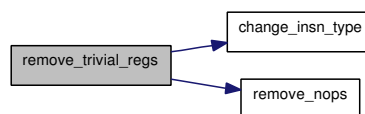
Here is the call graph for this function:

**5.69.3.36 int remove_dead (cuc_func *func)**

Here is the call graph for this function:

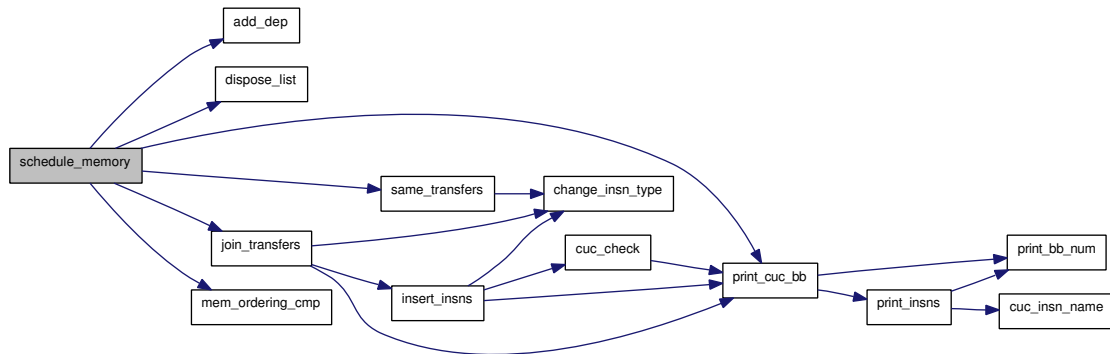
**5.69.3.37 int remove_dead_bb (cuc_func *func)****5.69.3.38 int remove_nops (cuc_func *func)****5.69.3.39 int remove_trivial_regs (cuc_func *f)**

Here is the call graph for this function:



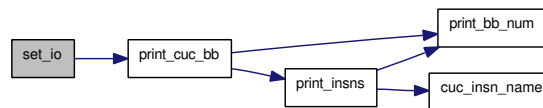
5.69.3.40 int schedule_memory (cuc_func *func, int otype)

Here is the call graph for this function:



5.69.3.41 void set_io (cuc_func *func)

Here is the call graph for this function:



5.69.4 Variable Documentation

5.69.4.1 const int caller_saved[MAX_REGS]

5.69.4.2 int cuc_debug

5.69.4.3 FILE* flog

5.69.4.4 cuc_insn insn[MAX_INSNS]

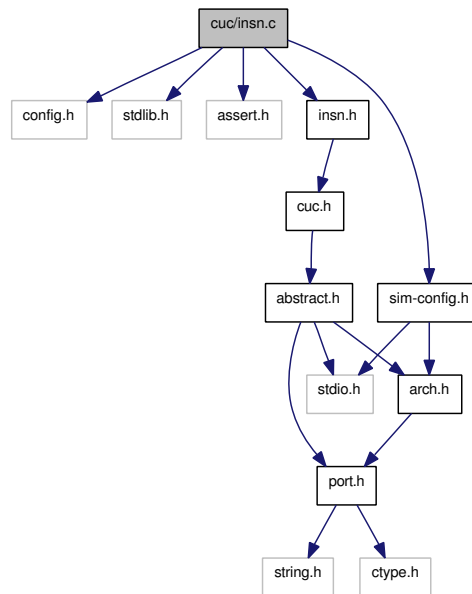
5.69.4.5 int num_insn

5.69.4.6 int reloc[MAX_INSNS]

5.70 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`

5.70.1 Function Documentation

5.70.1.1 void `add_data_dep` (`cuc_func` *f)

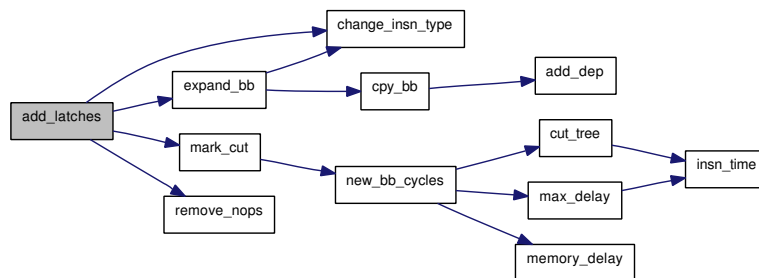
Here is the call graph for this function:



5.70.1.2 void `add_dep` (`dep_list` **list, int dep)

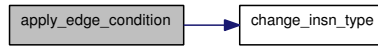
5.70.1.3 void `add_latches` (`cuc_func` *f)

Here is the call graph for this function:



5.70.1.4 int apply_edge_condition (cuc_insn * ii)

Here is the call graph for this function:



5.70.1.5 void change_insn_type (cuc_insn * i, int index)

5.70.1.6 static int cmov_needed (cuc_func * f, int ref) [static]

5.70.1.7 static int count_cmovs (cuc_insn * ii, int match) [static]

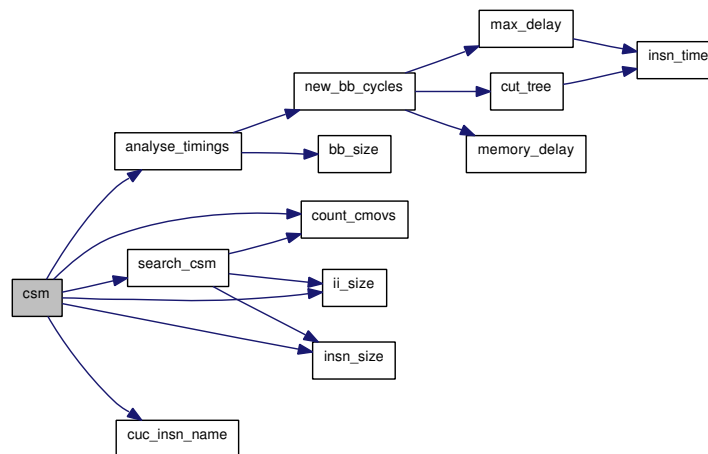
5.70.1.8 int cse (cuc_func * f)

Here is the call graph for this function:



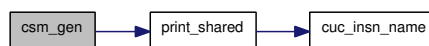
5.70.1.9 void csm (cuc_func * f)

Here is the call graph for this function:



5.70.1.10 void csm_gen (cuc_func * f, cuc_func * rf, cuc_shared_item * shared, int nshared)

Here is the call graph for this function:

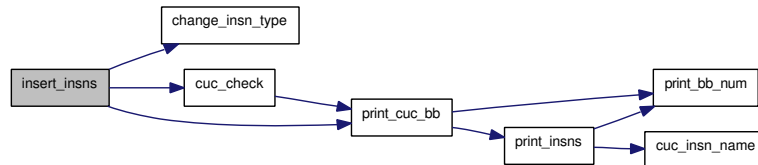


5.70.1.11 `const char* cuc_insn_name (cuc_insn * ii)`

5.70.1.12 `void dispose_list (dep_list ** list)`

5.70.1.13 `void insert_insns (cuc_func * f, int ref, int n)`

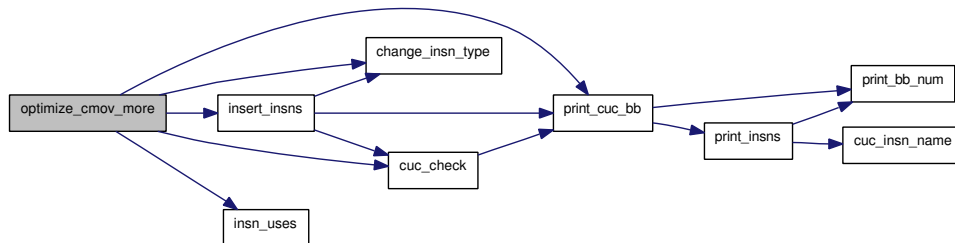
Here is the call graph for this function:



5.70.1.14 `static int insn_uses (cuc_func * f, int ref)` [static]

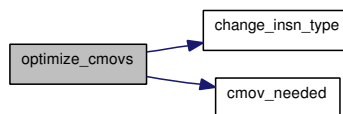
5.70.1.15 `static int optimize_cmov_more (cuc_func * f, int ref)` [static]

Here is the call graph for this function:



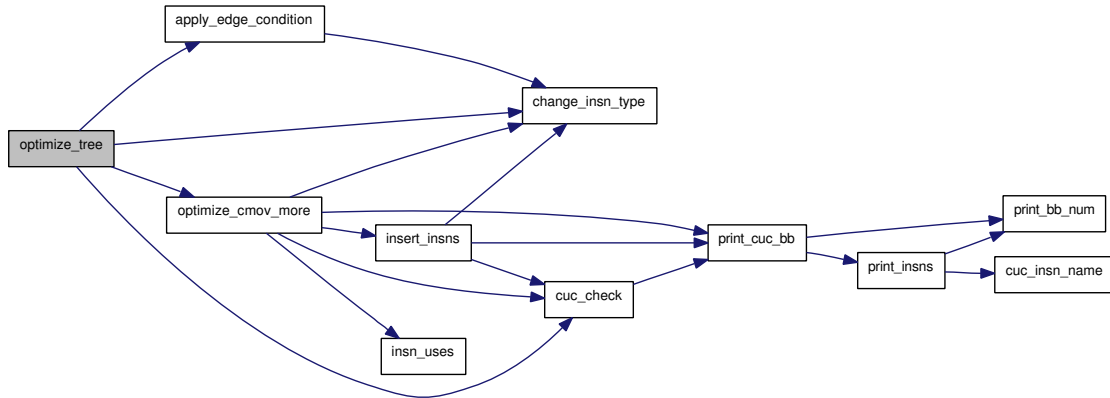
5.70.1.16 `int optimize_cmovs (cuc_func * f)`

Here is the call graph for this function:



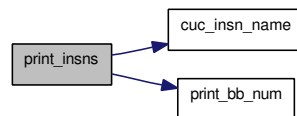
5.70.1.17 int optimize_tree (cuc_func * f)

Here is the call graph for this function:



5.70.1.18 void print_insn (int bb, cuc_insn * insn, int ninsn, int verbose)

Here is the call graph for this function:



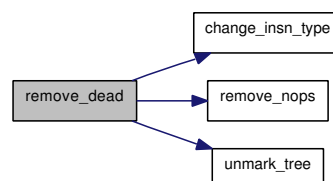
5.70.1.19 void print_shared (cuc_func * rf, cuc_shared_item * shared, int nshared)

Here is the call graph for this function:



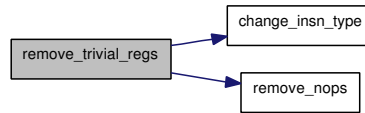
5.70.1.20 int remove_dead (cuc_func * f)

Here is the call graph for this function:

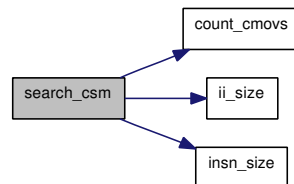


5.70.1.21 `int remove_nops (cuc_func *f)`**5.70.1.22** `int remove_trivial_regs (cuc_func *f)`

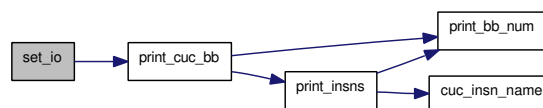
Here is the call graph for this function:

**5.70.1.23** `static void search_csm (int iter, cuc_func *f, cuc_shared_list *list)` [static]

Here is the call graph for this function:

**5.70.1.24** `void set_io (cuc_func *f)`

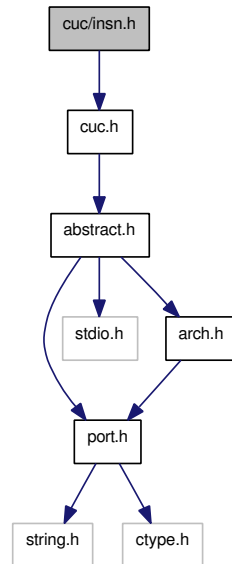
Here is the call graph for this function:

**5.70.1.25** `static void unmark_tree (cuc_func *f, int ref)` [static]**5.70.2** Variable Documentation**5.70.2.1** `int* iteration` [static]**5.70.2.2** `const cuc_known_insn known[II_LAST+1]`**5.70.2.3** `cuc_shared_list* main_list` [static]**5.70.2.4** `unsigned long tmp_op` [static]**5.70.2.5** `unsigned long tmp_opt` [static]

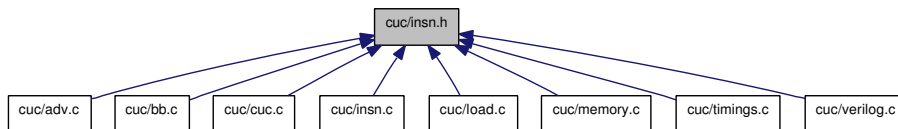
5.71 cuc/insn.h File Reference

```
#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]

5.71.1 Define Documentation

5.71.1.1 **#define II_ADD 0**

5.71.1.2 **#define II_AND 2**

5.71.1.3 **#define II_BF 21**

5.71.1.4 **#define II_CALL 26**

5.71.1.5 **#define II_CMOV 23**

5.71.1.6 **#define II_IS_LOAD(x) ((x) == II_LB || (x) == II_LH || (x) == II_LW)**

5.71.1.7 **#define II_IS_STORE(x) ((x) == II_SB || (x) == II_SH || (x) == II_SW)**

5.71.1.8 **#define II_LAST 26**

5.71.1.9 **#define II_LB 9**

5.71.1.10 **#define II_LH 10**

5.71.1.11 **#define II_LRBB 22**

5.71.1.12 **#define II_LW 11**

5.71.1.13 **#define II_MASK 0x0fff**

5.71.1.14 **#define II_MEM 0x1000**

5.71.1.15 **#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
```


5.71.1.16 `#define II_MUL 5`

5.71.1.17 `#define II_NOP 25`

5.71.1.18 `#define II_OR 3`

5.71.1.19 `#define II_REG 24`

5.71.1.20 `#define II_SB 12`

5.71.1.21 `#define II_SFEQ 15`

5.71.1.22 `#define II_SFGE 19`

5.71.1.23 `#define II_SFGT 20`

5.71.1.24 `#define II_SFLE 17`

5.71.1.25 `#define II_SFLT 18`

5.71.1.26 `#define II_SFNE 16`

5.71.1.27 `#define II_SH 13`

5.71.1.28 `#define II_SIGNED 0x2000`

5.71.1.29 `#define II_SLL 7`

5.71.1.30 `#define II_SRA 8`

5.71.1.31 `#define II_SRL 6`

5.71.1.32 `#define II_SUB 1`

5.71.1.33 `#define II_SW 14`

5.71.1.34 `#define II_XOR 4`

5.71.2 Function Documentation

5.71.2.1 `void change_insn_type (cuc_insn * i, int index)`

5.71.2.2 `const char* cuc_insn_name (cuc_insn * ii)`


5.71.2.3 `double ii_size (int index, int imm)`

5.71.2.4 `double insn_size (cuc_insn * ii)`

5.71.2.5 `double insn_time (cuc_insn * ii)`

5.71.2.6 `void load_timing_table (char * filename)`

5.71.2.7 `void print_shared (cuc_func * rf, cuc_shared_item * shared, int nshared)`

Here is the call graph for this function:  12 09:05:05 2008 for Or1ksim: The OpenRISC 1000 Architectural Simulator by Doxygen



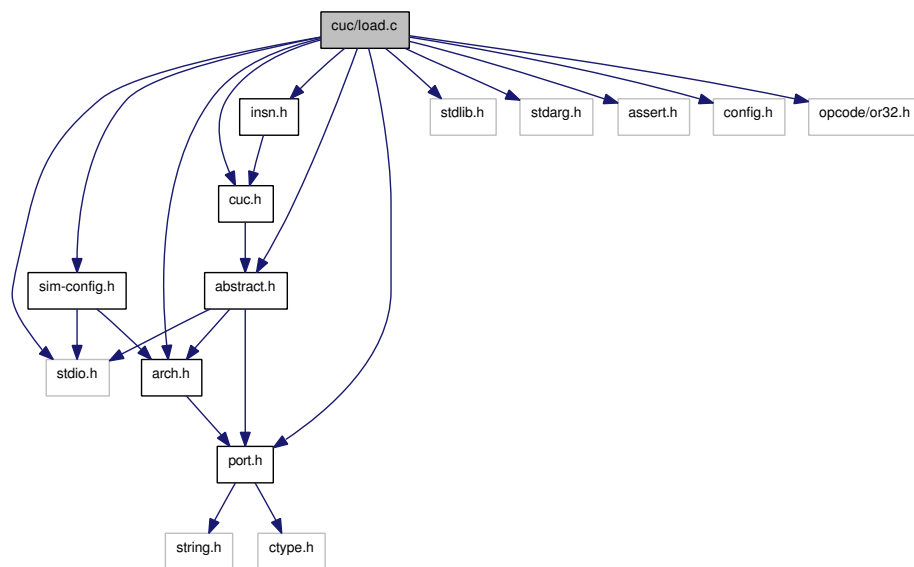
5.71.3 Variable Documentation

5.71.3.1 `const cuc_known_insn known[II_LAST+1]`

5.72 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)

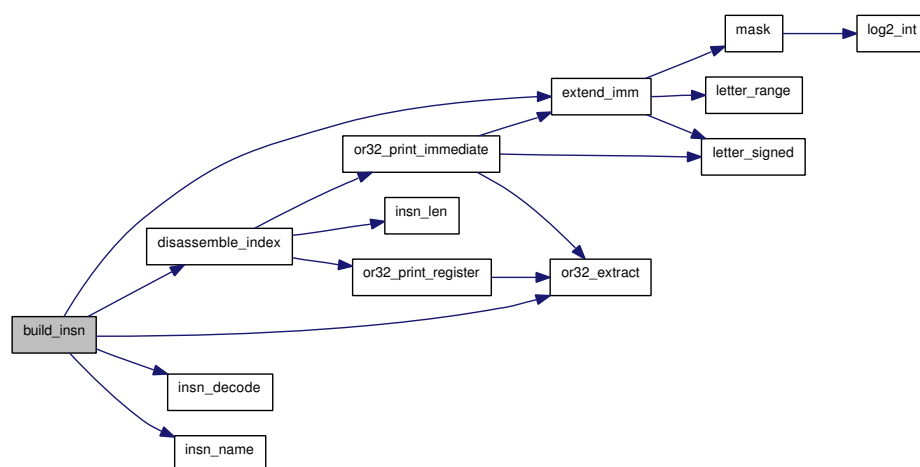
Variables

- static const [cuc_conv](#) conv []
- [cuc_insn](#) insn [MAX_INSNS]
- int [num_insn](#)
- int [reloc](#) [MAX_INSNS]

5.72.1 Function Documentation

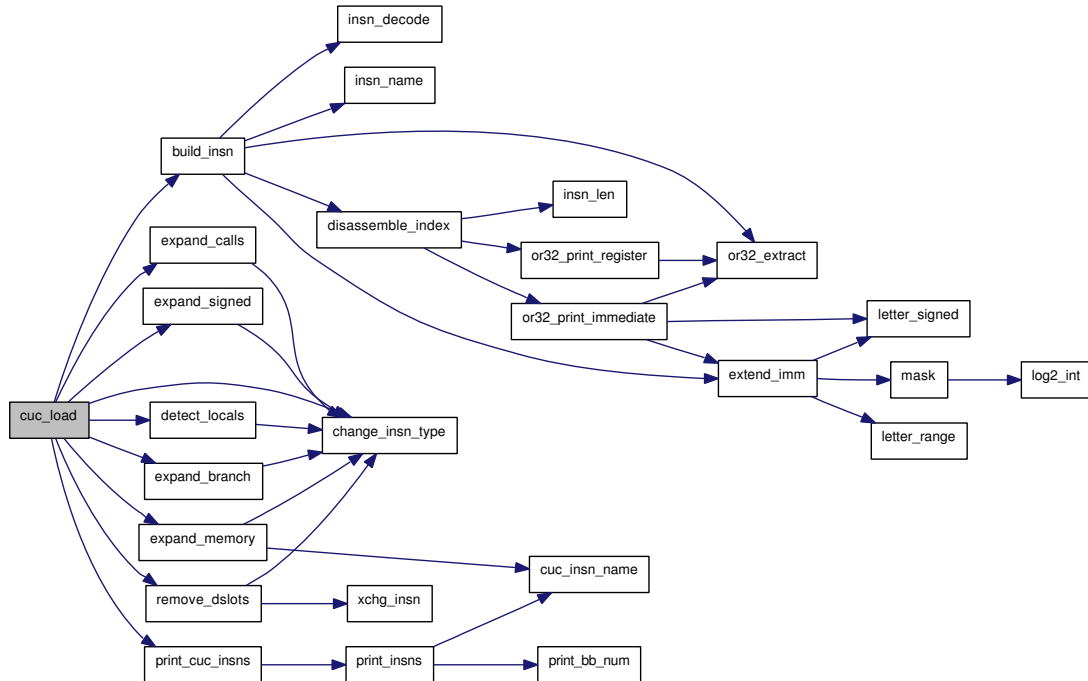
5.72.1.1 const char* [build_insn](#) (unsigned long *data*, cuc_insn * *insn*)

Here is the call graph for this function:



5.72.1.2 int cuc_load (char * in_fn)

Here is the call graph for this function:



5.72.1.3 void detect_locals ()

Here is the call graph for this function:



5.72.1.4 void expand_branch ()

Here is the call graph for this function:



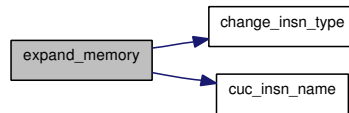
5.72.1.5 void expand_calls ()

Here is the call graph for this function:



5.72.1.6 void expand_memory ()

Here is the call graph for this function:



5.72.1.7 void expand_signed ()

Here is the call graph for this function:



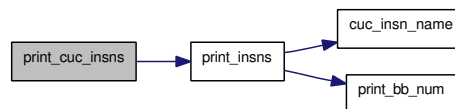
5.72.1.8 void negate_conditional (cuc_insn * ii)

Here is the call graph for this function:



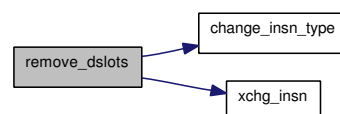
5.72.1.9 void print_cuc_insns (char * s, int verbose)

Here is the call graph for this function:



5.72.1.10 void remove_dslots ()

Here is the call graph for this function:



5.72.1.11 void xchg_insn (int *i*, int *j*)

5.72.2 Variable Documentation

5.72.2.1 const cuc_conv conv[] [static]

5.72.2.2 cuc_insn insn[MAX_INSNS]

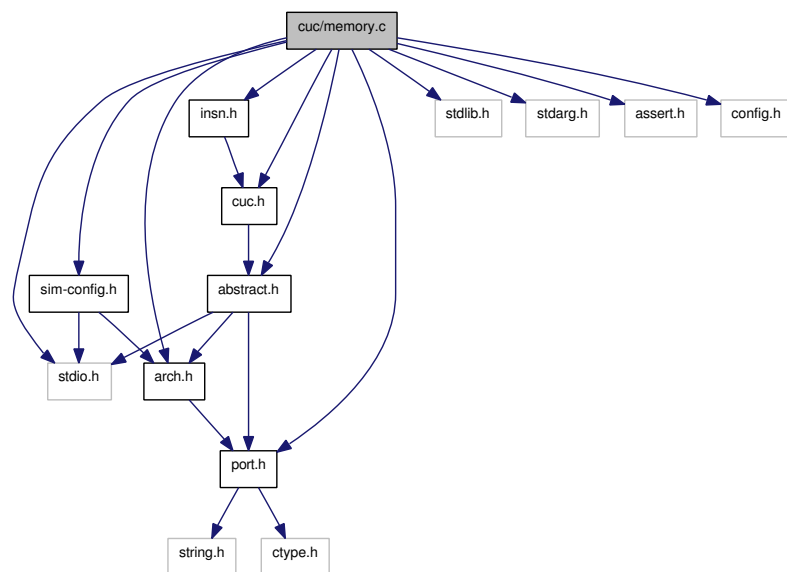
5.72.2.3 int num_insn

5.72.2.4 int reloc[MAX_INSNS]

5.73 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:



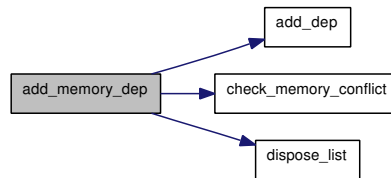
Functions

- 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)

5.73.1 Function Documentation

5.73.1.1 void add_memory_dep (cuc_func *f, int otype)

Here is the call graph for this function:

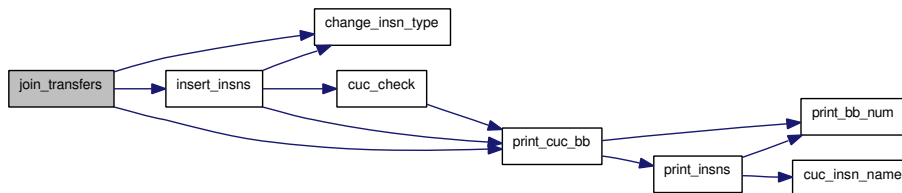


5.73.1.2 static int check_memory_conflict (cuc_func *f, cuc_insn *a, cuc_insn *b, int otype) [static]

5.73.1.3 void clean_deps (cuc_func *f)

5.73.1.4 static int join_transfers (cuc_func *f, int otype) [static]

Here is the call graph for this function:



5.73.1.5 int mem_ordering_cmp (cuc_func *f, cuc_insn *a, cuc_insn *b)

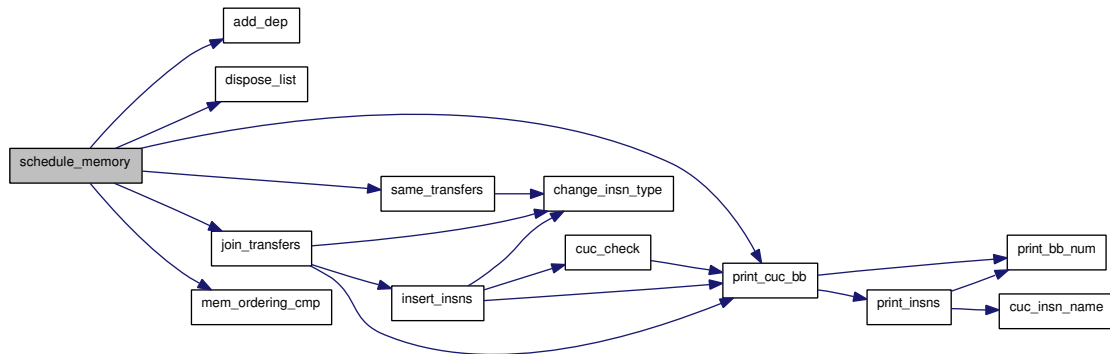
5.73.1.6 static int same_transfers (cuc_func *f, int otype) [static]

Here is the call graph for this function:



5.73.1.7 int schedule_memory (cuc_func *f, int otype)

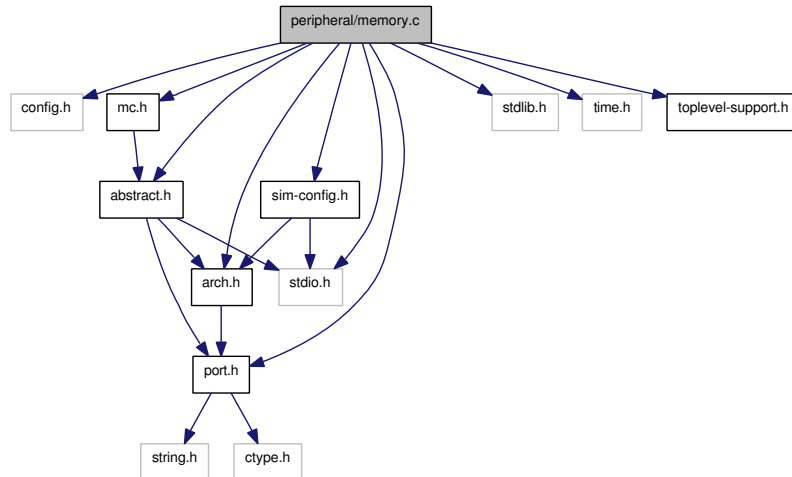
Here is the call graph for this function:



5.74 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`)

5.74.1 Function Documentation

5.74.1.1 static void `mem_reset` (`void * dat`) [static]

5.74.1.2 static void `memory_baseaddr` (`union param_val val`, `void * dat`) [static]

5.74.1.3 static void `memory_ce` (`union param_val val`, `void * dat`) [static]

5.74.1.4 static void `memory_delayr` (`union param_val val`, `void * dat`) [static]

5.74.1.5 static void `memory_delayw` (`union param_val val`, `void * dat`) [static]

5.74.1.6 static void `memory_log` (`union param_val val`, `void * dat`) [static]

5.74.1.7 static void `memory_mc` (`union param_val val`, `void * dat`) [static]

5.74.1.8 static void `memory_name` (`union param_val val`, `void * dat`) [static]

5.74.1.9 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:

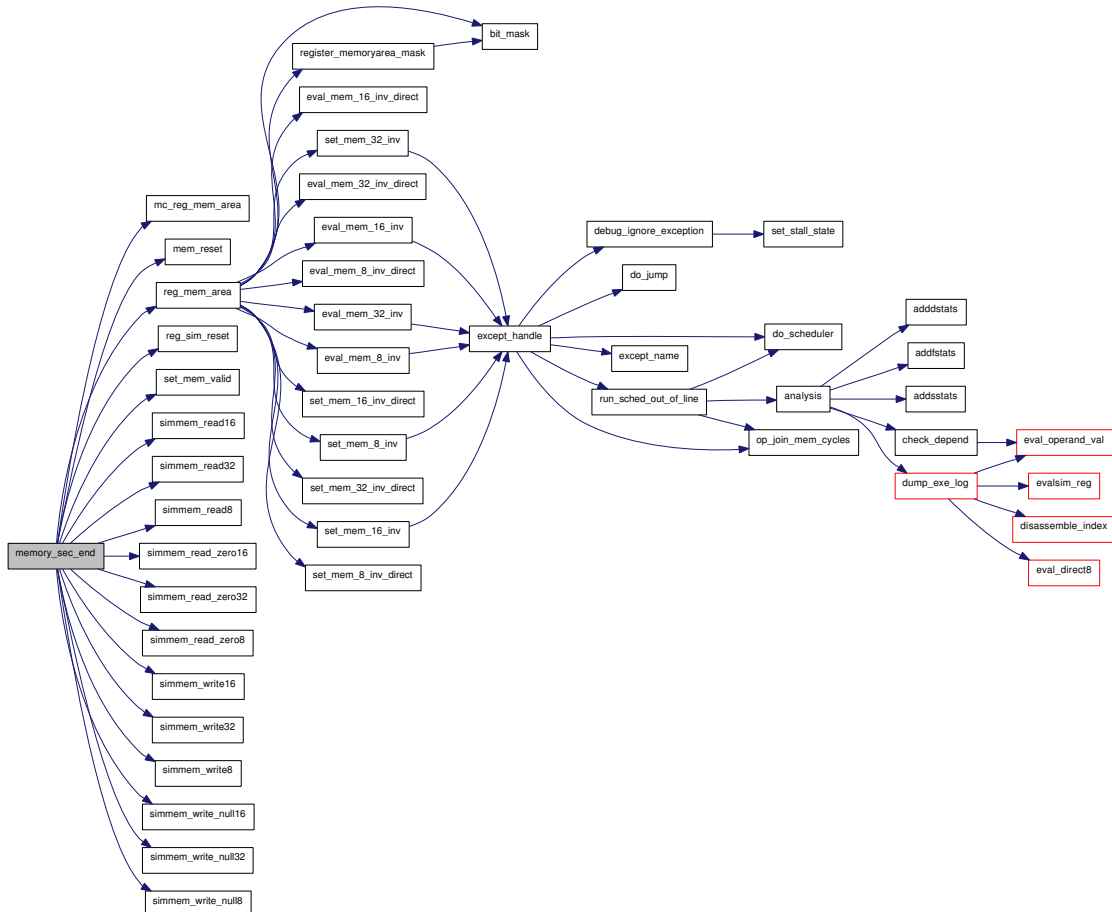
← *val* The value to use

← *dat* The `config` data structure

5.74.1.10 static void memory_random_seed (union param_val val, void * dat) [static]

5.74.1.11 static void memory_sec_end (void * dat) [static]

Here is the call graph for this function:



5.74.1.12 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

5.74.1.13 static void memory_size (union param_val val, void * dat) [static]

5.74.1.14 static void memory_type (union param_val val, void * dat) [static]

Set the memory type

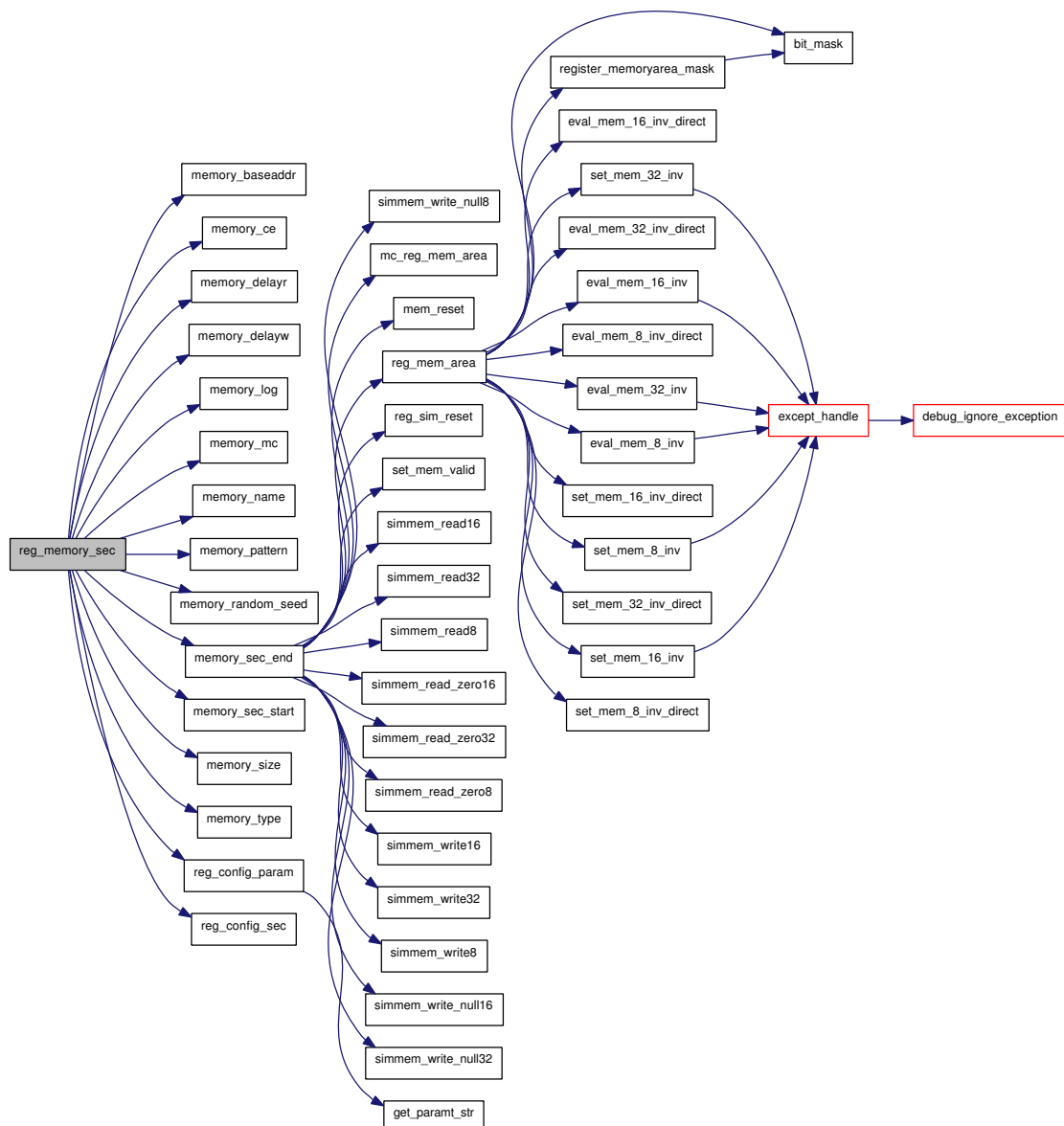
Value must be one of unknown, random, pattern or zero (case insensitive). Unrecognized values are ignored with a warning.

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure

5.74.1.15 void reg_memory_sec (void)

Here is the call graph for this function:

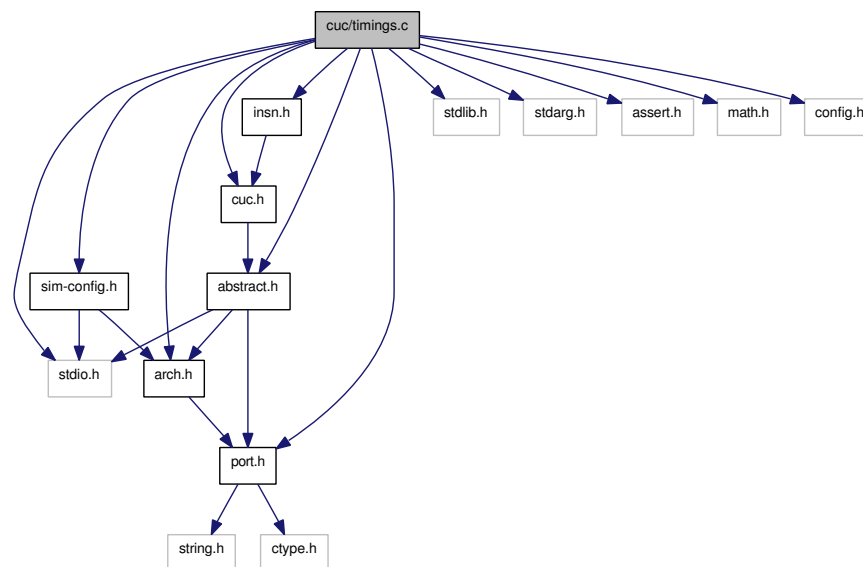


-
- 5.74.1.16 `static uint16_t simmem_read16 (oraddr_t addr, void * dat)` [static]
 - 5.74.1.17 `static uint32_t simmem_read32 (oraddr_t addr, void * dat)` [static]
 - 5.74.1.18 `static uint8_t simmem_read8 (oraddr_t addr, void * dat)` [static]
 - 5.74.1.19 `static uint16_t simmem_read_zero16 (oraddr_t addr, void * dat)` [static]
 - 5.74.1.20 `static uint32_t simmem_read_zero32 (oraddr_t addr, void * dat)` [static]
 - 5.74.1.21 `static uint8_t simmem_read_zero8 (oraddr_t addr, void * dat)` [static]
 - 5.74.1.22 `static void simmem_write16 (oraddr_t addr, uint16_t value, void * dat)` [static]
 - 5.74.1.23 `static void simmem_write32 (oraddr_t addr, uint32_t value, void * dat)` [static]
 - 5.74.1.24 `static void simmem_write8 (oraddr_t addr, uint8_t value, void * dat)` [static]
 - 5.74.1.25 `static void simmem_write_null16 (oraddr_t addr, uint16_t value, void * dat)`
[static]
 - 5.74.1.26 `static void simmem_write_null32 (oraddr_t addr, uint32_t value, void * dat)`
[static]
 - 5.74.1.27 `static void simmem_write_null8 (oraddr_t addr, uint8_t value, void * dat)` [static]

5.75 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`)

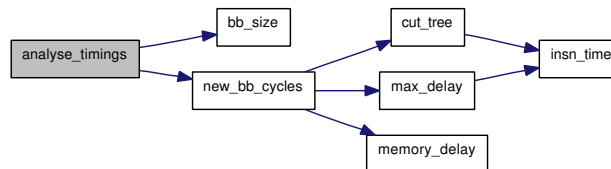
Variables

- static `cuc_timing_table * timing_table`
- static double `max_bb_delay`

5.75.1 Function Documentation

5.75.1.1 void `analyse_timings` (`cuc_func *f`, `cuc_timings *timings`)

Here is the call graph for this function:



5.75.1.2 static double `bb_size` (`cuc_bb *bb`) [static]

5.75.1.3 void `cut_tree` (`cuc_func *f`, `int b`, `double sd`)

Here is the call graph for this function:



5.75.1.4 `double ii_size (int index, int imm)`

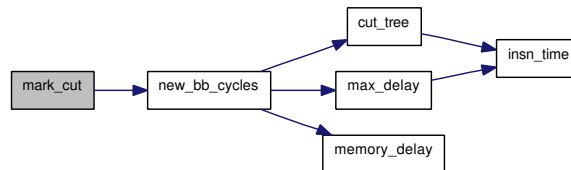
5.75.1.5 `double insn_size (cuc_insn * ii)`

5.75.1.6 `double insn_time (cuc_insn * ii)`

5.75.1.7 `void load_timing_table (char * filename)`

5.75.1.8 `void mark_cut (cuc_func * f)`

Here is the call graph for this function:



5.75.1.9 `static double max_delay (cuc_func * f, int b)` `[static]`

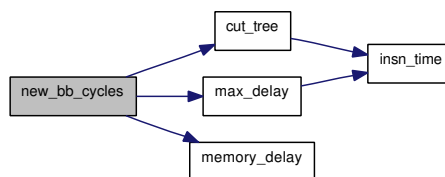
Here is the call graph for this function:



5.75.1.10 `static int memory_delay (cuc_func * f, int b)` `[static]`

5.75.1.11 `static int new_bb_cycles (cuc_func * f, int b, int cut)` `[static]`

Here is the call graph for this function:



5.75.1.12 `void recalc_cnts (cuc_func * f, char * bb_filename)`

5.75.2 Variable Documentation

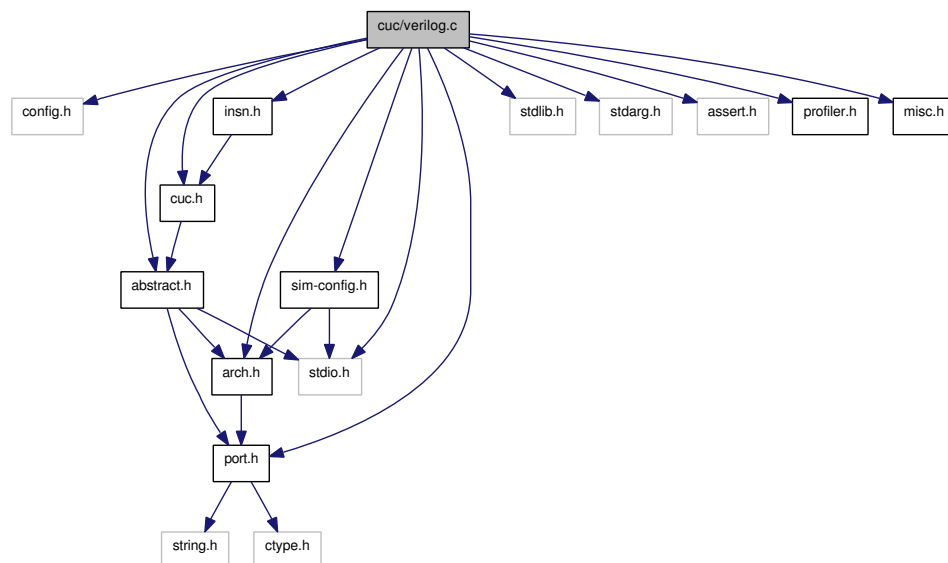
5.75.2.1 `double max_bb_delay` `[static]`

5.75.2.2 `cuc_timing_table* timing_table` `[static]`

5.76 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)

5.76.1 Define Documentation

5.76.1.1 `#define GEN(x...) fprintf (fo, x)`

5.76.2 Function Documentation

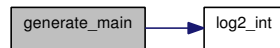
5.76.2.1 `static int branch_index (cuc_bb *bb)` [static]

5.76.2.2 `static int find_lsc_index (cuc_func *f, int ref)` [static]

5.76.2.3 `static int func_index (cuc_func *f, int ref)` [static]

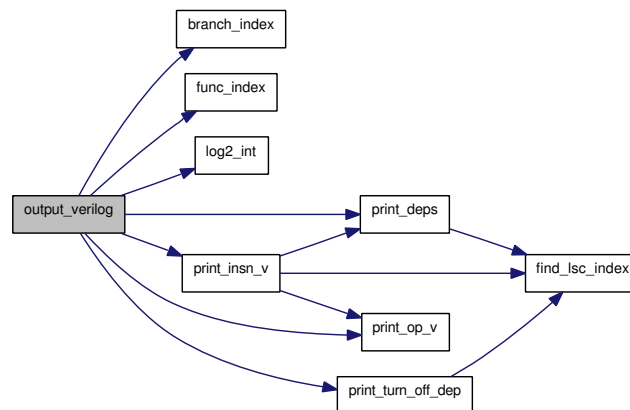
5.76.2.4 `void generate_main (int nfuncs, cuc_func **f, char *filename)`

Here is the call graph for this function:



5.76.2.5 `void output_verilog (cuc_func *f, char *filename, char *funcname)`

Here is the call graph for this function:

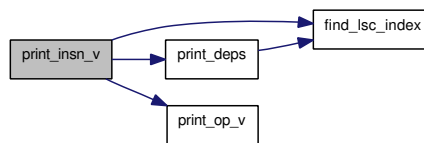


5.76.2.6 static void print_deps (FILE *fo, cuc_func *f, int b, dep_list *t, int registered)
[static]

Here is the call graph for this function:

**5.76.2.7 static void print_insn_v (FILE *fo, cuc_func *f, int b, int i)** [static]

Here is the call graph for this function:

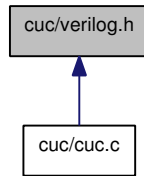
**5.76.2.8 static char* print_op_v (cuc_func *f, char *s, int ref, int j)** [static]**5.76.2.9 static void print_turn_off_dep (FILE *fo, cuc_func *f, dep_list *dep)** [static]

Here is the call graph for this function:



5.77 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)

5.77.1 Function Documentation

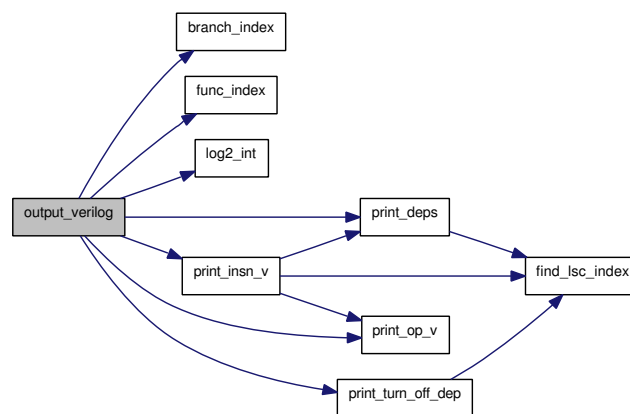
5.77.1.1 void generate_main (int nfuncs, cuc_func **f, char *filename)

Here is the call graph for this function:



5.77.1.2 void output_verilog (cuc_func *func, char *filename, char *funcname)

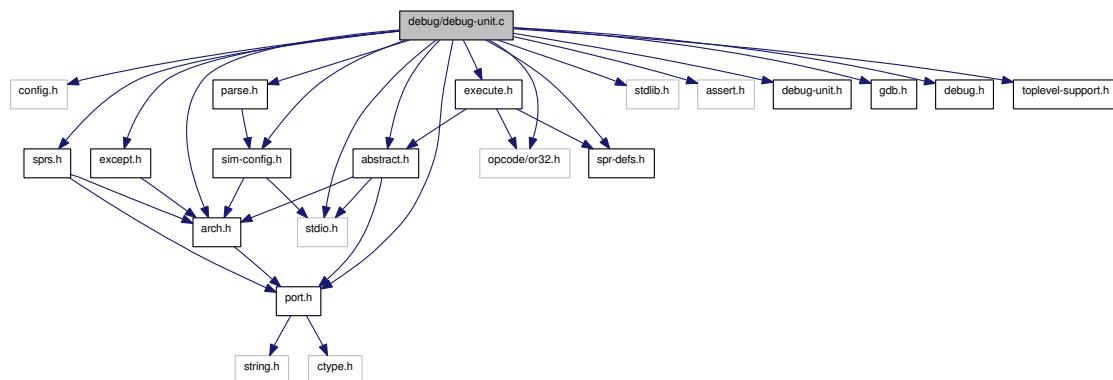
Here is the call graph for this function:



5.78 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 "debug.h"
#include "toplevel-support.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

- `DECLARE_DEBUG_CHANNEL` (jtag)
- 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)
- void `debug_enabled` (union `param_val` val, void *dat)
- void `debug_gdb_enabled` (union `param_val` val, void *dat)
- void `debug_server_port` (union `param_val` val, void *dat)
- 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

5.78.1 Define Documentation

5.78.1.1 #define RISCOP_RESET 0x00000002

Reset processor (clears stall)

5.78.1.2 #define RISCOP_STALL 0x00000001

The fields for the RISCOP register in the development interface scan chain (`JTAG_CHAIN_DEVELOPMENT`). Stall processor

5.78.2 Enumeration Type Documentation

5.78.2.1 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

5.78.3 Function Documentation

5.78.3.1 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

5.78.3.2 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:



5.78.3.8 void debug_server_port (union param_val val, void * dat)

Set the GDB server port

Ensure the value chosen is valid. Note that 0 is permitted, meaning the connection should be to the "or1ksim" service, rather than a port.

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure (not used here)

5.78.3.9 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

5.78.3.10 static int debug_set_mem (oraddr_t address, uint32_t data) [static]

Write to main bus

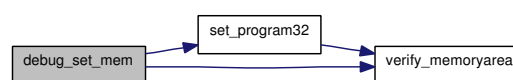
Parameters:

- ← *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:

**5.78.3.11 int debug_set_register (oraddr_t address, uorreg_t data)**

Set a JTAG register

Action depends on which scan chain is currently active.

Parameters:

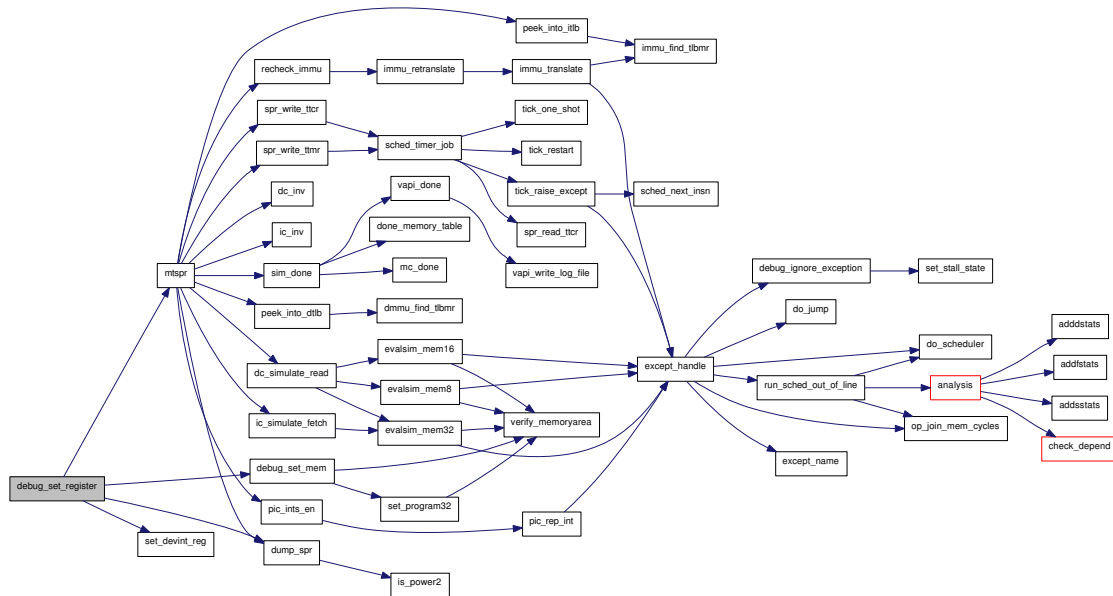
← *address* Address on the scan chain

→ *data* Data to set

Returns:

An error code (including ERR_NONE) if there is no error

Here is the call graph for this function:

**5.78.3.12 void debug_vapi_id (union param_val val, void * dat)**

Set the VAPI ID for the debug unit

Parameters:

← *val* The value to use

← *dat* The `config` data structure (not used here)

5.78.3.13 DECLARE_DEBUG_CHANNEL (jtag)**5.78.3.14 void du_reset ()**

Reset the debug unit

Clear all development interface registers

Here is the call graph for this function:



5.78.3.15 `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:

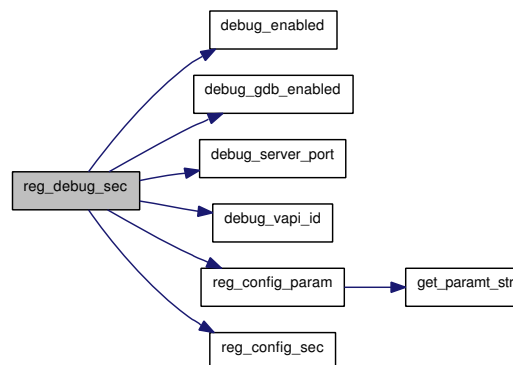
An error code (including ERR_NONE) if there is no error

5.78.3.16 `static int get_devint_reg (unsigned int addr, unsigned long *data)` [static]

5.78.3.17 `void reg_debug_sec ()`

Register the configuration functions for the debug unit

Here is the call graph for this function:



5.78.3.18 `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:

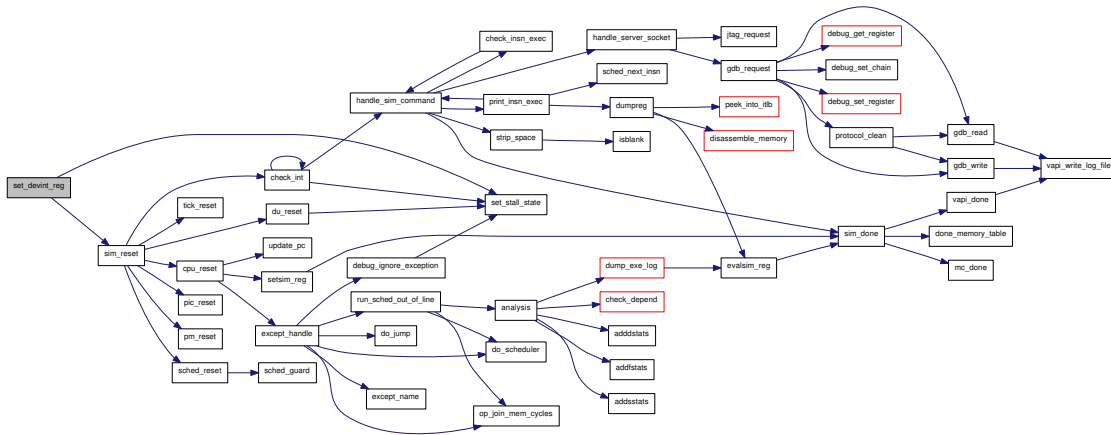
← *address* The register to set

← *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:



5.78.3.19 `static int set_devint_reg (unsigned int addr, unsigned long data)` [static]

5.78.3.20 `void set_stall_state (int state)`

Set the stall state of the processor

Parameters:

← *state* If non-zero stall the processor.

5.78.4 Variable Documentation

5.78.4.1 `enum debug_scan_chain_ids current_scan_chain = JTAG_CHAIN_GLOBAL`
[static]

The current scan chain being accessed

5.78.4.2 `unsigned long development[DEVELOPINT_MAX+1]`

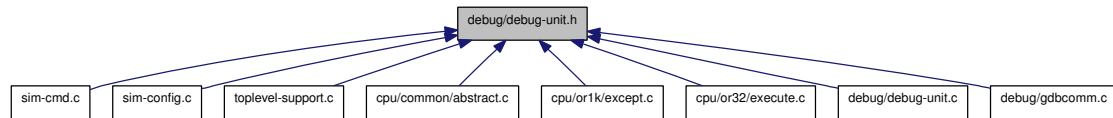
Data structure holding debug registers and their bits

5.78.4.3 `int in_reset = 0` [static]

External STALL signal to debug interface

5.79 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` ()

5.79.1 Enumeration Type Documentation

5.79.1.1 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

5.79.1.2 enum debug_unit_action

The possible debug unit actions

Enumerator:

DebugInstructionFetch

DebugLoadAddress

DebugStoreAddress

DebugLoadData

DebugStoreData

5.79.2 Function Documentation

5.79.2.1 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:



5.79.2.2 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

5.79.2.5 int debug_set_register (oraddr_t address, uorreg_t data)

Set a JTAG register

Action depends on which scan chain is currently active.

Parameters:

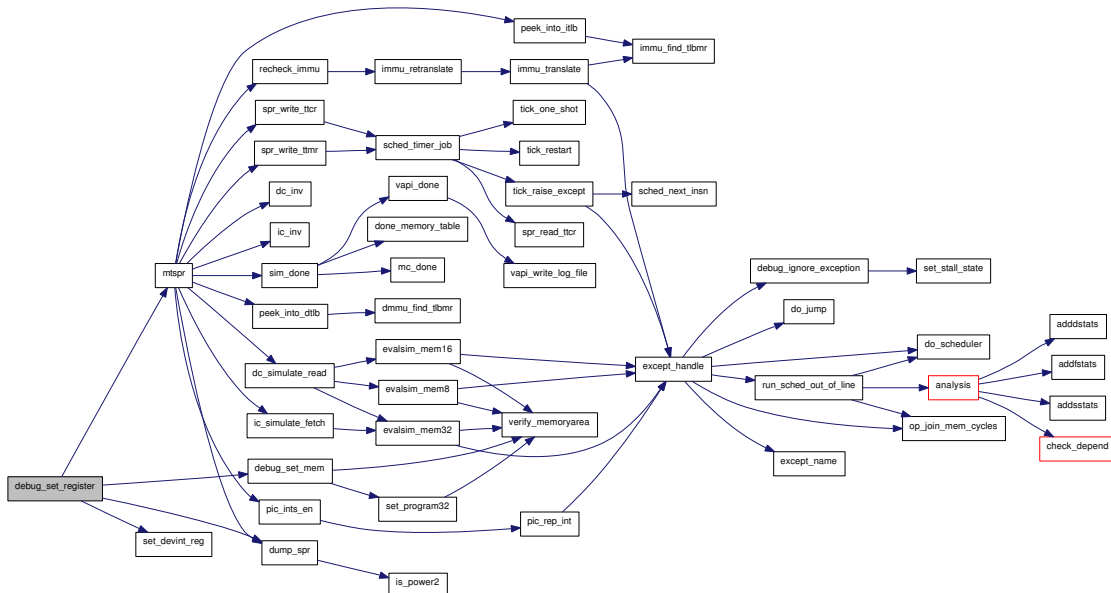
← *address* Address on the scan chain

→ *data* Data to set

Returns:

An error code (including ERR_NONE) if there is no error

Here is the call graph for this function:



5.79.2.6 void du_reset ()

Reset the debug unit

Clear all development interface registers

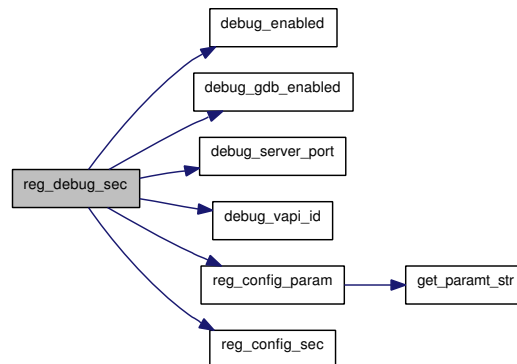
Here is the call graph for this function:



5.79.2.7 void reg_debug_sec ()

Register the configuration functions for the debug unit

Here is the call graph for this function:



5.79.2.8 void set_stall_state (int state)

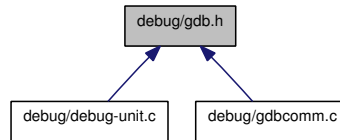
Set the stall state of the processor

Parameters:

← *state* If non-zero stall the processor.

5.80 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 }

5.80.1 Enumeration Type Documentation

5.80.1.1 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

5.80.1.2 enum or1k_jtag_proxy_protocol_commands

The OR1K JTAG proxy protocol commands.

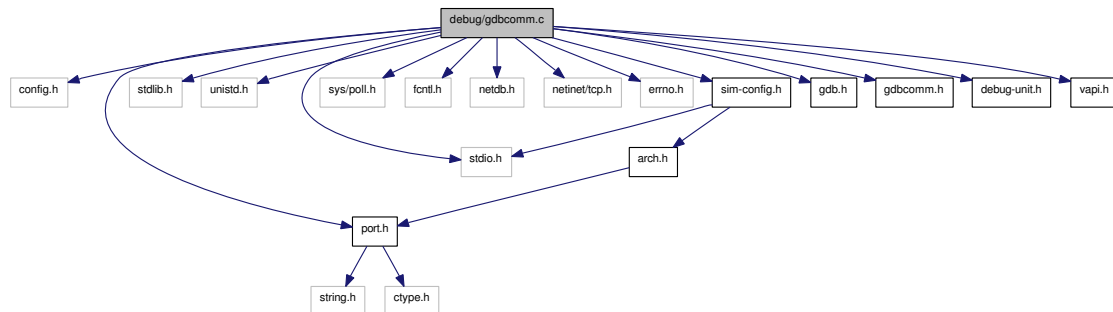
Enumerator:

ORIK_JTAG_COMMAND_READ
ORIK_JTAG_COMMAND_WRITE
ORIK_JTAG_COMMAND_READ_BLOCK
ORIK_JTAG_COMMAND_WRITE_BLOCK
ORIK_JTAG_COMMAND_CHAIN

5.81 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 <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

5.81.1 Function Documentation

5.81.1.1 void `block_jtag` ()

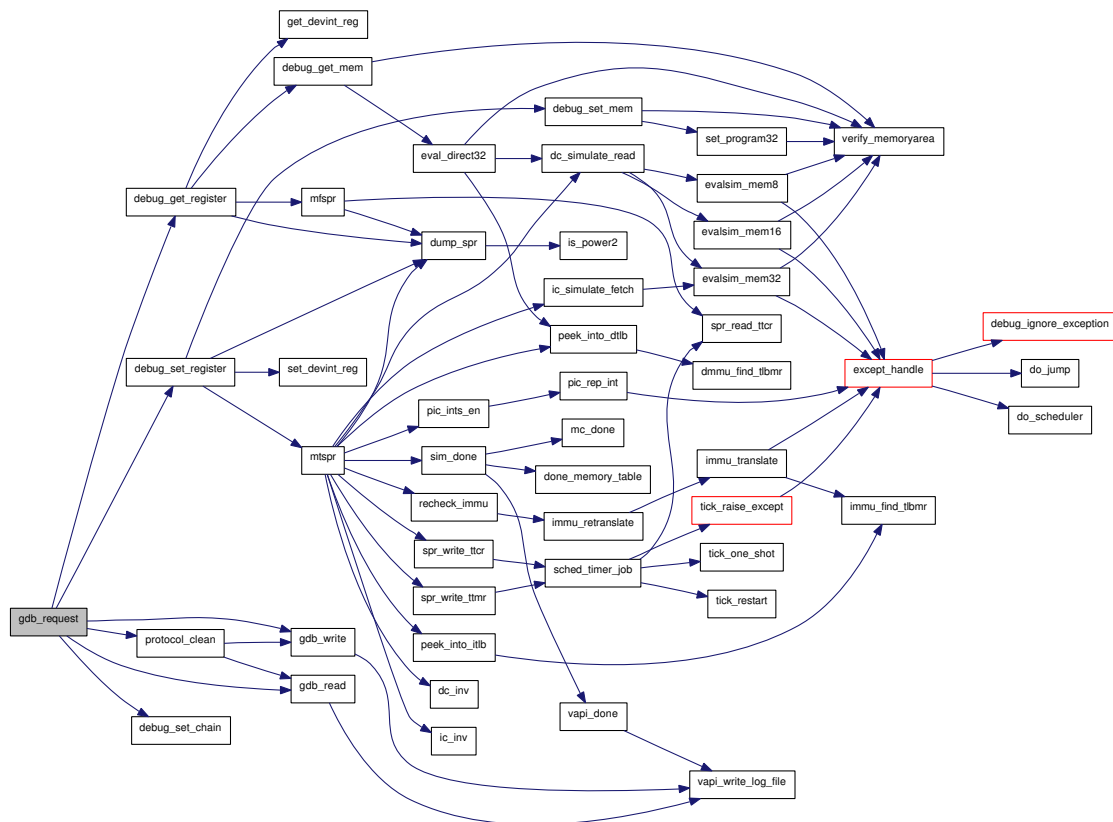
5.81.1.2 static int `gdb_read` (void * *buf*, int *len*) [static]

Here is the call graph for this function:



5.81.1.3 static void `gdb_request` (void) [static]

Here is the call graph for this function:



5.81.1.4 static int gdb_write (const void * buf, int len) [static]

Here is the call graph for this function:



5.81.1.5 void gdbcomm_init ()

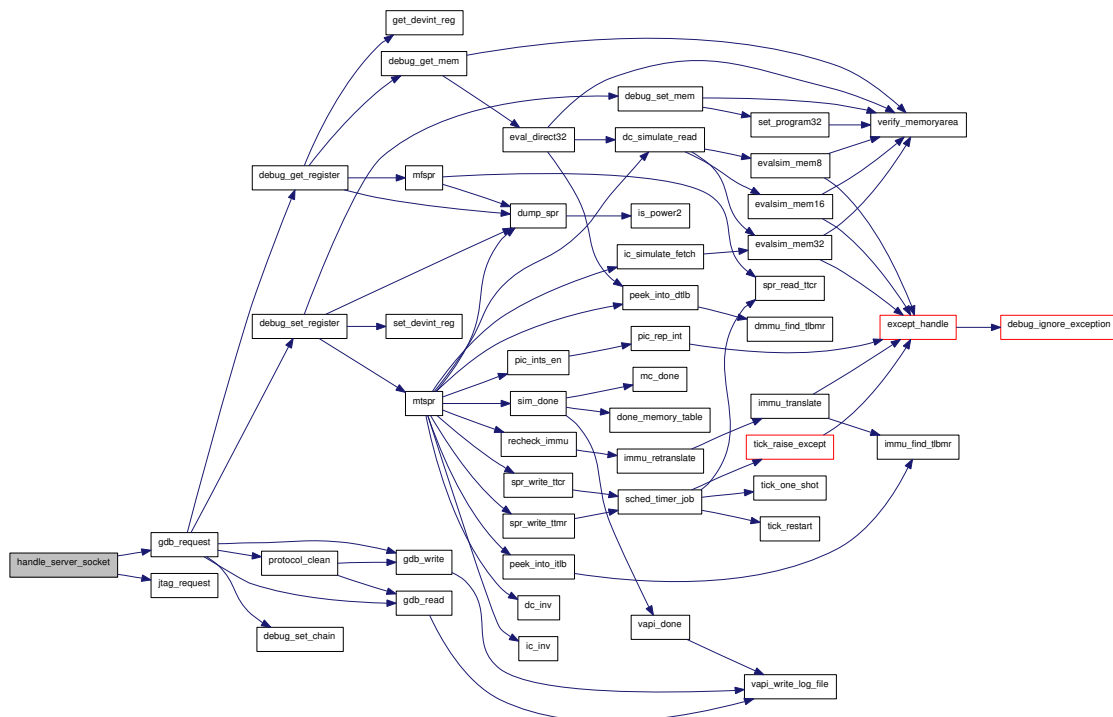
Here is the call graph for this function:



5.81.1.6 static int get_server_socket (const char * name, const char * proto, int port) [static]

5.81.1.7 void handle_server_socket (enum boolean block)

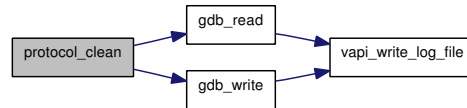
Here is the call graph for this function:



5.81.1.8 `static void jtag_request (void)` [static]

5.81.1.9 `static void protocol_clean (int length, int32_t err)` [static]

Here is the call graph for this function:



5.81.2 Variable Documentation

5.81.2.1 `unsigned int gdb_fd = 0` [static]

5.81.2.2 `unsigned int server_fd = 0` [static]

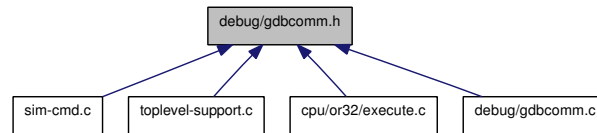
5.81.2.3 `unsigned int server_ip = 0` [static]

5.81.2.4 `unsigned int server_port = 0` [static]

5.81.2.5 `int tcp_level = 0` [static]

5.82 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` ()

5.82.1 Enumeration Type Documentation

5.82.1.1 enum `boolean`

Enumerator:

FALSE

TRUE

5.82.2 Function Documentation

5.82.2.1 void `block_jtag` ()

5.82.2.2 void `gdbcomm_init` ()

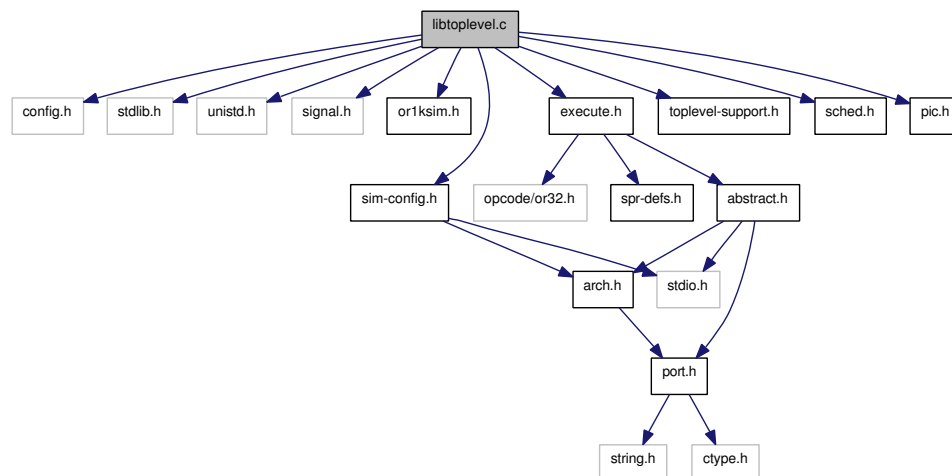
Here is the call graph for this function:



5.83 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)

5.83.1 Function Documentation

5.83.1.1 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

5.83.1.2 unsigned long int or1ksim_clock_rate ()

Return the clock rate

Value is part of the configuration

Returns:

Clock rate in Hz.

5.83.1.3 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:



5.83.1.4 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.

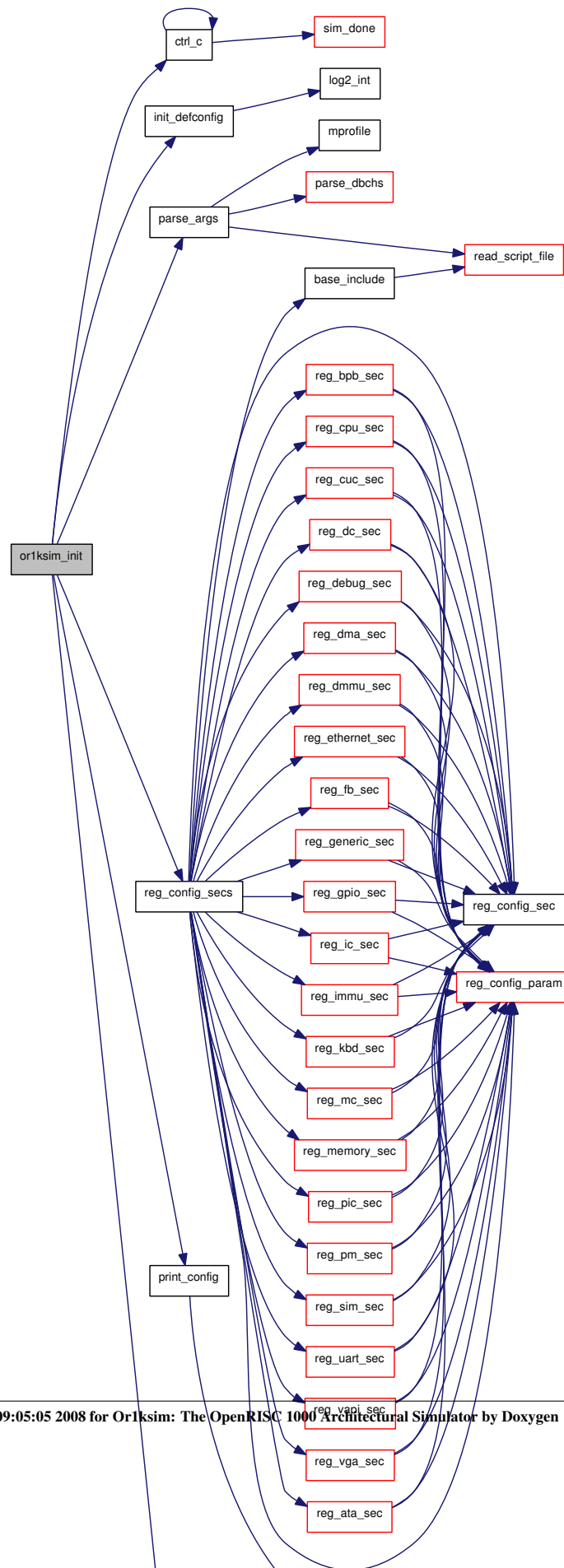
Parameters:

- ← *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:



5.83.1.5 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

5.83.1.6 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.

5.83.1.7 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

5.83.1.8 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)

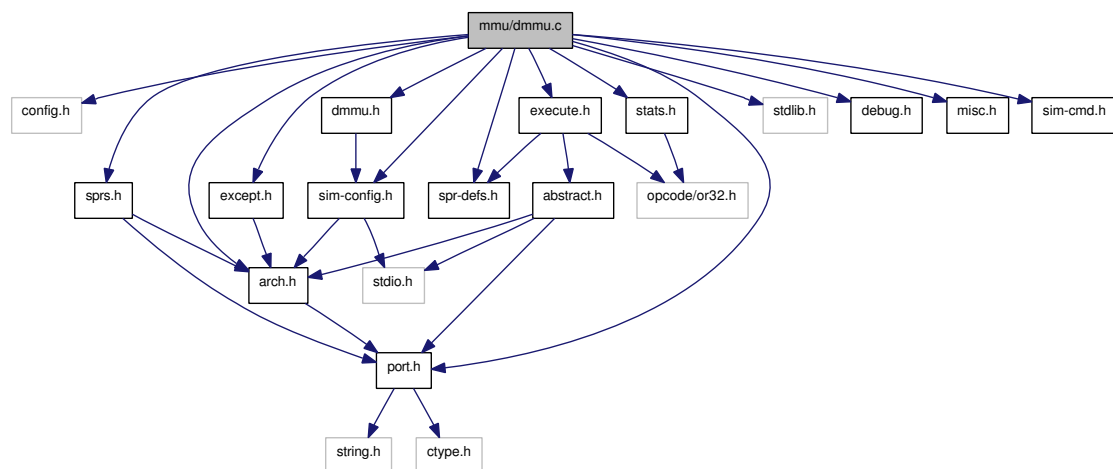
5.84 mainpage File Reference

5.84.1 Detailed Description

5.85 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 "debug.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

- `DEFAULT_DEBUG_CHANNEL` (dmmu)
- `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`

5.85.1 Function Documentation

5.85.1.1 DEFAULT_DEBUG_CHANNEL (dmmu)

5.85.1.2 static void `dmmu_enabled` (union `param_val` `val`, void `* dat`) [static]

Enable or disable the DMMU

Set the corresponding field in the UPR

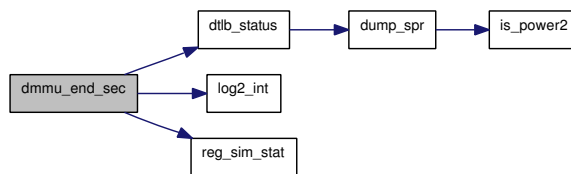
Parameters:

← `val` The value to use

← `dat` The `config` data structure

5.85.1.3 static void `dmmu_end_sec` (void `* dat`) [static]

Here is the call graph for this function:



5.85.1.4 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:



5.85.1.5 `static uorreg_t* dmmu_find_tlbmr (oraddr_t virtaddr, uorreg_t** dtlbmr_lru, struct dmmu * dmmu)` [static]

5.85.1.6 `static void dmmu_hitdelay (union param_val val, void * dat)` [static]

5.85.1.7 `static void dmmu_missdelay (union param_val val, void * dat)` [static]

5.85.1.8 `static void dmmu_nsets (union param_val val, void * dat)` [static]

Set the number of DMMU sets

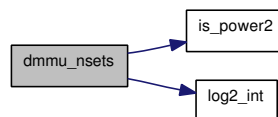
Value must be a power of 2 <= 256. Ignore any other values with a warning. Set the corresponding DMMU configuration flags.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



5.85.1.9 `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

5.85.1.10 `static void dmmu_pagesize (union param_val val, void * dat)` [static]

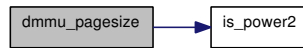
Set the DMMU page size

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:

**5.85.1.11 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.

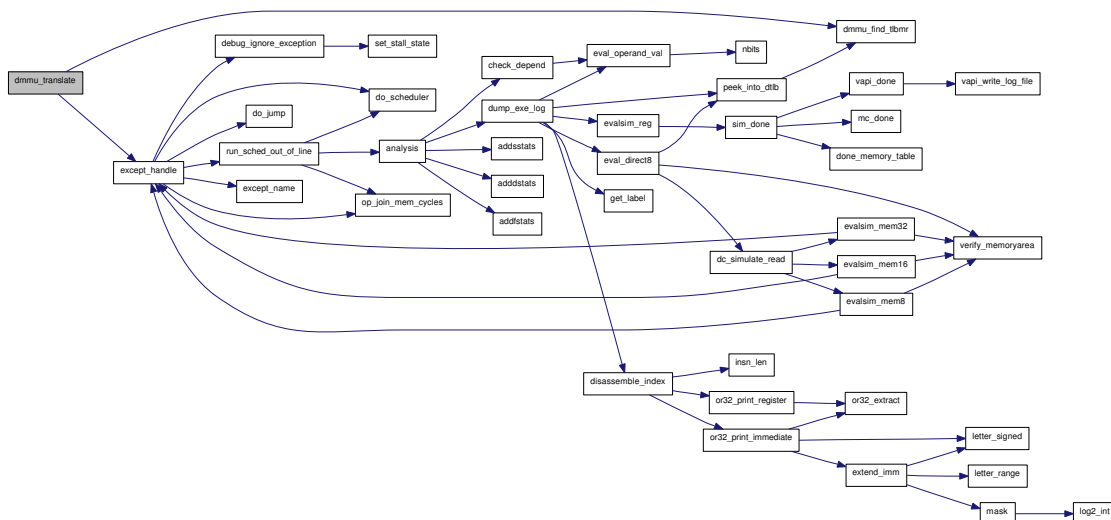
Returns:

The new memory configuration data structure

Here is the call graph for this function:

**5.85.1.12 oraddr_t dmmu_translate (oraddr_t virtaddr, int write_access)**

Here is the call graph for this function:



5.85.1.13 `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

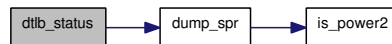
Parameters:

← *val* The value to use

← *dat* The `config` data structure

5.85.1.14 `static void dtlb_status (void * dat)` [static]

Here is the call graph for this function:

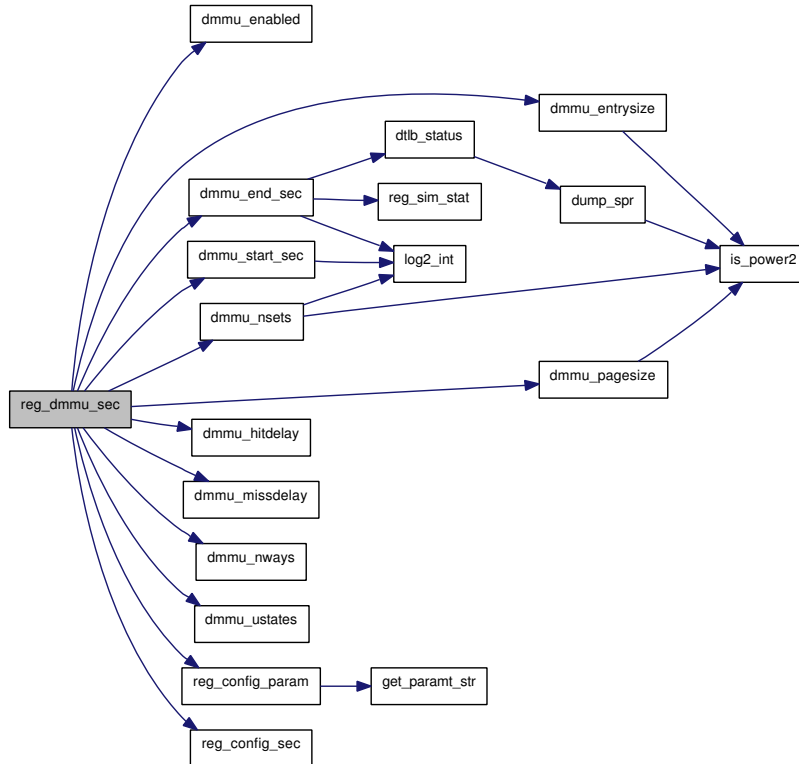
**5.85.1.15** `oraddr_t peek_into_dtlb (oraddr_t virtaddr, int write_access, int through_dc)`

Here is the call graph for this function:



5.85.1.16 void reg_dmmu_sec (void)

Here is the call graph for this function:



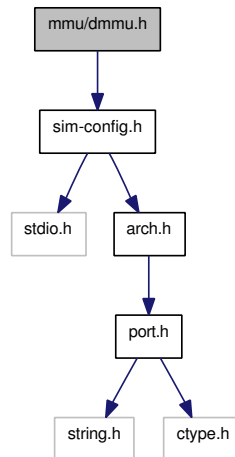
5.85.2 Variable Documentation

5.85.2.1 struct dmmu* dmmu_state

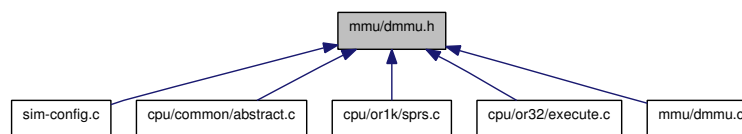
5.86 mmu/dmmu.h File Reference

```
#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](#)

5.86.1 Define Documentation

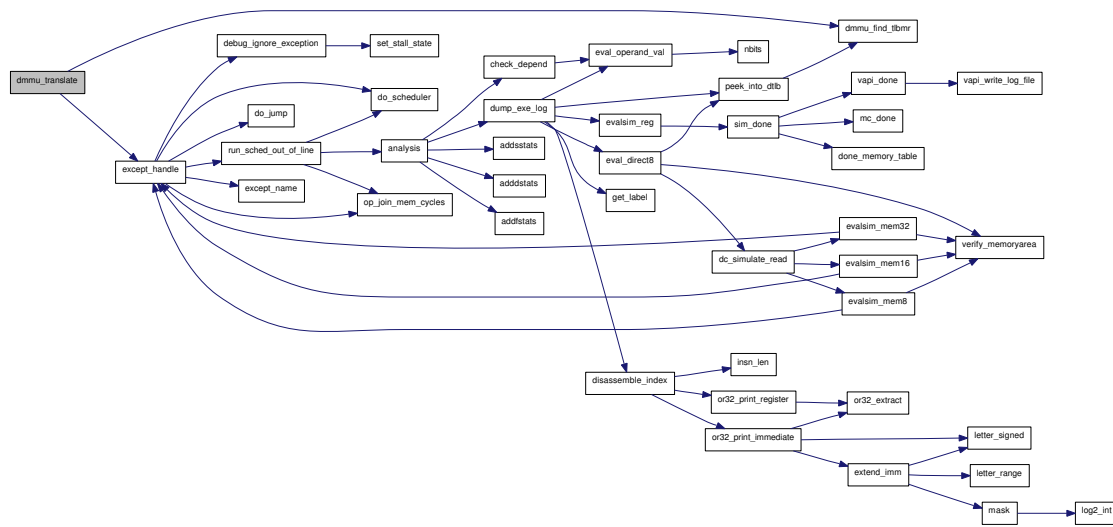
5.86.1.1 `#define DADDR_PAGE(addr) ((addr) & dmmu_state → page_mask)`

5.86.2 Function Documentation

5.86.2.1 `oraddr_t dmmu_simulate_tlb (oraddr_t virtaddr, int write_access)`

5.86.2.2 `oraddr_t dmmu_translate (oraddr_t virtaddr, int write_access)`

Here is the call graph for this function:



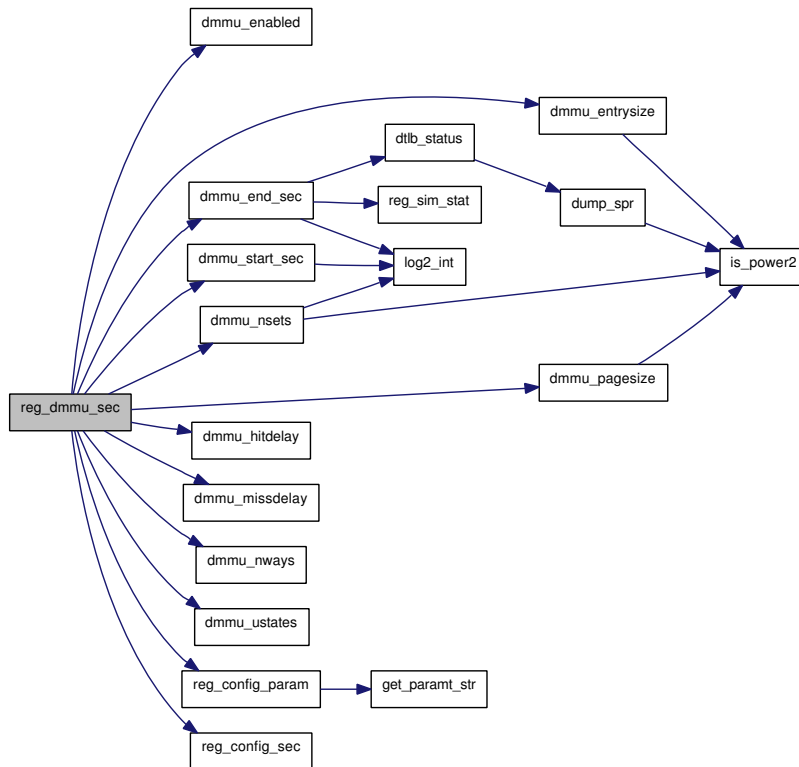
5.86.2.3 `oraddr_t peek_into_dtlb (oraddr_t virtaddr, int write_access, int through_dc)`

Here is the call graph for this function:



5.86.2.4 void reg_dmmu_sec ()

Here is the call graph for this function:



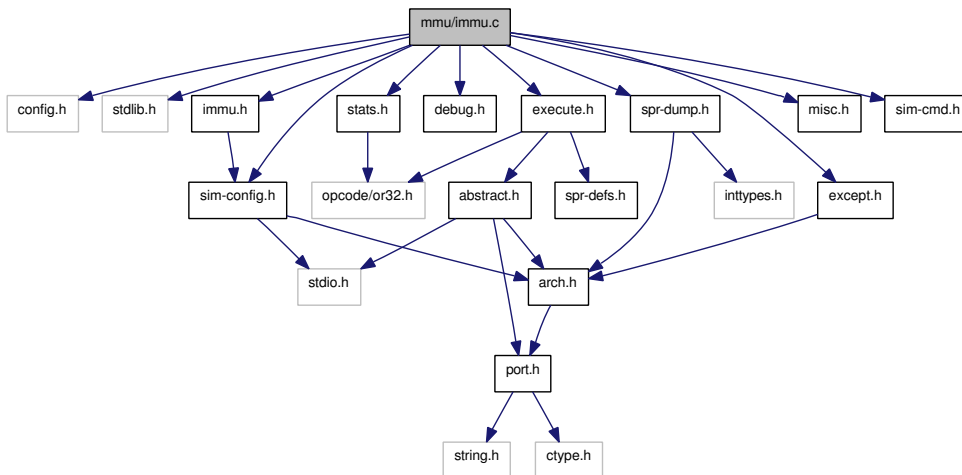
5.86.3 Variable Documentation

5.86.3.1 struct dmmu* dmmu_state

5.87 mmu/immu.c File Reference

```
#include "config.h"
#include <stdlib.h>
#include "immu.h"
#include "sim-config.h"
#include "execute.h"
#include "debug.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

- `DEFAULT_DEBUG_CHANNEL` (`immu`)
- 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_entrysize` (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`

5.87.1 Function Documentation

5.87.1.1 DEFAULT_DEBUG_CHANNEL (immu)

5.87.1.2 static void `immu_enabled` (union `param_val` val, void * dat) [static]

Enable or disable the IMMU

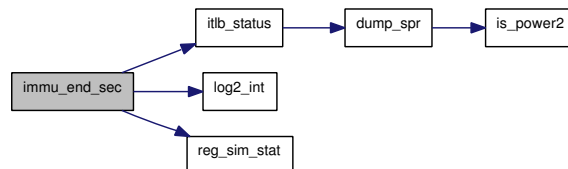
Set the corresponding field in the UPR

Parameters:

- ← *val* The value to use
- ← *dat* The `config` data structure

5.87.1.3 static void `immu_end_sec` (void * dat) [static]

Here is the call graph for this function:



5.87.1.4 static void `immu_entsize` (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:



5.87.1.5 `static uorreg_t* immu_find_tlbmr (oraddr_t virtaddr, uorreg_t** itlbmr_lru, struct immu * immu)` [static]

5.87.1.6 `static void immu_hitdelay (union param_val val, void * dat)` [static]

5.87.1.7 `static void immu_missdelay (union param_val val, void * dat)` [static]

5.87.1.8 `static void immu_nsets (union param_val val, void * dat)` [static]

Set the number of DMMU sets

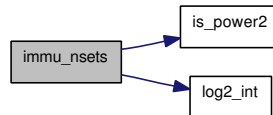
Value must be a power of 2 <= 256. Ignore any other values with a warning. Set the corresponding IMMU configuration flags.

Parameters:

← *val* The value to use

← *dat* The `config` data structure

Here is the call graph for this function:



5.87.1.9 `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

5.87.1.10 `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:



5.87.1.11 static void* immu_start_sec () [static]

Initialize a new DMMU configuration

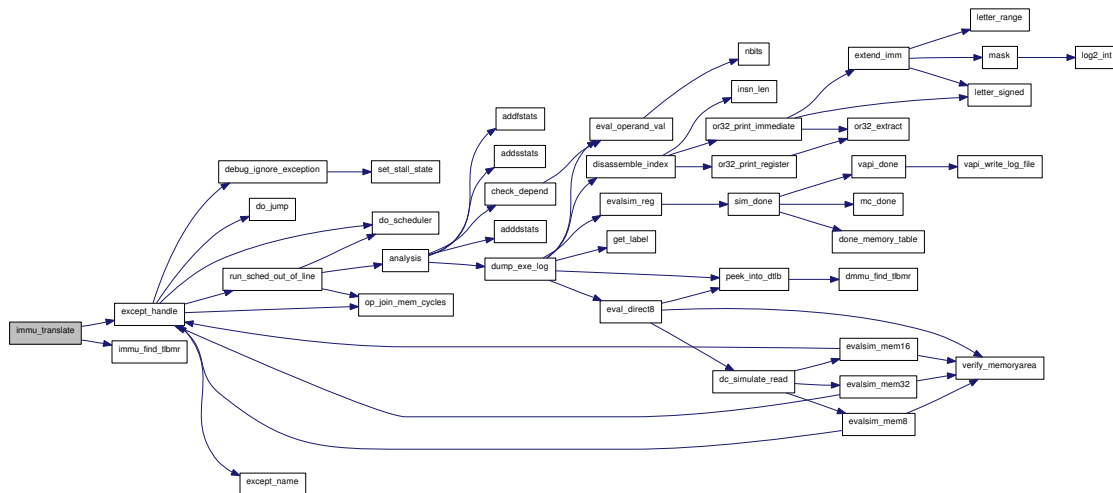
ALL parameters are set explicitly to default values.

Here is the call graph for this function:



5.87.1.12 oraddr_t immu_translate (oraddr_t virtaddr)

Here is the call graph for this function:



5.87.1.13 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

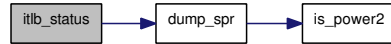
Parameters:

← *val* The value to use

← *dat* The `config` data structure

5.87.1.14 static void itlb_status (void * dat) [static]

Here is the call graph for this function:



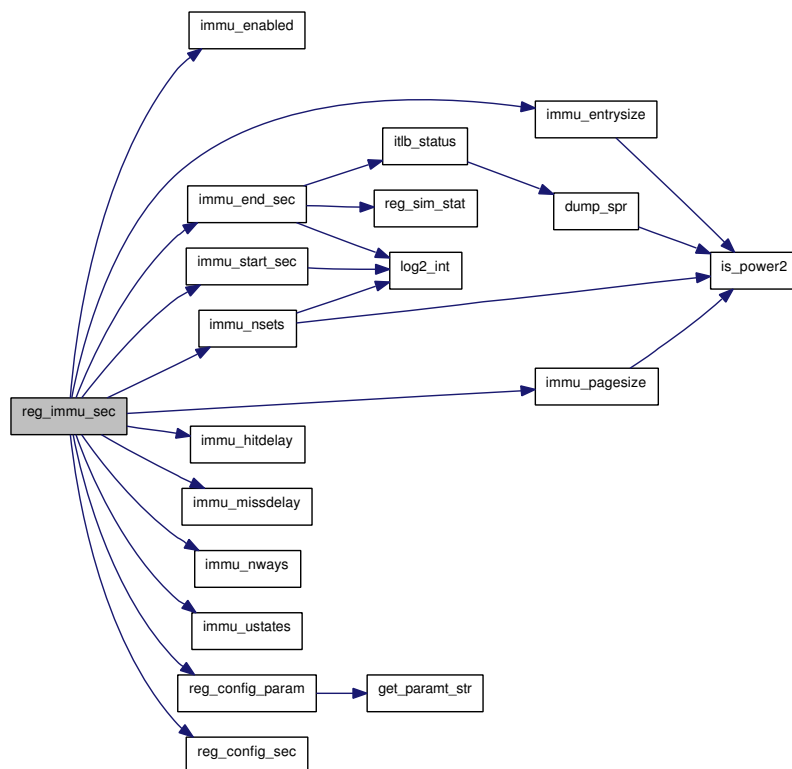
5.87.1.15 oraddr_t peek_into_itlb (oraddr_t virtaddr)

Here is the call graph for this function:



5.87.1.16 void reg_immu_sec (void)

Here is the call graph for this function:



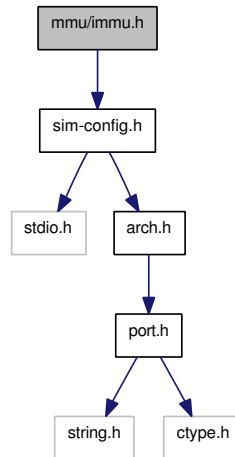
5.87.2 Variable Documentation

5.87.2.1 struct immu* immu_state

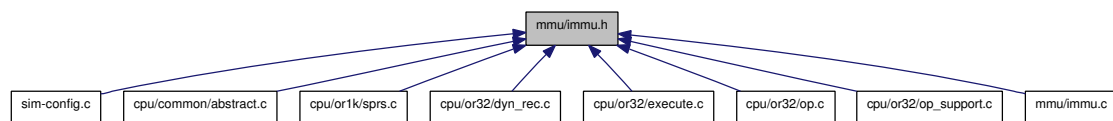
5.88 mmu/immu.h File Reference

```
#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)` $((addr) \& \text{immu_state} \rightarrow \text{page_mask})$

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`

5.88.1 Define Documentation

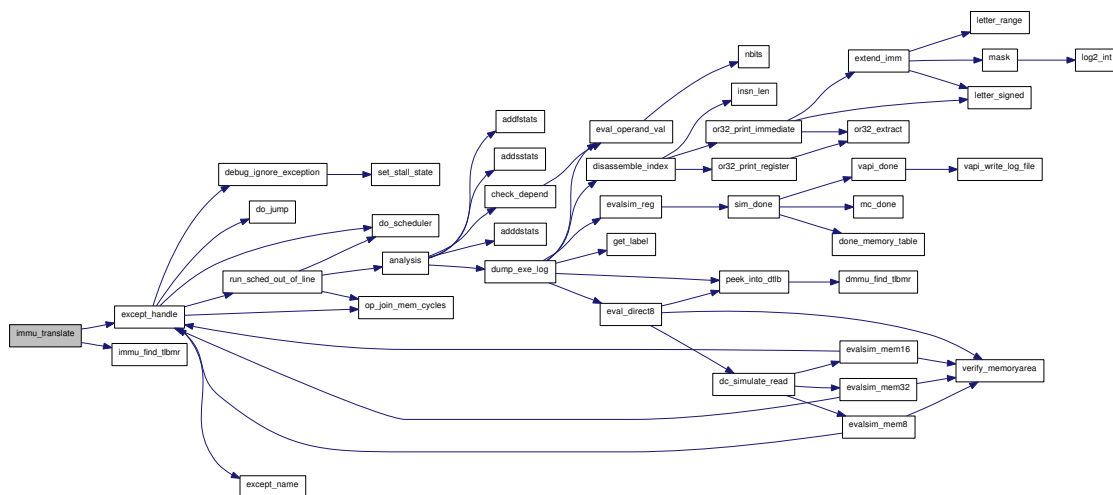
5.88.1.1 `#define IADDR_PAGE(addr) ((addr) & immu_state → page_mask)`

5.88.2 Function Documentation

5.88.2.1 `oraddr_t immu_simulate_tlb (oraddr_t virtaddr)`

5.88.2.2 `oraddr_t immu_translate (oraddr_t virtaddr)`

Here is the call graph for this function:



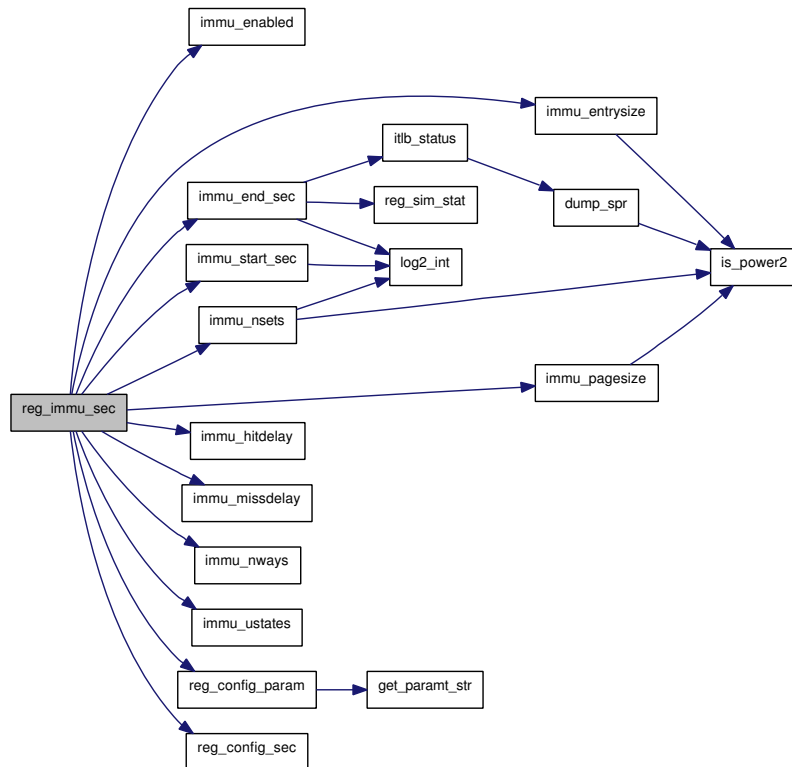
5.88.2.3 `oraddr_t peek_into_itlb (oraddr_t virtaddr)`

Here is the call graph for this function:



5.88.2.4 void reg_immu_sec ()

Here is the call graph for this function:



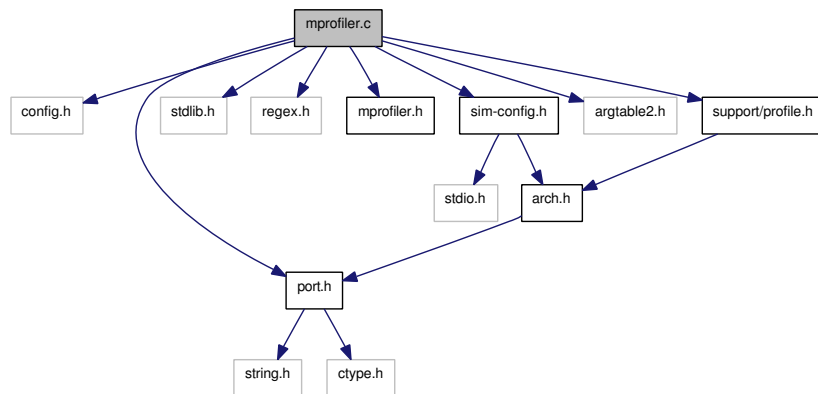
5.88.3 Variable Documentation

5.88.3.1 struct immu* immu_state

5.89 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

5.89.1 Define Documentation

5.89.1.1 `#define BUF_SIZE 256`

5.89.1.2 `#define HASH_FUNC(x) ((x) & 0xffff)`

5.89.1.3 `#define HASH_SIZE 0x10000`

5.89.1.4 `#define MODE_ACCESS 2`

5.89.1.5 `#define MODE_DETAIL 0`

5.89.1.6 `#define MODE_PRETTY 1`

5.89.1.7 `#define MODE_WIDTH 3`

5.89.2 Function Documentation

5.89.2.1 `static void hash_add (oraddr_t addr, int index)` [static]

5.89.2.2 `static unsigned long hash_get (oraddr_t addr, int index)` [static]

5.89.2.3 `static void init ()` [static]

5.89.2.4 `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.

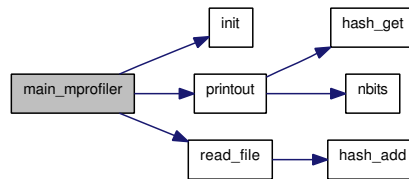
Parameters:

- ← *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

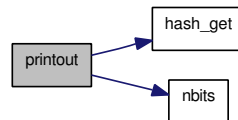
Here is the call graph for this function:



5.89.2.5 `static int nbits (unsigned long a)` [static]

5.89.2.6 `static void printout (int mode)` [static]

Here is the call graph for this function:



5.89.2.7 `static void read_file (FILE *f, int mode)` [static]

Here is the call graph for this function:



5.89.3 Variable Documentation

5.89.3.1 `oraddr_t end_addr = 0xffffffff` [static]

End address

5.89.3.2 `FILE* fprof = 0` [static]

5.89.3.3 `int group_bits = 2` [static]

Groups size – how much addresses should be joined together

5.89.3.4 `struct memory_hash * hash[HASH_SIZE]` [static]

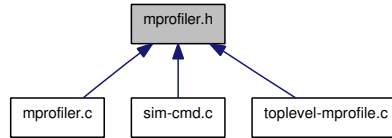
Hash table data structure

5.89.3.5 oraddr_t start_addr = 0 [static]

Start address

5.90 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)

5.90.1 Function Documentation

5.90.1.1 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.

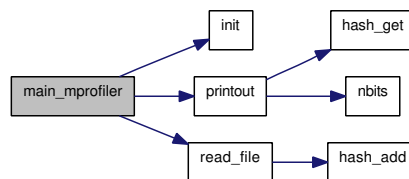
Parameters:

- ← *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:

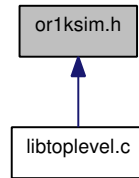
0 on success, 1 on failure

Here is the call graph for this function:



5.91 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)

5.91.1 Enumeration Type Documentation

5.91.1.1 enum `or1ksim_rc`

Enumerator:

`ORIKSIM_RC_OK`
`ORIKSIM_RC_BADINIT`
`ORIKSIM_RC_BRKPT`

5.91.2 Function Documentation

5.91.2.1 unsigned long int `or1ksim_clock_rate` ()

Return the clock rate

Value is part of the configuration

Returns:

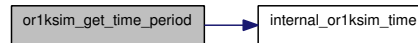
Clock rate in Hz.

5.91.2.2 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:



5.91.2.3 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*)

5.91.2.4 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

5.91.2.5 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.

5.91.2.6 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).

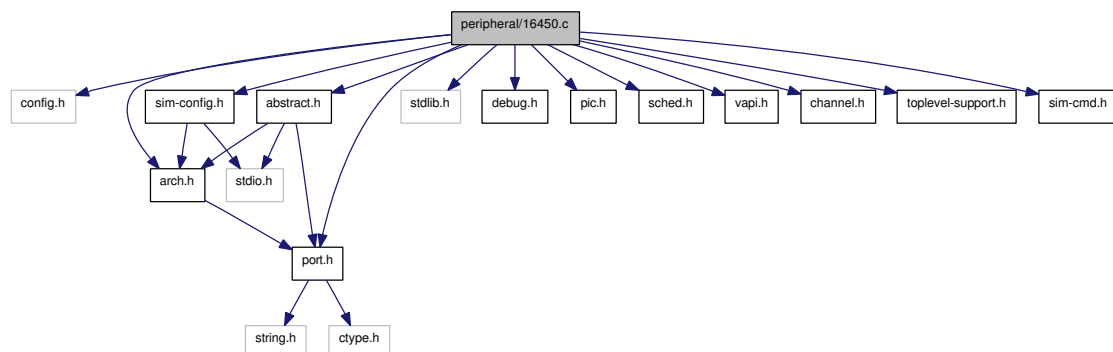
Parameters:

← *duration* Time to run for in seconds

5.92 peripheral/16450.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "sim-config.h"
#include "debug.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_TXSERE` 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

- `DEFAULT_DEBUG_CHANNEL` (uart)
- 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)

5.92.1 Define Documentation

5.92.1.1 `#define MAX_SKEW 1`

max. clock skew in subclocks

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

5.92.1.3 `#define UART_ADDR_SPACE 8`

UART addr space size in bytes

5.92.1.4 `#define UART_BREAK_COUNT 1`

5.92.1.5 `#define UART_CHAR_TIMEOUT 4`

5.92.1.6 `#define UART_CLOCK_DIVIDER 16`

UART clock divider

5.92.1.7 `#define UART_DLH 1`

5.92.1.8 `#define UART_DLL 0`

5.92.1.9 `#define UART_FCR 2`

5.92.1.10 `#define UART_FCR_FIE 0x01`

5.92.1.11 `#define UART_FCR_RRXFI 0x02`

5.92.1.12 `#define UART_FCR_RTXFI 0x04`

5.92.1.13 `#define UART_FGETC_SLOWDOWN 100`

fgetc() slowdown factor

5.92.1.14 #define UART_FIFO_TRIGGER(x)**Value:**

```
/* Trigger values for indexes 0..3 */\  
((x) == 0 ? 1\  
:(x) == 1 ? 4\  
:(x) == 2 ? 8\  
:(x) == 3 ? 14 : 0)
```

5.92.1.15 **#define** UART_IER 1

5.92.1.16 **#define** UART_IER_MSI 0x08

5.92.1.17 **#define** UART_IER_RDI 0x01

5.92.1.18 **#define** UART_IER_RLSI 0x04

5.92.1.19 **#define** UART_IER_THRI 0x02

5.92.1.20 **#define** UART_IIR 2

5.92.1.21 **#define** UART_IIR_CTI 0x0c

5.92.1.22 **#define** UART_IIR_ID 0x06

5.92.1.23 **#define** UART_IIR_MSI 0x00

5.92.1.24 **#define** UART_IIR_NO_INT 0x01

5.92.1.25 **#define** UART_IIR_RDI 0x04

5.92.1.26 **#define** UART_IIR_RLSI 0x06

5.92.1.27 **#define** UART_IIR_THRI 0x02

5.92.1.28 **#define** UART_LCR 3

5.92.1.29 **#define** UART_LCR_DLAB 0x80

5.92.1.30 **#define** UART_LCR_EPAR 0x10

5.92.1.31 **#define** UART_LCR_PARITY 0x08

5.92.1.32 **#define** UART_LCR_RESET 0x03

5.92.1.33 **#define** UART_LCR_SBC 0x40

5.92.1.34 **#define** UART_LCR_SPAR 0x20

5.92.1.35 **#define** UART_LCR_STOP 0x04

5.92.1.36 **#define** UART_LCR_WLEN5 0x00

5.92.1.37 **#define** UART_LCR_WLEN6 0x01

5.92.1.38 **#define** UART_LCR_WLEN7 0x02

5.92.1.39 **#define** UART_LCR_WLEN8 0x03

5.92.1.40 **#define** UART_LSR 5

5.92.1.41 **#define** UART_LSR_BREAK 0x10

5.92.1.42 **#define** UART_LSR_FRAME 0x08

5.92.1.43 **#define** UART_LSR_OVRRUN 0x02

5.92.1.44 **#define** UART_LSR_PARITY 0x04

5.92.1.45 **#define** UART_LSR_RDRDY 0x01

5.92.1.50 **#define UART_MCR 4**

5.92.1.51 **#define UART_MCR_AUX1 0x04**

5.92.1.52 **#define UART_MCR_AUX2 0x08**

5.92.1.53 **#define UART_MCR_DTR 0x01**

5.92.1.54 **#define UART_MCR_LOOP 0x10**

5.92.1.55 **#define UART_MCR_RTS 0x02**

5.92.1.56 **#define UART_MSR 6**

5.92.1.57 **#define UART_MSR_CTS 0x10**

5.92.1.58 **#define UART_MSR_DCD 0x80**

5.92.1.59 **#define UART_MSR_DCTS 0x01**

5.92.1.60 **#define UART_MSR_DDCD 0x08**

5.92.1.61 **#define UART_MSR_DDSR 0x02**

5.92.1.62 **#define UART_MSR_DSR 0x20**

5.92.1.63 **#define UART_MSR_RI 0x40**

5.92.1.64 **#define UART_MSR_TERI 0x04**

5.92.1.65 **#define UART_RXBUF 0**

5.92.1.66 **#define UART_SCR 7**

5.92.1.67 **#define UART_TXBUF 0**

5.92.1.68 **#define UART_VALID_FCR 0xc0**

5.92.1.69 **#define UART_VALID_IER 0x0f**

5.92.1.70 **#define UART_VALID_IIR 0x0f**

5.92.1.71 **#define UART_VALID_LCR 0xff**

5.92.1.72 **#define UART_VALID_LSR 0xff**

5.92.1.73 **#define UART_VALID_MCR 0x1f**

5.92.1.74 **#define UART_VALID_MSR 0xff**

5.92.1.75 **#define UART_VAPI_BUF_LEN 128**

Size of VAPI command buffer

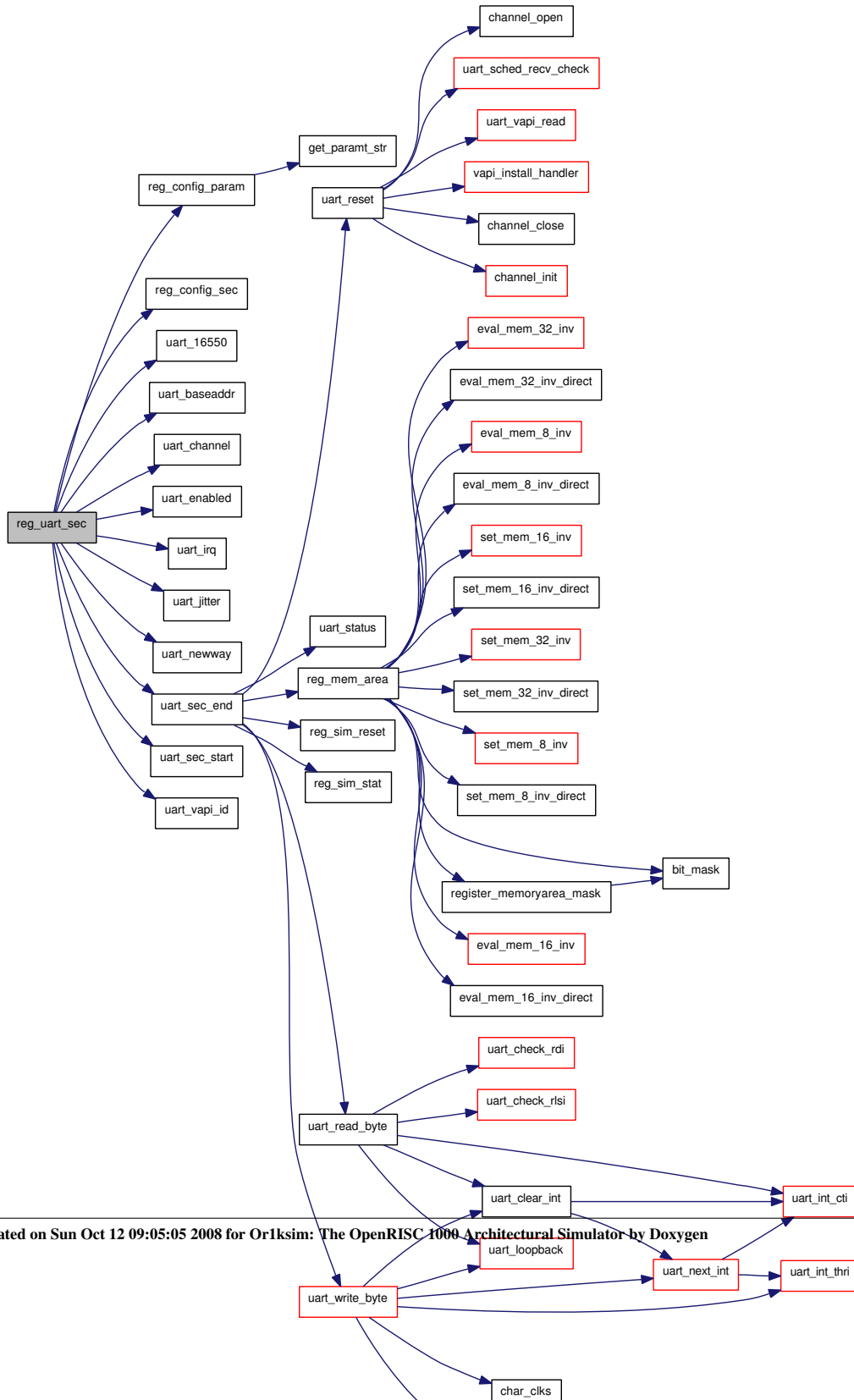
5.92.2 Function Documentation

5.92.2.1 static unsigned long char_clks (int dll, int dlh, int lcr) [static]

5.92.2.2 DEFAULT_DEBUG_CHANNEL (uart)

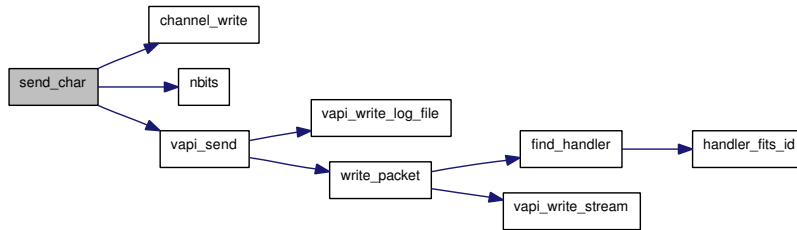
5.92.2.3 void reg_uart_sec (void)

Here is the call graph for this function:



5.92.2.4 static void send_char (struct dev_16450 * uart, int bits_send) [static]

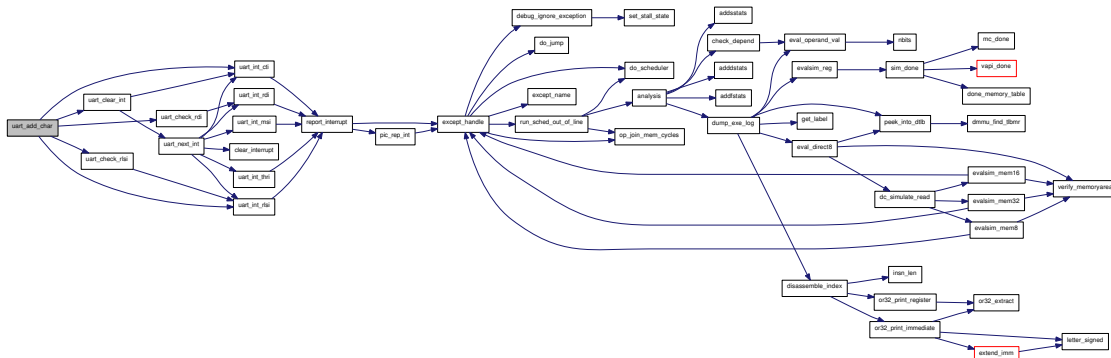
Here is the call graph for this function:



5.92.2.5 static void uart_16550 (union param_val val, void * dat) [static]

5.92.2.6 static void uart_add_char (struct dev_16450 * uart, int ch) [static]

Here is the call graph for this function:



5.92.2.7 static void uart_baseaddr (union param_val val, void * dat) [static]

5.92.2.8 static void uart_channel (union param_val val, void * dat) [static]

Set the [channel](#) description

Free any existing string.

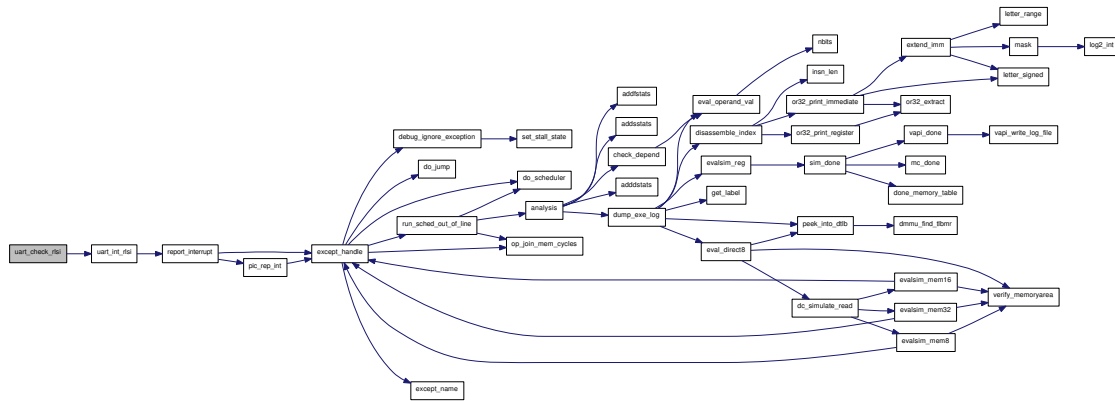
Parameters:

← *val* The value to use

← *dat* The [config](#) data structure

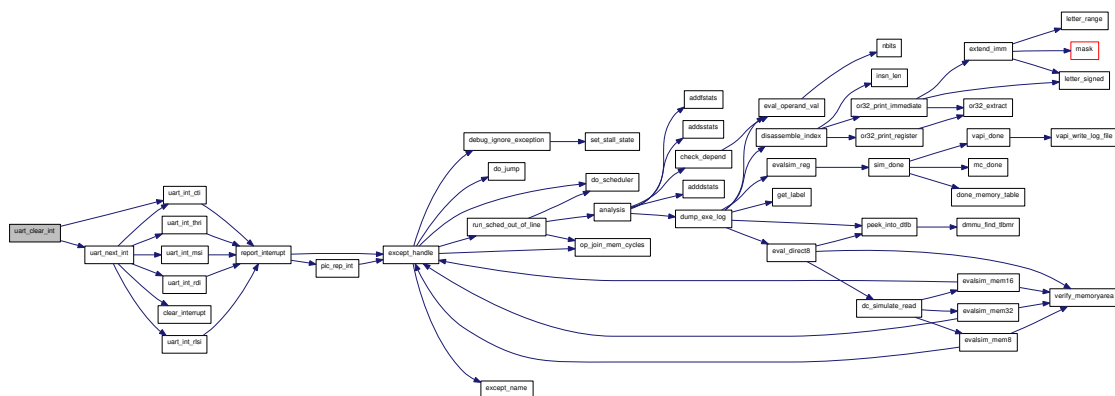
5.92.2.12 static void uart_check_rlsi (void * dat) [static]

Here is the call graph for this function:



5.92.2.13 static void uart_clear_int (struct dev_16450 * uart, int intr) [static]

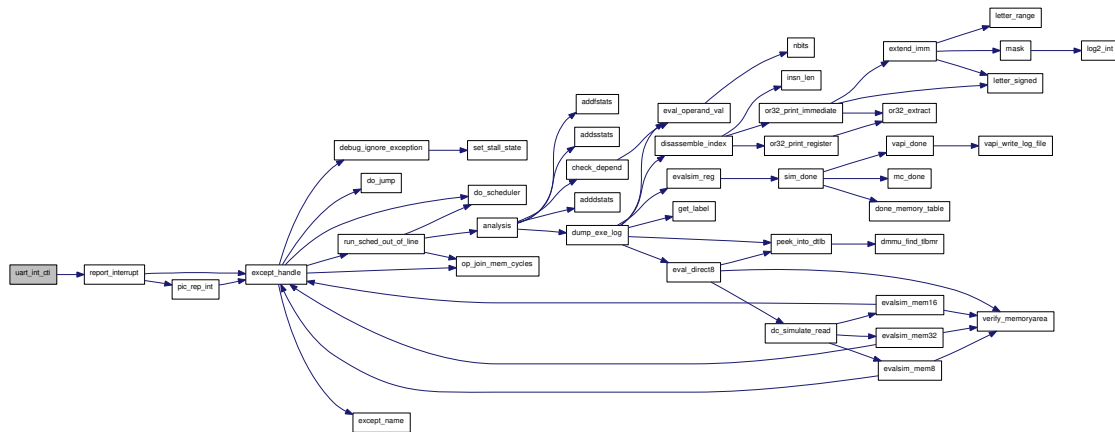
Here is the call graph for this function:



5.92.2.14 static void `uart_enabled` (union `param_val val`, void * `dat`) [static]

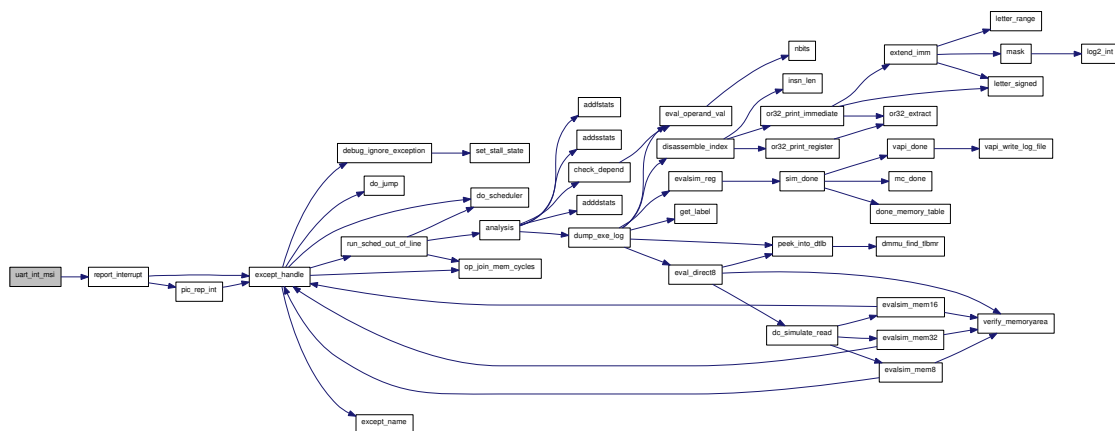
5.92.2.15 static void `uart_int_cti` (void * `dat`) [static]

Here is the call graph for this function:



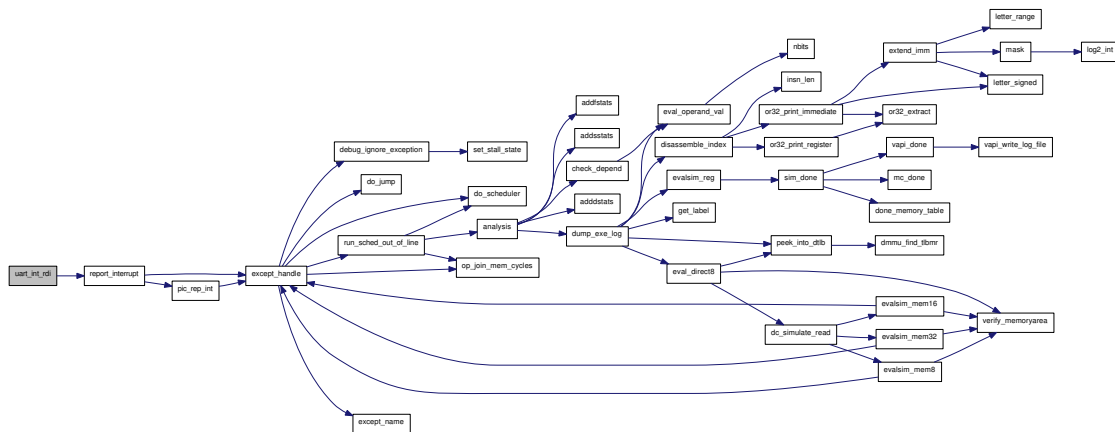
5.92.2.16 static void `uart_int_msi` (void * `dat`) [static]

Here is the call graph for this function:



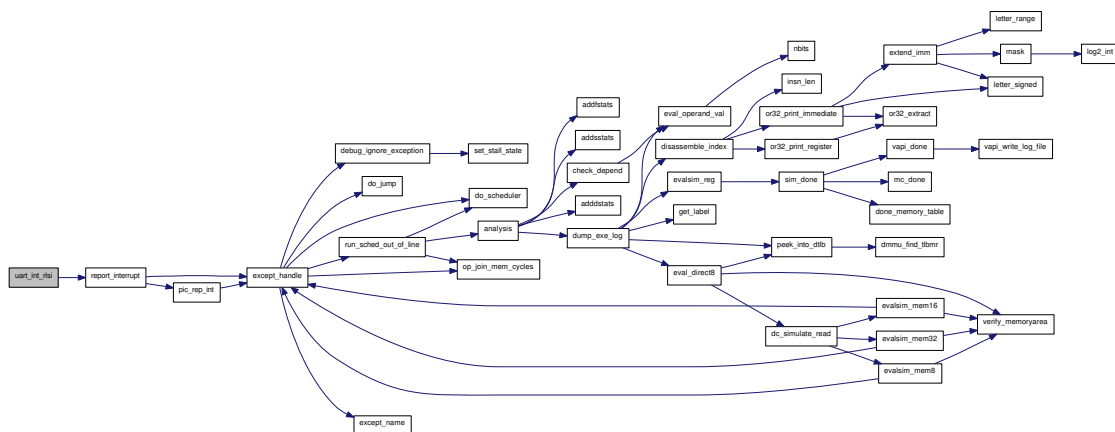
5.92.2.17 static void uart_int_rdi (void * dat) [static]

Here is the call graph for this function:



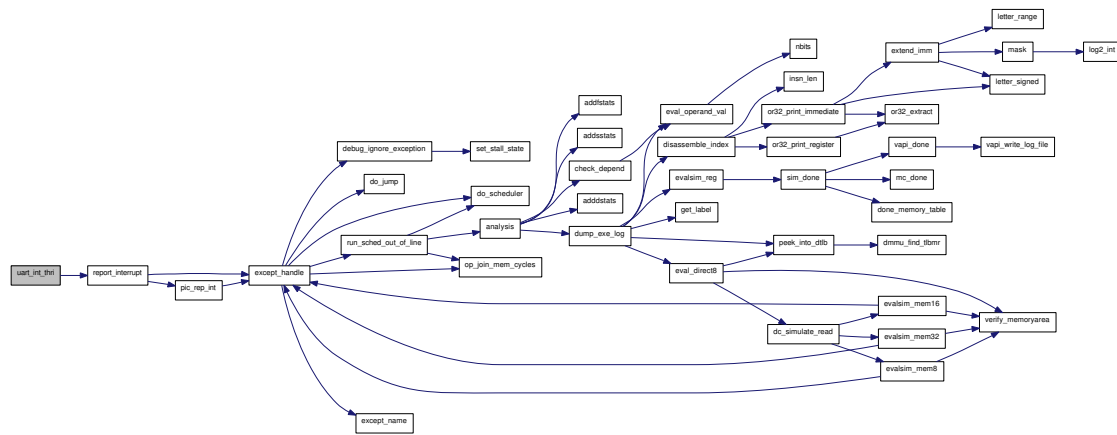
5.92.2.18 static void uart_int_rlsi (void * dat) [static]

Here is the call graph for this function:



5.92.2.19 static void uart_int_thri (void * dat) [static]

Here is the call graph for this function:

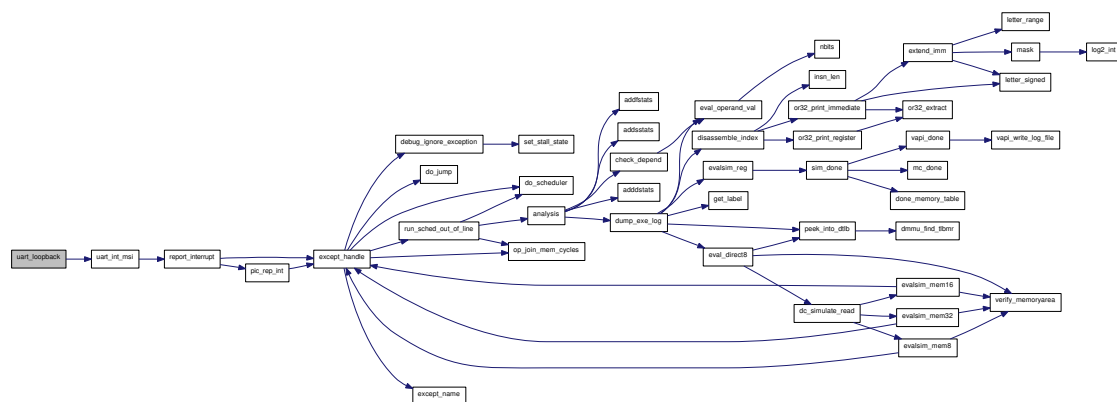


5.92.2.20 static void uart_irq (union param_val val, void * dat) [static]

5.92.2.21 static void uart_jitter (union param_val val, void * dat) [static]

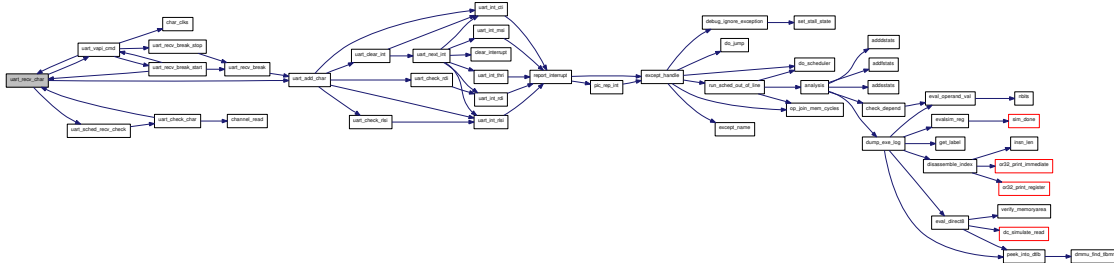
5.92.2.22 static void uart_loopback (struct dev_16450 * uart) [static]

Here is the call graph for this function:



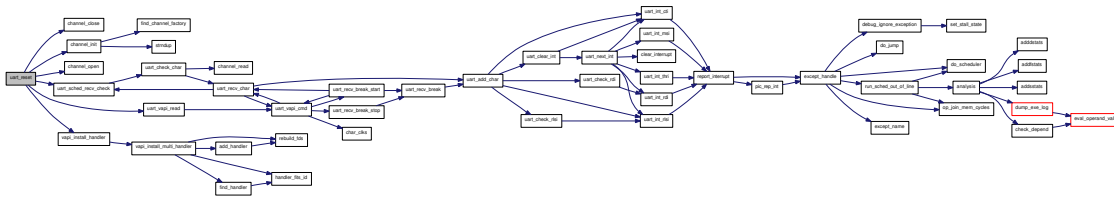
5.92.2.29 static void uart_rcv_char (void * dat) [static]

Here is the call graph for this function:



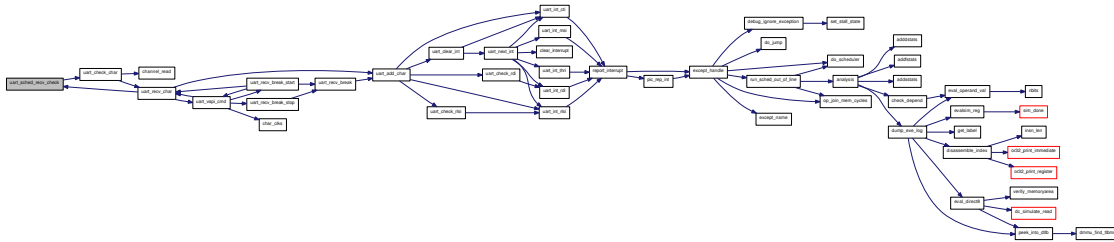
5.92.2.30 void uart_reset (void * dat)

Here is the call graph for this function:



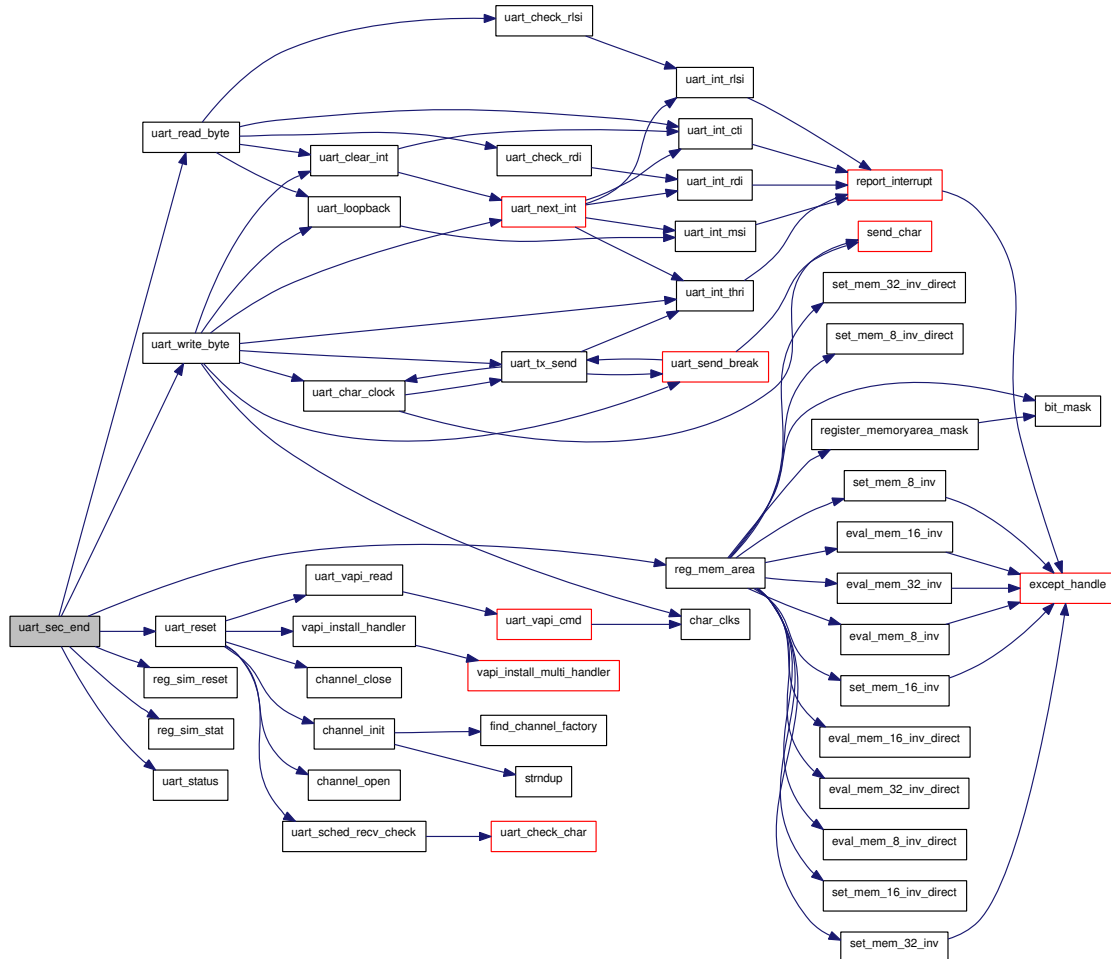
5.92.2.31 static void uart_sched_rcv_check (struct dev_16450 * uart) [static]

Here is the call graph for this function:



5.92.2.32 static void uart_sec_end (void * *dat*) [static]

Here is the call graph for this function:



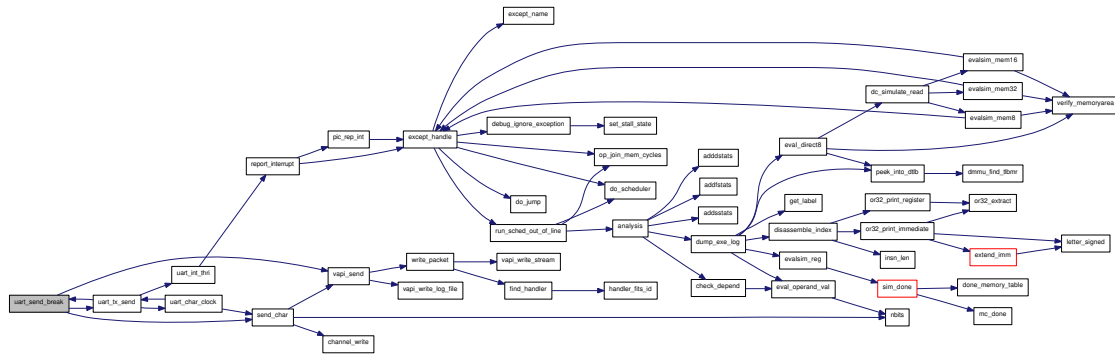
5.92.2.33 static void* uart_sec_start () [static]

Initialize a new UART configuration

ALL parameters are set explicitly to default values.

5.92.2.34 void uart_send_break (void * dat)

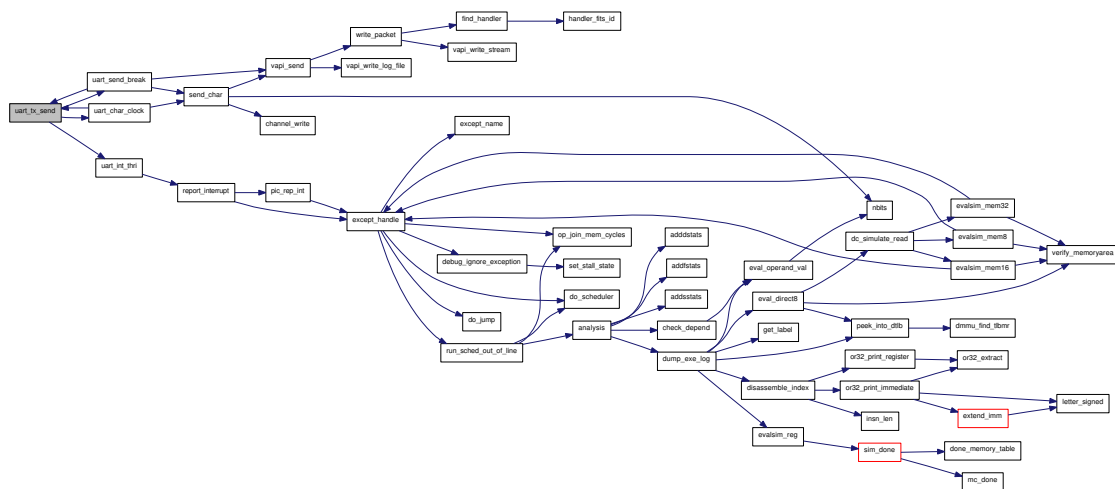
Here is the call graph for this function:



5.92.2.35 void uart_status (void * dat)

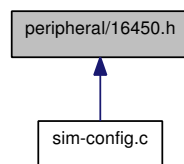
5.92.2.36 void uart_tx_send (void * dat) [static]

Here is the call graph for this function:



5.93 peripheral/16450.h File Reference

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



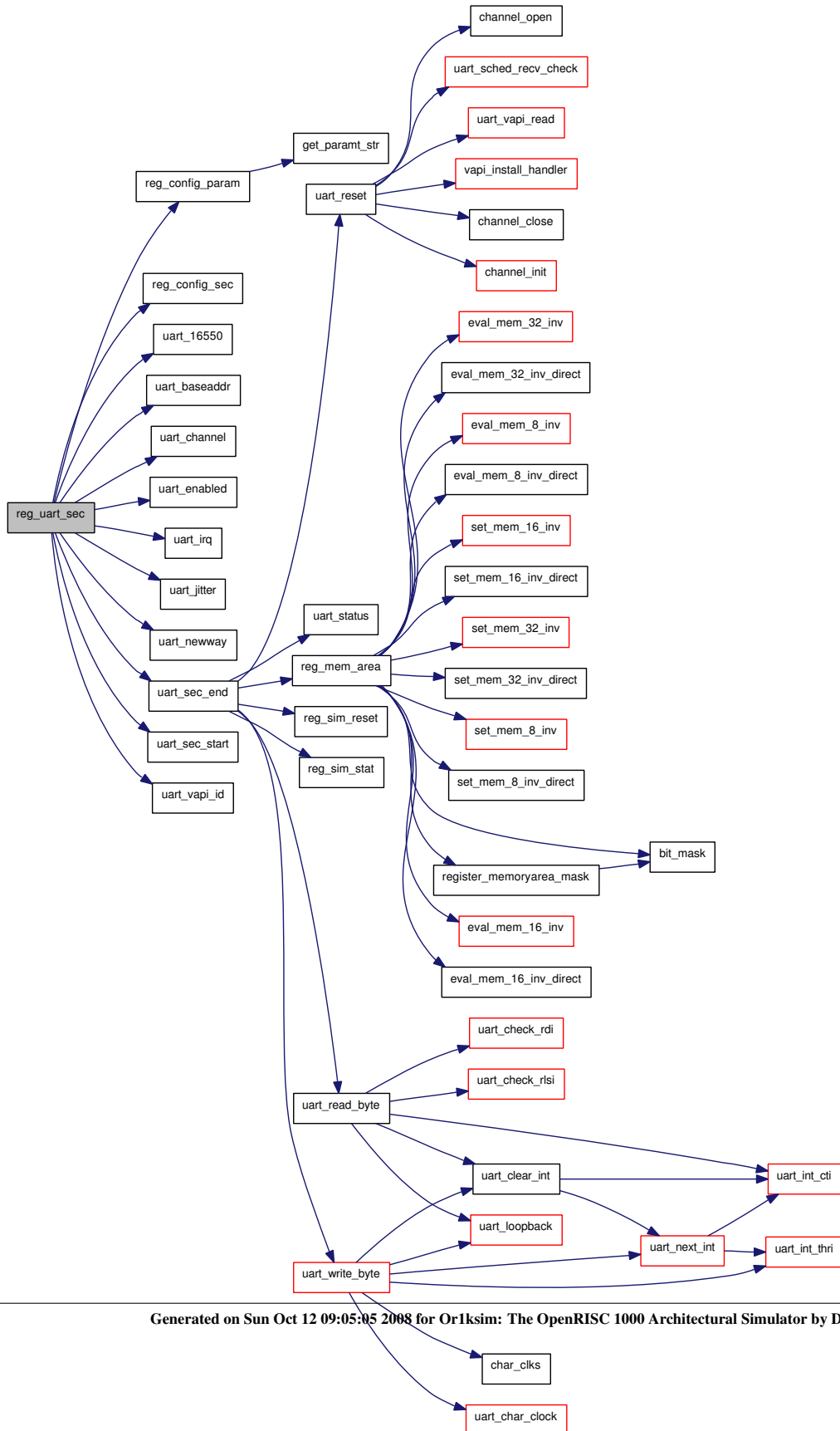
Functions

- void [uart_reset](#) ()
- void [uart_status](#) ()
- void [reg_uart_sec](#) ()

5.93.1 Function Documentation

5.93.1.1 void reg_uart_sec ()

Here is the call graph for this function:

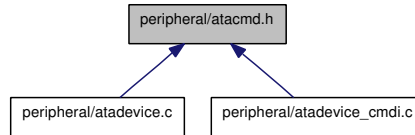


5.93.1.2 void uart_reset ()

5.93.1.3 void uart_status ()

5.94 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

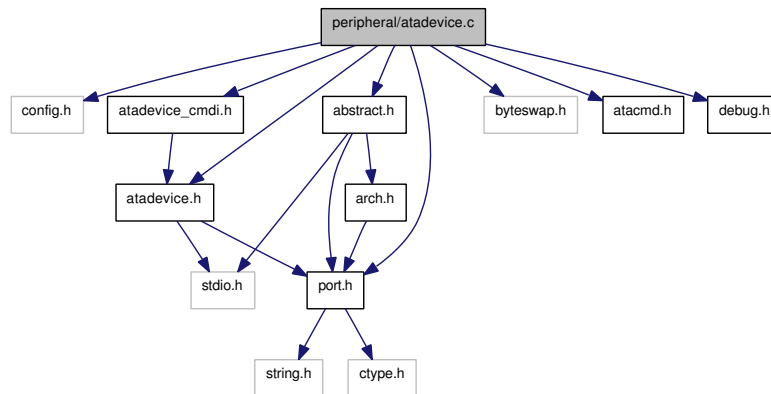
5.94.1 Define Documentation

- 5.94.1.1 `#define CFA_DISABLE_8BIT_PIO_TRANSFER_MODE 0x81`
- 5.94.1.2 `#define CFA_DISABLE_POWER_MODE1 0x8A`
- 5.94.1.3 `#define CFA_ENABLE_8BIT_PIO_TRANSFER_MODE 0x01`
- 5.94.1.4 `#define CFA_ENABLE_POWER_MODE1 0x0A`
- 5.94.1.5 `#define CFA_ERASE_SECTORS 0xC0`
- 5.94.1.6 `#define CFA_REQUEST_EXTENDED_ERROR_CODE 0x03`
- 5.94.1.7 `#define CFA_TRANSLATE_SECTOR 0x87`
- 5.94.1.8 `#define CFA_WRITE_MULTIPLE_WITHOUT_ERASE 0xCD`
- 5.94.1.9 `#define CFA_WRITE_SECTORS_WITHOUT_ERASE 0x38`
- 5.94.1.10 `#define CHECK_POWER_MODE 0xE5`
- 5.94.1.11 `#define DEVICE_RESET 0x08`
- 5.94.1.12 `#define DISABLE_ADVANCED_POWER_MANAGEMENT 0x85`
- 5.94.1.13 `#define DISABLE_MEDIA_STATUS_NOTIFICATION 0x31`
- 5.94.1.14 `#define DISABLE_POWERUP_IN_STANDBY_FEATURE_SET 0x86`
- 5.94.1.15 `#define DISABLE_READ_LOOKAHEAD 0x55`
- 5.94.1.16 `#define DISABLE_RELEASE_INTERRUPT 0xDD`
- 5.94.1.17 `#define DISABLE_REVERTING_TO_POWERON_DEFAULTS 0x66`
- 5.94.1.18 `#define DISABLE_SERVICE_INTERRUPT 0xDE`
- 5.94.1.19 `#define DISABLE_WRITE_CACHE 0x82`
- 5.94.1.20 `#define DOWNLOAD_MICROCODE 0x92`
- 5.94.1.21 `#define ENABLE_ADVANCED_POWER_MANAGEMENT 0x05`
- 5.94.1.22 `#define ENABLE_MEDIA_STATUS_NOTIFICATION 0x95`
- 5.94.1.23 `#define ENABLE_POWERUP_IN_STANDBY_FEATURE_SET 0x06`
- 5.94.1.24 `#define ENABLE_READ_LOOKAHEAD_FEATURE 0xAA`
- 5.94.1.25 `#define ENABLE_RELEASE_INTERRUPT 0x5D`
- 5.94.1.26 `#define ENABLE_REVERTING_TO_POWERON_DEFAULTS 0xCC`
- 5.94.1.27 `#define ENABLE_SERVICE_INTERRUPT 0x5E`
- 5.94.1.28 `#define ENABLE_WRITE_CACHE 0x02`
- 5.94.1.29 `#define EXECUTE_DEVICE_DIAGNOSTICS 0x90`
- 5.94.1.30 `#define FLUSH_CACHE 0xE7`

5.95 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 "debug.h"
#include "abstract.h"
```

Include dependency graph for atadevice.c:



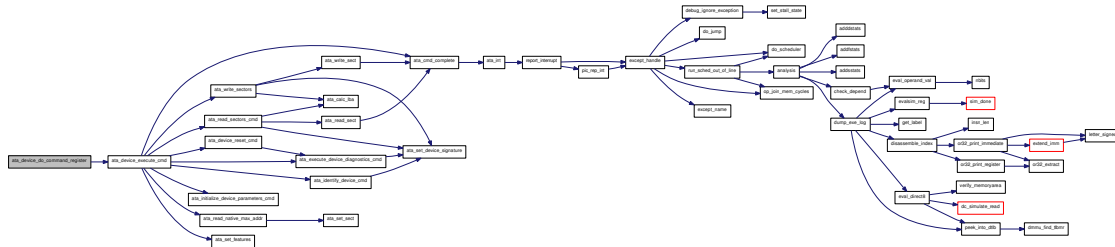
Functions

- `DEFAULT_DEBUG_CHANNEL` (ata)
- 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)
- static const char * `ata_pretty_status` (uint8_t status)
- 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)

5.95.1 Function Documentation

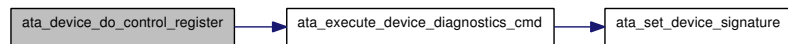
5.95.1.1 static void ata_device_do_command_register (struct ata_device * device) [static]

Here is the call graph for this function:



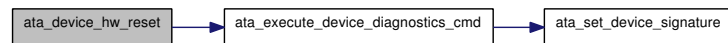
5.95.1.2 static void ata_device_do_control_register (struct ata_device * device) [static]

Here is the call graph for this function:



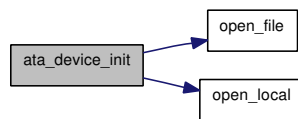
5.95.1.3 static void ata_device_hw_reset (struct ata_device * device, int reset_signal, int daspo, int pdiagi, int daspi) [static]

Here is the call graph for this function:



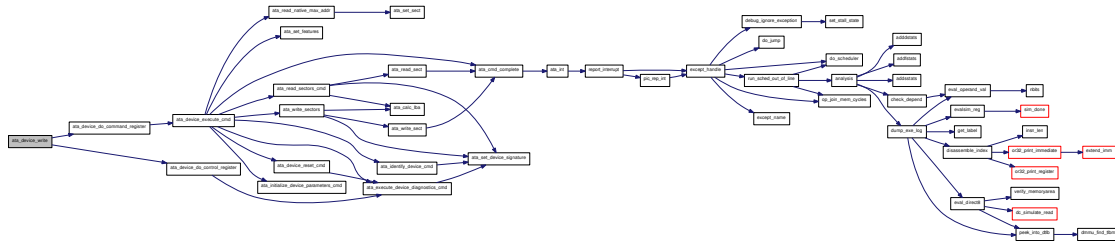
5.95.1.4 static void ata_device_init (struct ata_device * device, int dev) [static]

Here is the call graph for this function:



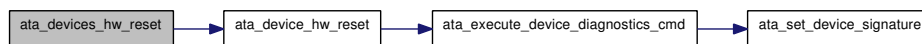
5.95.1.5 static void ata_device_write (struct ata_device * device, char adr, short value) [static]

Here is the call graph for this function:



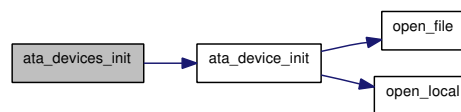
5.95.1.6 void ata_devices_hw_reset (struct ata_devices * devices, int reset_signal)

Here is the call graph for this function:



5.95.1.7 void ata_devices_init (struct ata_devices * devices)

Here is the call graph for this function:



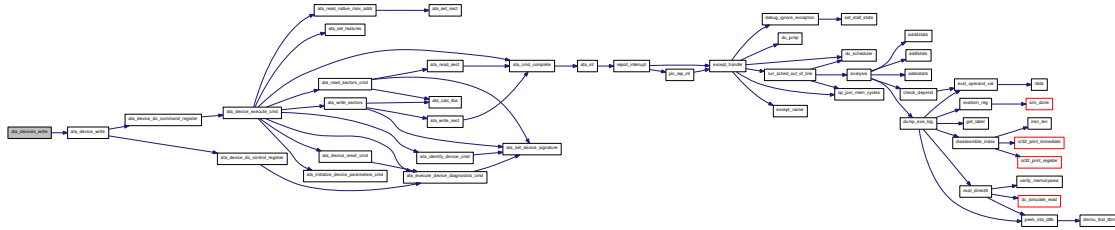
5.95.1.8 short ata_devices_read (struct ata_devices * devices, char adr)

Here is the call graph for this function:



5.95.1.9 void ata_devices_write (struct ata_devices * *devices*, char *adr*, short *value*)

Here is the call graph for this function:



5.95.1.10 static const char* ata_pretty_status (uint8_t *status*) [static]

5.95.1.11 DEFAULT_DEBUG_CHANNEL (ata)

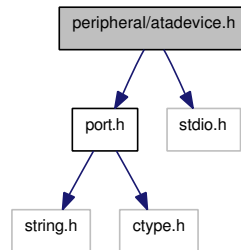
5.95.1.12 static FILE* open_file (uint32_t * *size*, const char * *filename*) [static]

5.95.1.13 static FILE* open_local (void) [static]

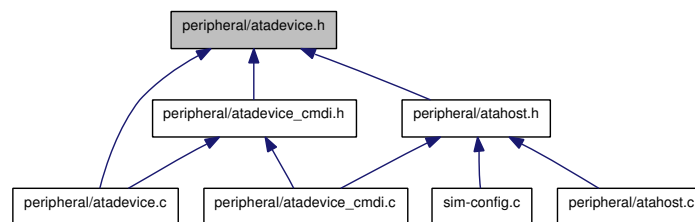
5.96 peripheral/atadevice.h File Reference

```
#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)

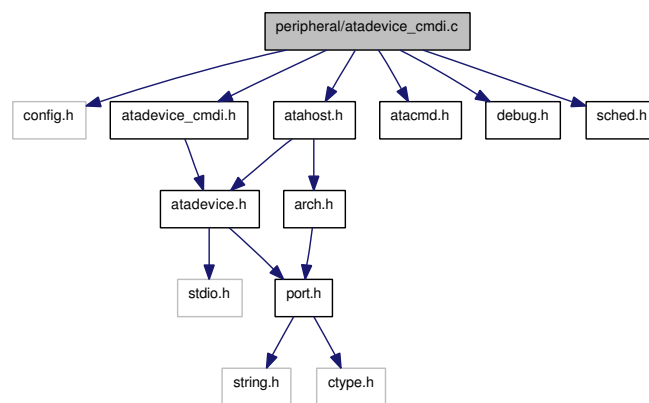
5.96.1 Define Documentation

- 5.96.1.1 **#define ATA_ASR 0x78**
- 5.96.1.2 **#define ATA_CHR 0x54**
- 5.96.1.3 **#define ATA_CLR 0x50**
- 5.96.1.4 **#define ATA_CR 0x5c**
- 5.96.1.5 **#define ATA_DA 0x7c**
- 5.96.1.6 **#define ATA_DAR_DS0 0x01**
- 5.96.1.7 **#define ATA_DAR_DS1 0x02**
- 5.96.1.8 **#define ATA_DAR_H 0x3c**
- 5.96.1.9 **#define ATA_DAR_WTG 0x40**
- 5.96.1.10 **#define ATA_DCR 0x78**
- 5.96.1.11 **#define ATA_DCR_IEN 0x02**
- 5.96.1.12 **#define ATA_DCR_RST 0x04**
- 5.96.1.13 **#define ATA_DHR 0x58**
- 5.96.1.14 **#define ATA_DHR_DEV 0x10**
- 5.96.1.15 **#define ATA_DHR_H 0x0f**
- 5.96.1.16 **#define ATA_DHR_LBA 0x40**
- 5.96.1.17 **#define ATA_DR 0x40**
- 5.96.1.18 **#define ATA_ERR 0x44**
- 5.96.1.19 **#define ATA_ERR_ABT 0x04**
- 5.96.1.20 **#define ATA_ERR_AMN 0x01**
- 5.96.1.21 **#define ATA_ERR_BBK 0x80**
- 5.96.1.22 **#define ATA_ERR_IDNF 0x10**
- 5.96.1.23 **#define ATA_ERR_TON 0x02**
- 5.96.1.24 **#define ATA_ERR_UNC 0x40**
- 5.96.1.25 **#define ATA_FR 0x44**
- 5.96.1.26 **#define ATA_SCR 0x48**
- 5.96.1.27 **#define ATA_SNR 0x4c**
- 5.96.1.28 **#define ATA_SR 0x5c**
- 5.96.1.29 **#define ATA_SR_BSY 0x80**
- 5.96.1.30 **#define ATA_SR_COR 0x04**

5.97 peripheral/atadevice_cmdi.c File Reference

```
#include "config.h"
#include "atadevice_cmdi.h"
#include "atahost.h"
#include "atacmd.h"
#include "debug.h"
#include "sched.h"
```

Include dependency graph for atadevice_cmdi.c:

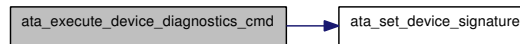


Functions

- `DEFAULT_DEBUG_CHANNEL` (ata)
- 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)

5.97.1.5 void ata_execute_device_diagnostics_cmd (struct ata_device * device)

Here is the call graph for this function:



5.97.1.6 static void ata_identify_device_cmd (struct ata_device * device) [static]

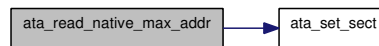
Here is the call graph for this function:



5.97.1.7 static void ata_initialize_device_parameters_cmd (struct ata_device * device) [static]

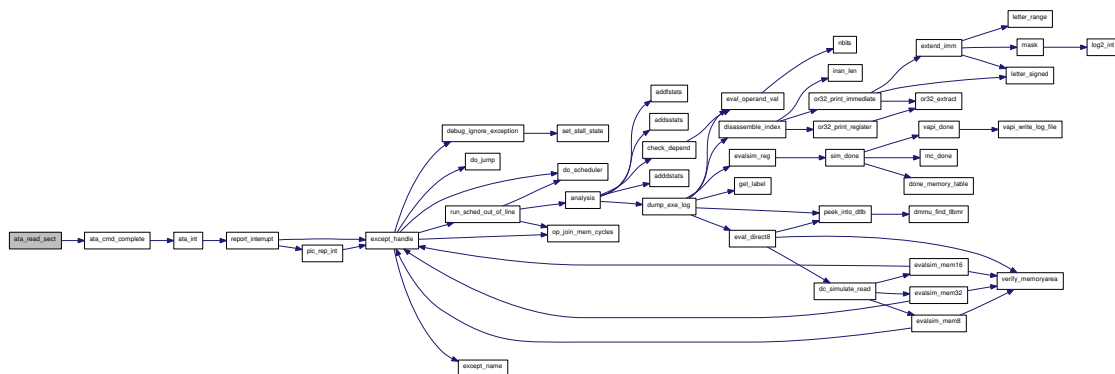
5.97.1.8 static void ata_read_native_max_addr (struct ata_device * dev) [static]

Here is the call graph for this function:



5.97.1.9 static void ata_read_sect (struct ata_device * dev) [static]

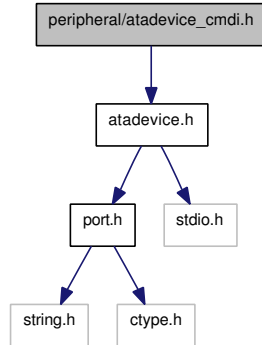
Here is the call graph for this function:



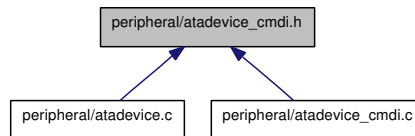
5.98 peripheral/atadevice_cmdi.h File Reference

```
#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)

5.98.1 Define Documentation

- 5.98.1.1 `#define BYTES_PER_SECTOR 512`
- 5.98.1.2 `#define MIN_MWDMA_CYCLE_TIME 100`
- 5.98.1.3 `#define MIN_PIO_CYCLE_TIME_IORDY 100`
- 5.98.1.4 `#define MIN_PIO_CYCLE_TIME_NO_IORDY 100`
- 5.98.1.5 `#define QUEUE_DEPTH 0`
- 5.98.1.6 `#define RECOMMENDED_MWDMA_CYCLE_TIME 100`
- 5.98.1.7 `#define SET_FEATURES_REQUIRED_AFTER_POWER_UP 0`
- 5.98.1.8 `#define SUPPORT_APM 0`
- 5.98.1.9 `#define SUPPORT_CFA 0`
- 5.98.1.10 `#define SUPPORT_DEVICE_RESET_CMD 1`
- 5.98.1.11 `#define SUPPORT_DOWNLOAD_MICROCODE 0`
- 5.98.1.12 `#define SUPPORT_HOST_PROTECTED_AREA 0`
- 5.98.1.13 `#define SUPPORT_LOOKAHEAD 0`
- 5.98.1.14 `#define SUPPORT_NOP_CMD 0`
- 5.98.1.15 `#define SUPPORT_POWER_MANAGEMENT 0`
- 5.98.1.16 `#define SUPPORT_POWER_UP_IN_STANDBY_MODE 0`
- 5.98.1.17 `#define SUPPORT_READ_BUFFER_CMD 0`
- 5.98.1.18 `#define SUPPORT_READ_WRITE_DMA_QUEUED 0`
- 5.98.1.19 `#define SUPPORT_RELEASE_INTERRUPT 0`
- 5.98.1.20 `#define SUPPORT_REMOVABLE_MEDIA 0`
- 5.98.1.21 `#define SUPPORT_REMOVABLE_MEDIA_NOTIFICATION 0`
- 5.98.1.22 `#define SUPPORT_SECURITY_MODE 0`
- 5.98.1.23 `#define SUPPORT_SERVICE_INTERRUPT 0`
- 5.98.1.24 `#define SUPPORT_SET_MAX 0`
- 5.98.1.25 `#define SUPPORT_SMART 0`
- 5.98.1.26 `#define SUPPORT_WRITE_BUFFER_CMD 0`
- 5.98.1.27 `#define SUPPORT_WRITE_CACHE 0`

Generated on Sun Oct 12 09:05:03 2008 for OpenMSP: The OpenRISC 1000 Architectural Simulator by Doxygen

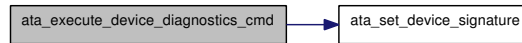
5.98.2 Function Documentation

- 5.98.2.1 `int ata_device_execute_cmd (struct ata_device * device)`

Here is the call graph for this function:

5.98.2.2 void ata_execute_device_diagnostics_cmd (struct ata_device * device)

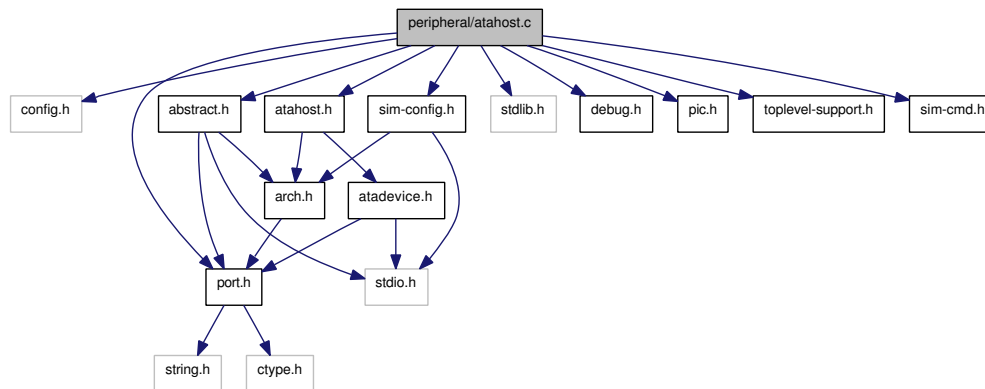
Here is the call graph for this function:



5.99 peripheral/atahost.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "atahost.h"
#include "sim-config.h"
#include "debug.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

- [DEFAULT_DEBUG_CHANNEL](#) (ata)
- 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](#)

5.99.1 Define Documentation

5.99.1.1 `#define DMA_MODE0_TD 21`

5.99.1.2 `#define DMA_MODE0_TEOC 21`

5.99.1.3 `#define DMA_MODE0_TM 4`

5.99.1.4 `#define PIO_MODE0_T1 6`

5.99.1.5 `#define PIO_MODE0_T2 28`

5.99.1.6 `#define PIO_MODE0_T4 2`

5.99.1.7 `#define PIO_MODE0_TEOC 23`

5.99.2 Function Documentation

5.99.2.1 `static void ata_baseaddr (union param_val val, void * dat) [static]`

5.99.2.2 `static void ata_dev_id (union param_val val, void * dat) [static]`

5.99.2.3 `static void ata_dma_mode0_td (union param_val val, void * dat) [static]`

5.99.2.4 `static void ata_dma_mode0_tioc (union param_val val, void * dat) [static]`

5.99.2.5 `static void ata_dma_mode0_tm (union param_val val, void * dat) [static]`

5.99.2.6 `static void ata_enabled (union param_val val, void * dat) [static]`

5.99.2.7 `static void ata_enddevice (union param_val val, void * dat) [static]`

5.99.2.8 `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

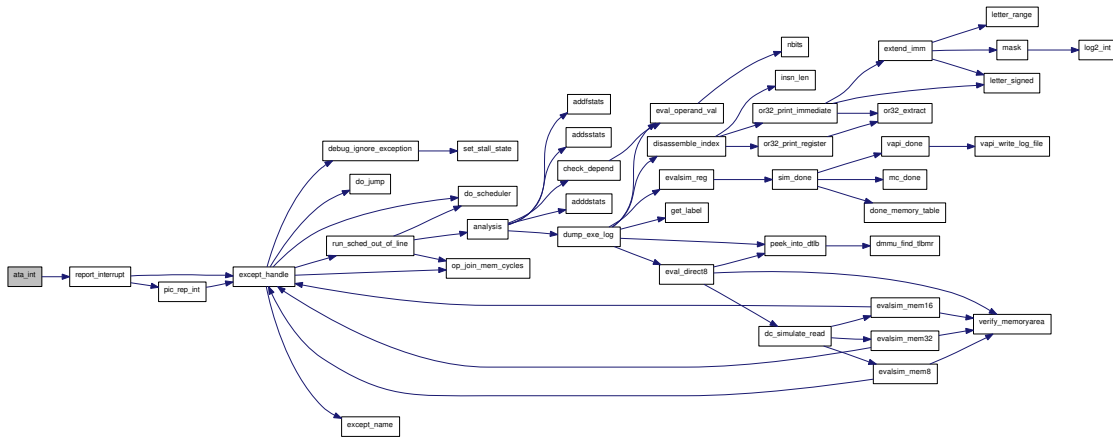
← *dat* The [config](#) data structure

5.99.2.9 `static void ata_firmware (union param_val val, void * dat)` [static]

5.99.2.10 `static void ata_heads (union param_val val, void * dat)` [static]

5.99.2.11 `void ata_int (void * dat)`

Here is the call graph for this function:



5.99.2.12 `static void ata_irq (union param_val val, void * dat)` [static]

5.99.2.13 `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

5.99.2.14 `static void ata_packet (union param_val val, void * dat)` [static]

5.99.2.15 `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

5.99.2.16 `static void ata_pio_mode0_t1 (union param_val val, void * dat)` [static]

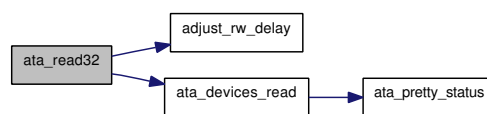
5.99.2.17 `static void ata_pio_mode0_t2 (union param_val val, void * dat)` [static]

5.99.2.18 `static void ata_pio_mode0_t4 (union param_val val, void * dat)` [static]

5.99.2.19 `static void ata_pio_mode0_tec (union param_val val, void * dat)` [static]

5.99.2.20 `static uint32_t ata_read32 (oraddr_t addr, void * dat)` [static]

Here is the call graph for this function:



5.99.2.21 `static void ata_reset (void * dat)` [static]

Here is the call graph for this function:



5.99.2.22 `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.

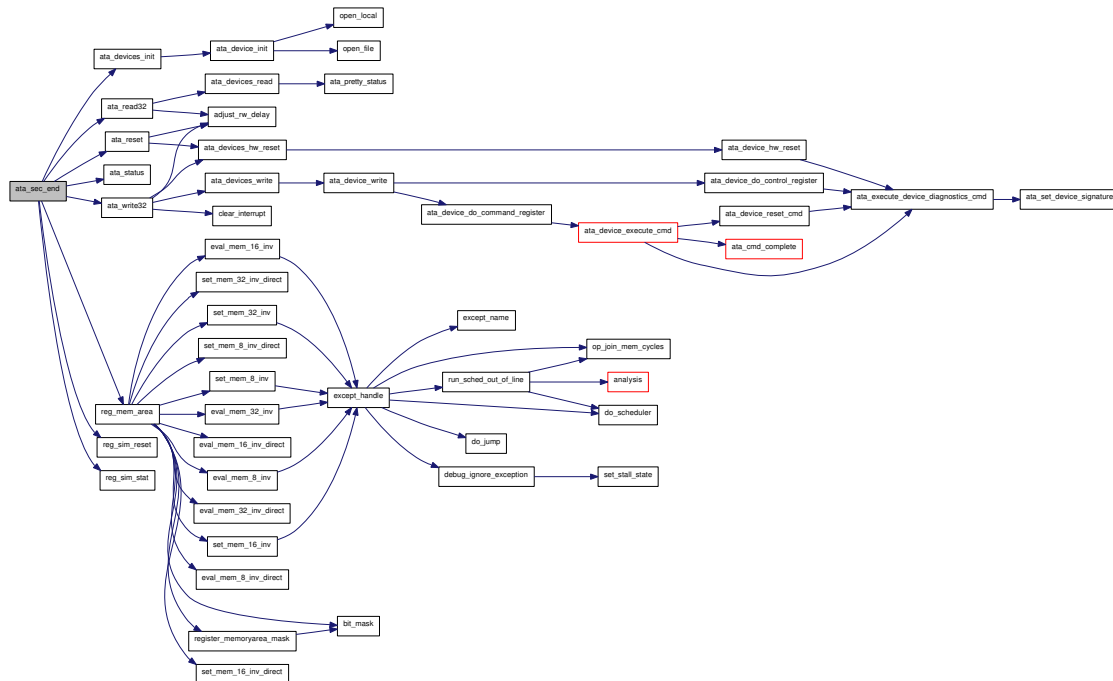
Parameters:

← *val* The value to use

← *dat* The `config` data structure

5.99.2.23 static void ata_sec_end (void * dat) [static]

Here is the call graph for this function:



5.99.2.24 static void* ata_sec_start (void) [static]

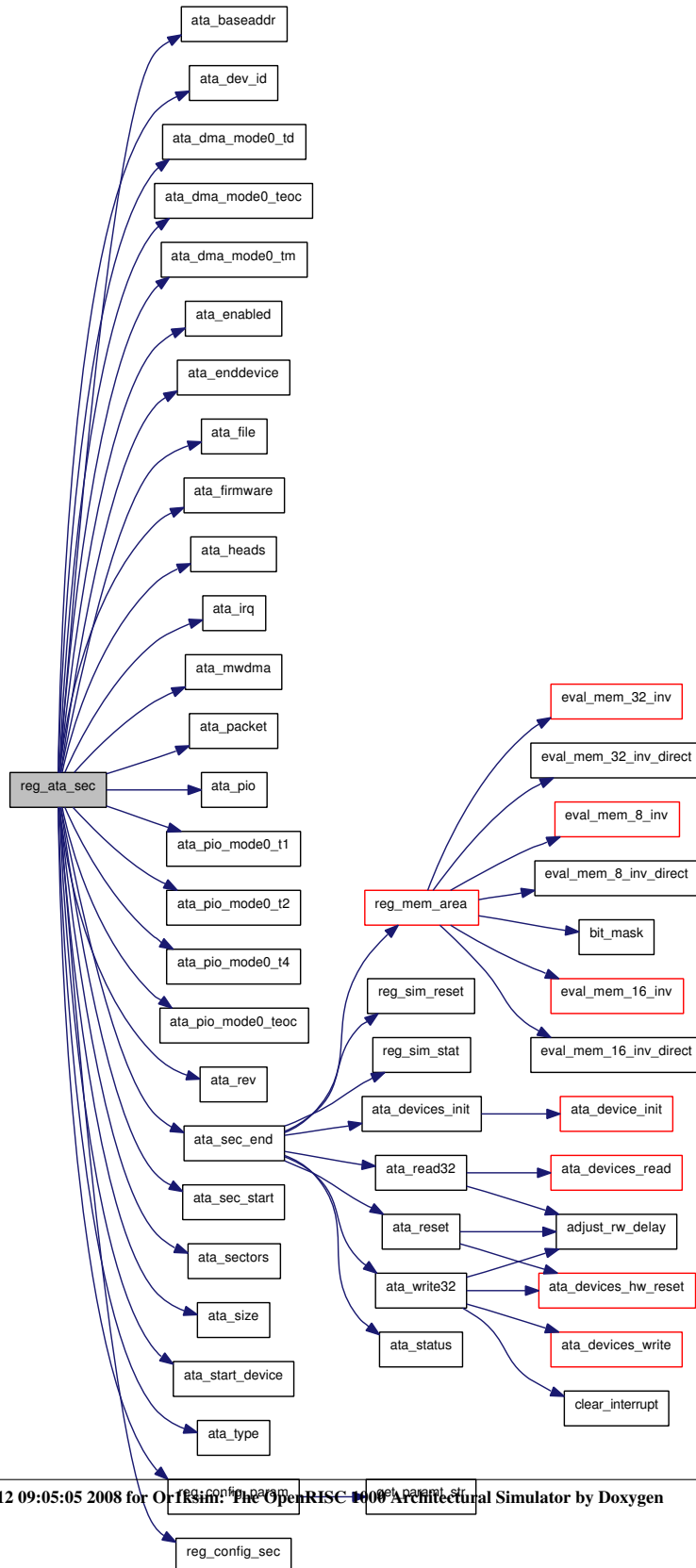
Initialize a new ATA configuration

ALL parameters are set explicitly to default values.

5.99.2.31 DEFAULT_DEBUG_CHANNEL (ata)

5.99.2.32 void reg_ata_sec ()

Here is the call graph for this function:



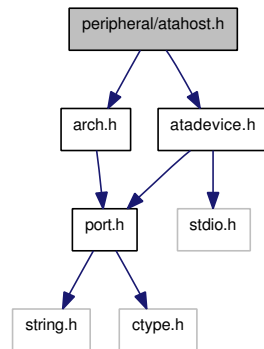
5.99.3 Variable Documentation

5.99.3.1 `unsigned int conf_dev` `[static]`

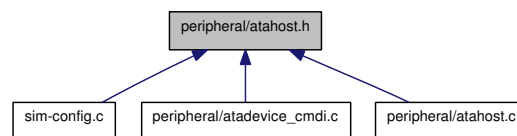
5.100 peripheral/atahost.h File Reference

```
#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` ()

5.100.1 Define Documentation

5.100.1.1 #define ATA_ADDR_SPACE 0x80

5.100.1.2 #define ATA_BELEC0 (1<< 8)

5.100.1.3 #define ATA_BELEC1 (1<< 9)

5.100.1.4 #define ATA_CTRL 0x00

5.100.1.5 #define ATA_DEVID 0xf0000000

5.100.1.6 #define ata_dma_delay(dmareg) ((((dmareg >> ATA_TD) & 0xff) +1) + (((pioreg >> ATA_TM) & 0xff) +1) +1)

5.100.1.7 #define ATA_DMA_EN (0<<15)

5.100.1.8 #define ATA_DMA_RD (0<<14)

5.100.1.9 #define ATA_DMA_TIP (1<<15)

5.100.1.10 #define ATA_DMA_WR (1<<14)

5.100.1.11 #define ATA_DMARQ (1<< 8)

5.100.1.12 #define ATA_DRBE (1<<10)

5.100.1.13 #define ATA_DTBF (1<< 9)

5.100.1.14 #define ATA_DTR0 0x14

5.100.1.15 #define ATA_DTR1 0x18

5.100.1.16 #define ATA_FTE0 (1<< 5)

5.100.1.17 #define ATA_FTE1 (1<< 6)

5.100.1.18 #define ATA_IDE_EN (1<< 7)

5.100.1.19 #define ATA_IDEIS (1<< 0)

5.100.1.20 #define ATA_IORDY (1<< 1)

5.100.1.21 #define ATA_IORDY_FTE0 (1<< 2)

5.100.1.22 #define ATA_IORDY_FTE1 (1<< 3)

5.100.1.23 #define ATA_PCTR 0x08

5.100.1.24 #define ATA_PFTR0 0x0c

5.100.1.25 #define ATA_PFTR1 0x10

5.100.1.26 #define ata_pio_delay(pioreg) ((((pioreg >> ATA_T1) & 0xff) +1) + (((pioreg >> ATA_T2) & 0xff) +1) + (((pioreg >> ATA_T3) & 0xff) +1) + (((pioreg >> ATA_T4) & 0xff) +1) +1)

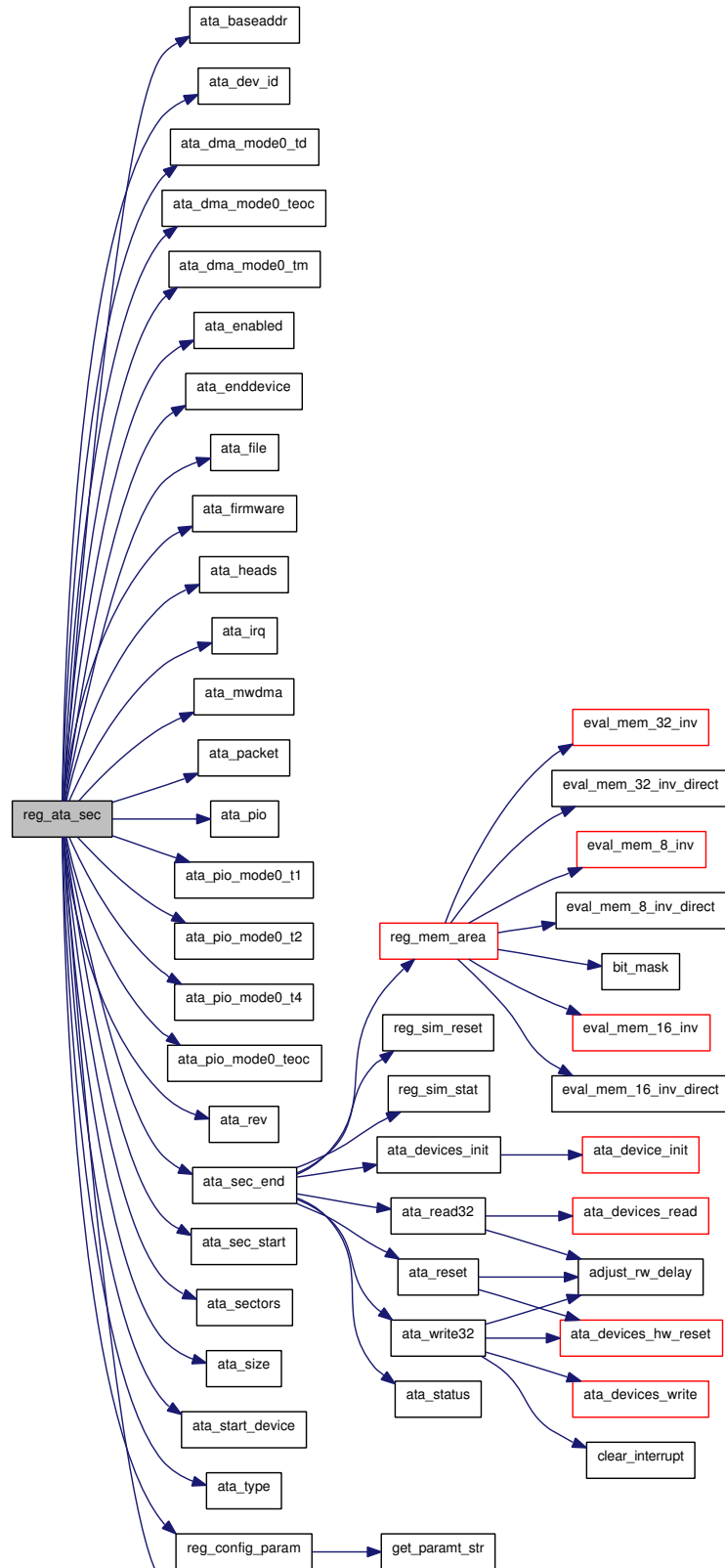
5.100.1.27 #define ATA_PIO_TIP (1<< 7)

5.100.1.28 #define ATA_PWPP (1<< 4)

5.100.1.29 #define ATA_PWPPF (1<< 6)

5.100.2.2 void reg_ata_sec ()

Here is the call graph for this function:



5.101 peripheral/atahost_define.h File Reference

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

5.101.1 Define Documentation

5.101.1.1 #define DMA_MODE0_TD 21

5.101.1.2 #define DMA_MODE0_TEOC 21

5.101.1.3 #define DMA_MODE0_TM 4

5.101.1.4 #define PIO_MODE0_T1 6

5.101.1.5 #define PIO_MODE0_T2 28

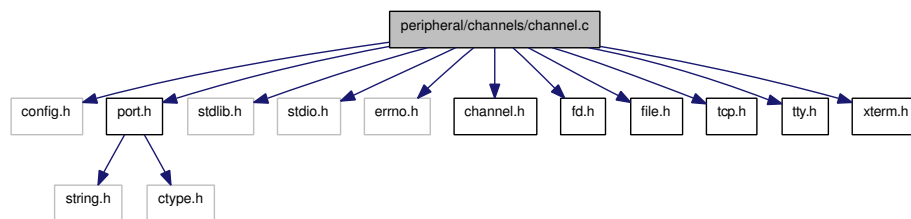
5.101.1.6 #define PIO_MODE0_T4 2

5.101.1.7 #define PIO_MODE0_TEOC 23

5.102 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

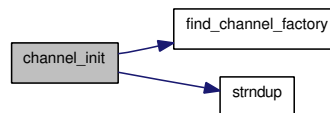
- static struct [channel_factory](#) [preloaded](#) []
- static struct [channel_factory](#) * [head](#) = &[preloaded](#)[0]

5.102.1 Function Documentation

5.102.1.1 void `channel_close` (`struct channel * channel`)

5.102.1.2 `struct channel*` `channel_init` (`const char * descriptor`) [read]

Here is the call graph for this function:



5.102.1.3 int `channel_open` (`struct channel * channel`)

5.102.1.4 int `channel_read` (`struct channel * channel`, `char * buffer`, `int size`)

5.102.1.5 int `channel_write` (`struct channel * channel`, `const char * buffer`, `int size`)

5.102.1.6 `static struct channel_factory *` `find_channel_factory` (`const char * name`) [static, read]

5.102.2 Variable Documentation

5.102.2.1 `struct channel_factory*` `head = &preloaded[0]` [static]

5.102.2.2 `struct channel_factory` `preloaded[]` [static]

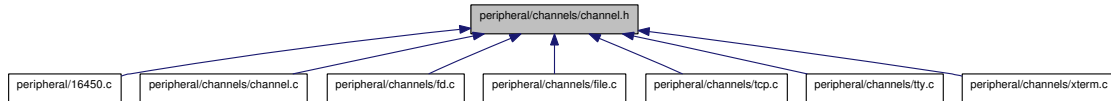
Initial value:

```

{
  {"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}
}
  
```

5.103 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

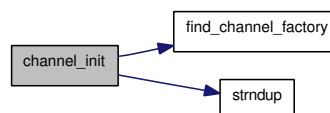
- struct [channel](#) * [channel_init](#) (const char *descriptor)
- int [channel_open](#) (struct [channel](#) *channel)
- int [channel_read](#) (struct [channel](#) *channel, char *buffer, int size)
- int [channel_write](#) (struct [channel](#) *channel, const char *buffer, int size)
- void [channel_close](#) (struct [channel](#) *channel)

5.103.1 Function Documentation

5.103.1.1 void [channel_close](#) (struct [channel](#) * *channel*)

5.103.1.2 struct [channel](#)* [channel_init](#) (const char * *descriptor*) [read]

Here is the call graph for this function:



5.103.1.3 int [channel_open](#) (struct [channel](#) * *channel*)

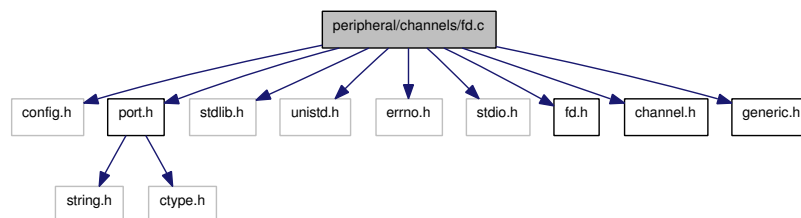
5.103.1.4 int [channel_read](#) (struct [channel](#) * *channel*, char * *buffer*, int *size*)

5.103.1.5 int [channel_write](#) (struct [channel](#) * *channel*, const char * *buffer*, int *size*)

5.104 peripheral/channels/fd.c File Reference

```
#include "config.h"  
#include "port.h"  
#include <stdlib.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](#)

5.104.1 Function Documentation

5.104.1.1 `static void * fd_init (const char * args) [static]`

5.104.1.2 `static int fd_isok (void * data) [static]`

5.104.1.3 `int fd_read (void * data, char * buffer, int size)`

5.104.1.4 `static char * fd_status (void * data) [static]`

Here is the call graph for this function:



5.104.1.5 `static int fd_status_fd (int fd, char * str, int size) [static]`

5.104.1.6 `int fd_write (void * data, const char * buffer, int size)`

5.104.2 Variable Documentation

5.104.2.1 `struct channel_ops fd_channel_ops`

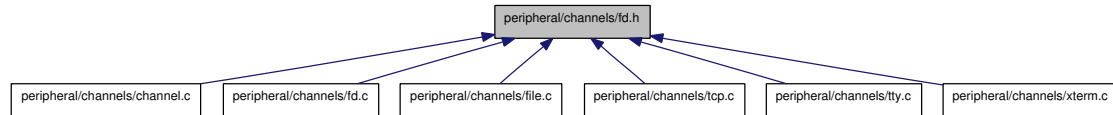
Initial value:

```
{
    .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](#)

5.105 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](#)

5.105.1 Function Documentation

5.105.1.1 int [fd_read](#) (void * *data*, char * *buffer*, int *size*)

5.105.1.2 int [fd_write](#) (void * *data*, const char * *buffer*, int *size*)

5.105.2 Variable Documentation

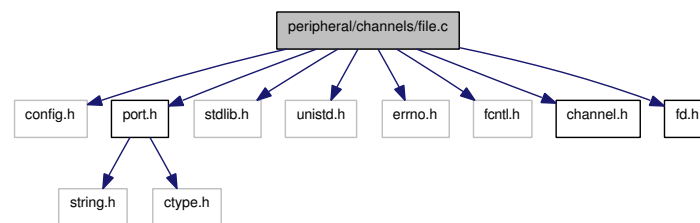
5.105.2.1 struct [channel_ops](#) [fd_channel_ops](#)

Global data structure representing the operations for communicating through a file descriptor [channel](#)

5.106 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](#)

5.106.1 Function Documentation

5.106.1.1 `static void file_close (void * data)` [static]

5.106.1.2 `static void file_free (void * data)` [static]

5.106.1.3 `static void * file_init (const char * args)` [static]

Here is the call graph for this function:



5.106.1.4 `static int file_open (void * data)` [static]

5.106.2 Variable Documentation

5.106.2.1 `struct channel_ops file_channel_ops`

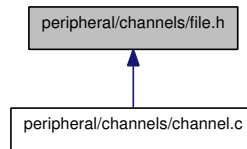
Initial value:

```
{
    .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](#)

5.107 peripheral/channels/file.h File Reference

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



Variables

- struct [channel_ops](#) [file_channel_ops](#)

5.107.1 Variable Documentation

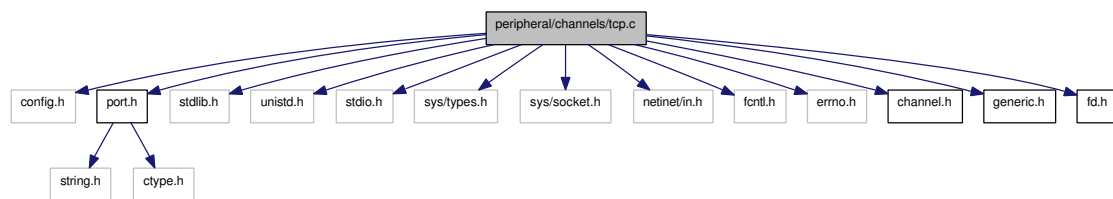
5.107.1.1 struct [channel_ops](#) [file_channel_ops](#)

Data structure with all the operations for communicating with a file [channel](#)

5.108 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](#)

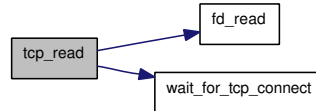
5.108.1 Function Documentation

5.108.1.1 `static void * tcp_init (const char * input)` [static]

5.108.1.2 `static int tcp_open (void * data)` [static]

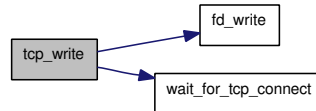
5.108.1.3 `static int tcp_read (void * data, char * buffer, int size)` [static]

Here is the call graph for this function:



5.108.1.4 `static int tcp_write (void * data, const char * buffer, int size)` [static]

Here is the call graph for this function:



5.108.1.5 `static int wait_for_tcp_connect (struct tcp_channel * channel)` [static]

5.108.2 Variable Documentation

5.108.2.1 `struct channel_ops tcp_channel_ops`

Initial value:

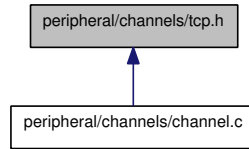
```

{
    .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](#)

5.109 peripheral/channels/tcp.h File Reference

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



Variables

- struct [channel_ops](#) [tcp_channel_ops](#)

5.109.1 Variable Documentation

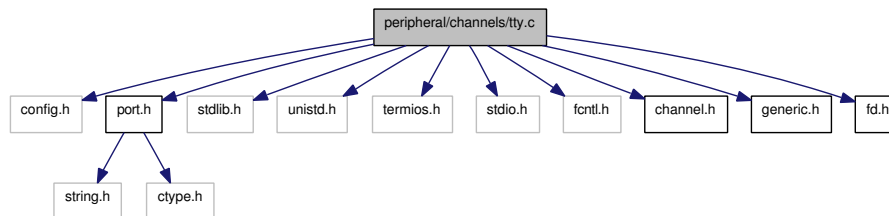
5.109.1.1 struct [channel_ops](#) [tcp_channel_ops](#)

Data structure holding all the operations for a TCP/IP [channel](#)

5.110 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](#)

5.110.1 Define Documentation

5.110.1.1 `#define DEFAULT_BAUD B19200`

5.110.1.2 `#define DEFAULT_TTY_DEVICE "/dev/ttyS0"`

5.110.2 Function Documentation

5.110.2.1 `static int parse_baud (char * baud_string)` [static]

5.110.2.2 `static void * tty_init (const char * input)` [static]

Here is the call graph for this function:



5.110.2.3 `static int tty_open (void * data)` [static]

5.110.3 Variable Documentation

5.110.3.1 `struct { ... } baud_table[]` [static]

Table of Baud rates

5.110.3.2 `char* name`

5.110.3.3 `struct channel_ops tty_channel_ops`

Initial value:

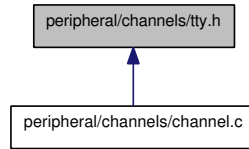
```
{
    .init = tty_init,
    .open = tty_open,
    .close = generic_close,
    .read = fd_read,
    .write = fd_write,
    .free = generic_free,
}
```

Global data structure representing the operations on a TTY [channel](#)

5.110.3.4 `int value`

5.111 peripheral/channels/tty.h File Reference

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



Variables

- struct [channel_ops](#) [tty_channel_ops](#)

5.111.1 Variable Documentation

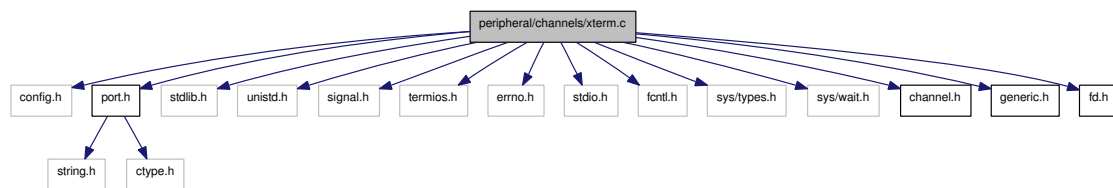
5.111.1.1 struct [channel_ops](#) [tty_channel_ops](#)

Global data structure representing the operations on a TTY [channel](#)

5.112 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](#)

5.112.1 Define Documentation

5.112.1.1 #define MAX_XTERM_ARGS 100

5.112.2 Function Documentation

5.112.2.1 static char* basename (const char **filename*) [static]

5.112.2.2 static void xterm_close (void **data*) [static]

5.112.2.3 static void * xterm_init (const char **input*) [static]

5.112.2.4 static int xterm_open (void **data*) [static]

Here is the call graph for this function:



5.112.3 Variable Documentation

5.112.3.1 struct channel_ops xterm_channel_ops

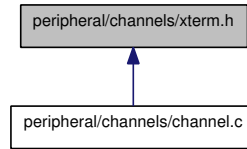
Initial value:

```
{
    .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

5.113 peripheral/channels/xterm.h File Reference

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



Variables

- struct [channel_ops](#) [xterm_channel_ops](#)

5.113.1 Variable Documentation

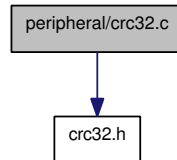
5.113.1.1 struct [channel_ops](#) [xterm_channel_ops](#)

Global data structure with the xterm interface functions

5.114 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)

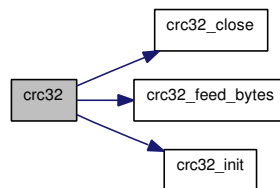
Variables

- static unsigned long [crc32_table](#) [256]

5.114.1 Function Documentation

5.114.1.1 unsigned long [crc32](#) (const void * buf, unsigned len)

Here is the call graph for this function:



5.114.1.2 void [crc32_close](#) (unsigned long * value)

5.114.1.3 void [crc32_feed_bytes](#) (unsigned long * value, const void * buf, unsigned len)

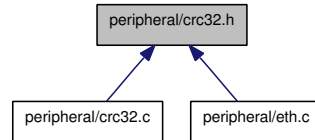
5.114.1.4 void [crc32_init](#) (unsigned long * value)

5.114.2 Variable Documentation

5.114.2.1 unsigned long [crc32_table](#)[256] [static]

5.115 peripheral/crc32.h File Reference

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



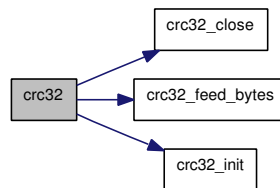
Functions

- 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)

5.115.1 Function Documentation

5.115.1.1 unsigned long [crc32](#) (const void * buf, unsigned len)

Here is the call graph for this function:



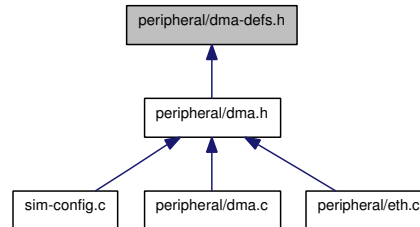
5.115.1.2 void [crc32_close](#) (unsigned long * value)

5.115.1.3 void [crc32_feed_bytes](#) (unsigned long * value, const void * buf, unsigned len)

5.115.1.4 void [crc32_init](#) (unsigned long * value)

5.116 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

5.116.1 Define Documentation

5.116.1.1 `#define DMA_ADDR_SPACE 0x400`

5.116.1.2 `#define DMA_CH_A0 0x08`

5.116.1.3 `#define DMA_CH_A0_ADDR_OFFSET 2`

5.116.1.4 `#define DMA_CH_A0_ADDR_WIDTH 30`

5.116.1.5 `#define DMA_CH_A1 0x10`

5.116.1.6 `#define DMA_CH_A1_ADDR_OFFSET 2`

5.116.1.7 `#define DMA_CH_A1_ADDR_WIDTH 30`

5.116.1.8 `#define DMA_CH_AM0 0x0C`

5.116.1.9 `#define DMA_CH_AM0_MASK_OFFSET 4`

5.116.1.10 `#define DMA_CH_AM0_MASK_WIDTH 28`

5.116.1.11 `#define DMA_CH_AM1 0x14`

5.116.1.12 `#define DMA_CH_AM1_MASK_OFFSET 4`

5.116.1.13 `#define DMA_CH_AM1_MASK_WIDTH 28`

5.116.1.14 `#define DMA_CH_BASE 0x20`

Offset of first [channel](#) registers


```

5.116.1.15 #define DMA_CH_CSR 0x00
5.116.1.16 #define DMA_CH_CSR_ARS_OFFSET 6
5.116.1.17 #define DMA_CH_CSR_BUSY_OFFSET 10
5.116.1.18 #define DMA_CH_CSR_CH_EN_OFFSET 0
5.116.1.19 #define DMA_CH_CSR_DONE_OFFSET 11
5.116.1.20 #define DMA_CH_CSR_DST_SEL_OFFSET 1
5.116.1.21 #define DMA_CH_CSR_ERR_OFFSET 12
5.116.1.22 #define DMA_CH_CSR_INC_DST_OFFSET 3
5.116.1.23 #define DMA_CH_CSR_INC_SRC_OFFSET 4
5.116.1.24 #define DMA_CH_CSR_INE_CHK_DONE_OFFSET 19
5.116.1.25 #define DMA_CH_CSR_INE_DONE_OFFSET 18
5.116.1.26 #define DMA_CH_CSR_INE_ERR_OFFSET 17
5.116.1.27 #define DMA_CH_CSR_INT_CHUNK_DONE_OFFSET 22
5.116.1.28 #define DMA_CH_CSR_INT_DONE_OFFSET 21
5.116.1.29 #define DMA_CH_CSR_INT_ERR_OFFSET 20
5.116.1.30 #define DMA_CH_CSR_MODE_OFFSET 5
5.116.1.31 #define DMA_CH_CSR_PRIORITY_OFFSET 13
5.116.1.32 #define DMA_CH_CSR_PRIORITY_WIDTH 3
5.116.1.33 #define DMA_CH_CSR_RESERVED_OFFSET 23
5.116.1.34 #define DMA_CH_CSR_RESERVED_WIDTH 9
5.116.1.35 #define DMA_CH_CSR_REST_EN_OFFSET 16
5.116.1.36 #define DMA_CH_CSR_SRC_SEL_OFFSET 2
5.116.1.37 #define DMA_CH_CSR_STOP_OFFSET 9
5.116.1.38 #define DMA_CH_CSR_SZ_WB_OFFSET 8
5.116.1.39 #define DMA_CH_CSR_USE_ED_OFFSET 7
5.116.1.40 #define DMA_CH_CSR_WRITE_MASK 0x000FE3FF
5.116.1.41 #define DMA_CH_DESC 0x18
5.116.1.42 #define DMA_CH_DESC_ADDR_OFFSET 2
5.116.1.43 #define DMA_CH_DESC_ADDR_WIDTH 30
5.116.1.44 #define DMA_CH_SIZE 0x20

```

5.116.1.45 **#define DMA_CH_SWPTR 0x1C**

5.116.1.46 **#define DMA_CH_SWPTR_EN_OFFSET 31**

5.116.1.47 **#define DMA_CH_SWPTR_PTR_OFFSET 2**

5.116.1.48 **#define DMA_CH_SWPTR_PTR_WIDTH 29**

5.116.1.49 **#define DMA_CH_SZ 0x04**

5.116.1.50 **#define DMA_CH_SZ_CHK_SZ_OFFSET 16**

5.116.1.51 **#define DMA_CH_SZ_CHK_SZ_WIDTH 9**

5.116.1.52 **#define DMA_CH_SZ_TOT_SZ_OFFSET 0**

5.116.1.53 **#define DMA_CH_SZ_TOT_SZ_WIDTH 12**

5.116.1.54 **#define DMA_CSR 0x00**

5.116.1.55 **#define DMA_CSR_PAUSE_OFFSET 0**

5.116.1.56 **#define DMA_DESC_ADR0 0x04**

5.116.1.57 **#define DMA_DESC_ADR1 0x08**

5.116.1.58 **#define DMA_DESC_CSR 0x00**

5.116.1.59 **#define DMA_DESC_CSR_DST_SEL_OFFSET 16**

5.116.1.60 **#define DMA_DESC_CSR_EOL_OFFSET 20**

5.116.1.61 **#define DMA_DESC_CSR_INC_DST_OFFSET 18**

5.116.1.62 **#define DMA_DESC_CSR_INC_SRC_OFFSET 19**

5.116.1.63 **#define DMA_DESC_CSR_SRC_SEL_OFFSET 17**

5.116.1.64 **#define DMA_DESC_CSR_TOT_SZ_OFFSET 0**

5.116.1.65 **#define DMA_DESC_CSR_TOT_SZ_WIDTH 12**

5.116.1.66 **#define DMA_DESC_NEXT 0x0C**

5.116.1.67 **#define DMA_INT_MSK_A 0x04**

5.116.1.68 **#define DMA_INT_MSK_B 0x08**

5.116.1.69 **#define DMA_INT_SRC_A 0x0C**

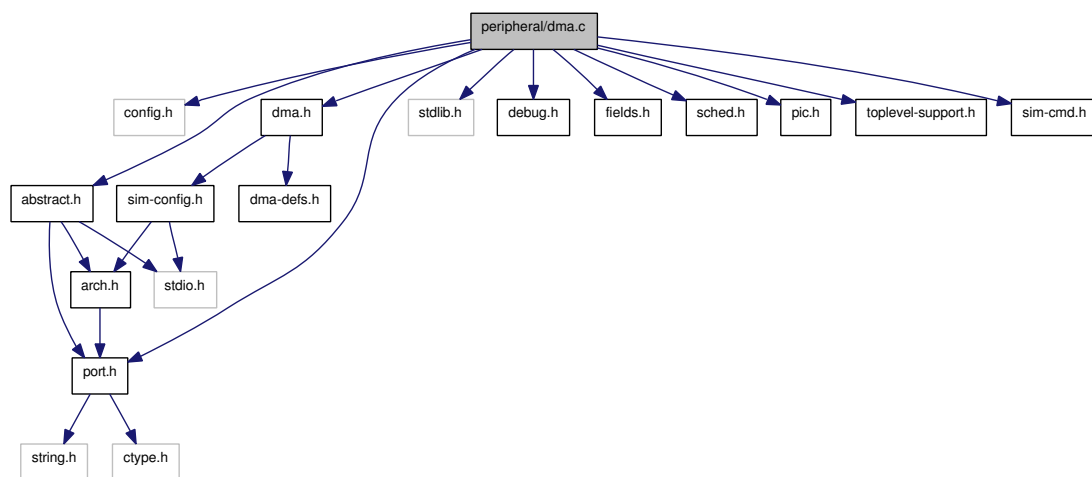
5.116.1.70 **#define DMA_INT_SRC_B 0x10**

5.116.1.71 **#define DMA_NUM_CHANNELS 31**

5.117 peripheral/dma.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "dma.h"
#include "debug.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

- [DEFAULT_DEBUG_CHANNEL](#) (dma)
- 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

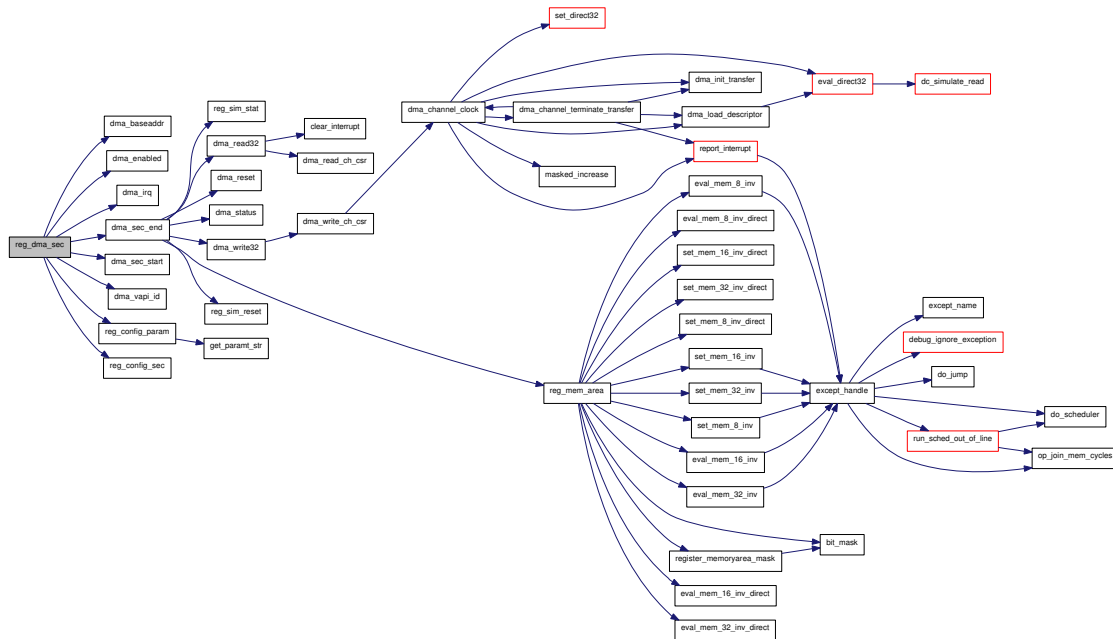
- static struct `dma_controller * dmas = NULL`

5.117.2.21 `struct dma_channel* find_dma_controller_ch (unsigned controller, unsigned channel)`
[read]

5.117.2.22 `static void masked_increase (oraddr_t * value, unsigned long mask)` [static]

5.117.2.23 `void reg_dma_sec (void)`

Here is the call graph for this function:



5.117.2.24 `void set_dma_nd_i (struct dma_channel * channel)`

5.117.2.25 `void set_dma_req_i (struct dma_channel * channel)`

5.117.3 Variable Documentation

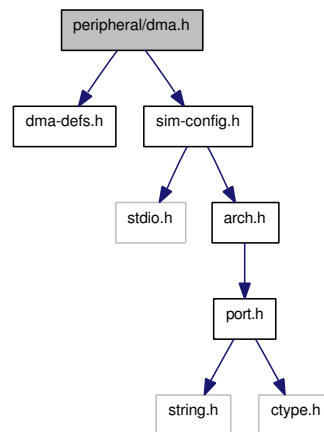
5.117.3.1 `struct dma_controller* dmas = NULL` [static]

5.118 peripheral/dma.h File Reference

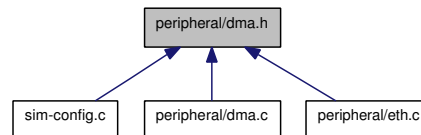
```
#include "dma-defs.h"
```

```
#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](#) ()

5.118.1 Function Documentation

5.118.1.1 unsigned `check_dma_ack_o` (struct `dma_channel` * *channel*)

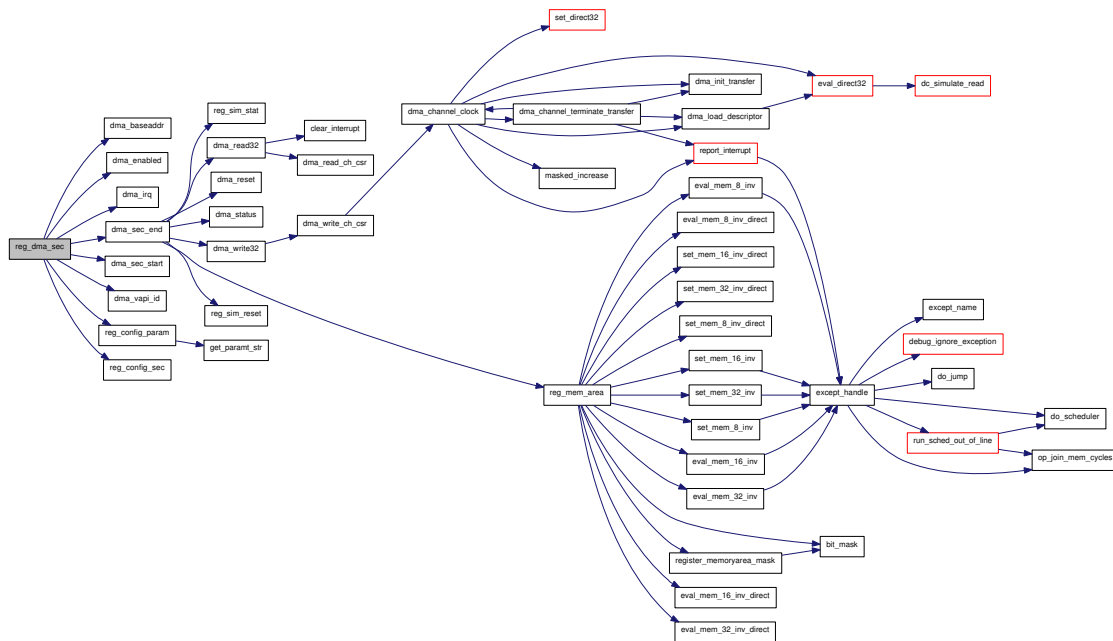
5.118.1.2 void `clear_dma_nd_i` (struct `dma_channel` * *channel*)

5.118.1.3 void `clear_dma_req_i` (struct `dma_channel` * *channel*)

5.118.1.4 struct `dma_channel`* `find_dma_controller_ch` (unsigned *controller*, unsigned *channel*)
[read]

5.118.1.5 void `reg_dma_sec` ()

Here is the call graph for this function:



5.118.1.6 void `set_dma_nd_i` (struct `dma_channel` * *channel*)

5.118.1.7 void `set_dma_req_i` (struct `dma_channel` * *channel*)

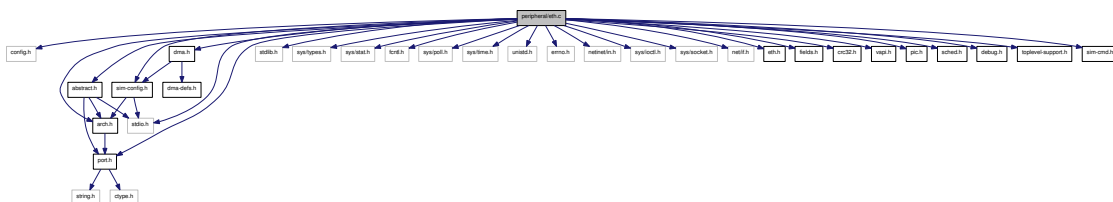
5.119 peripheral/eth.c File Reference

```

#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 "debug.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_RECSMALL_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

- [DEFAULT_DEBUG_CHANNEL](#) (eth)
- 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](#) ()

5.119.1 Define Documentation

5.119.1.1 **#define ETH_ADDR_SPACE 0x1000**

5.119.1.2 **#define ETH_ALEN 6**

5.119.1.3 **#define ETH_BD_BASE 0x400**

5.119.1.4 **#define ETH_BD_COUNT 0x100**

5.119.1.5 **#define ETH_BD_SPACE (4 * ETH_BD_COUNT)**

5.119.1.6 **#define ETH_CMODER_PASSALL_OFFSET 0**

5.119.1.7 **#define ETH_CMODER_RXFLOW_OFFSET 1**

5.119.1.8 **#define ETH_CMODER_TXFLOW_OFFSET 2**

5.119.1.9 **#define ETH_COLLCONF (4 * 0x07)**

5.119.1.10 **#define ETH_COLLCONF_COLLVALID_OFFSET 0**

5.119.1.11 **#define ETH_COLLCONF_COLLVALID_WIDTH 6**

5.119.1.12 **#define ETH_COLLCONF_MAXRET_OFFSET 16**

5.119.1.13 **#define ETH_COLLCONF_MAXRET_WIDTH 4**

5.119.1.14 **#define ETH_CTRLMODER (4 * 0x09)**

5.119.1.15 **#define ETH_DMA_RX_TX 0x800**

5.119.1.16 **#define ETH_HASH0 (4 * 0x12)**

5.119.1.17 **#define ETH_HASH1 (4 * 0x13)**

5.119.1.18 **#define ETH_INT_MASK (4 * 0x02)**

5.119.1.19 **#define ETH_INT_MASK_BUSY_M_OFFSET 4**

5.119.1.20 **#define ETH_INT_MASK_RXB_M_OFFSET 2**

5.119.1.21 **#define ETH_INT_MASK_RXC_M_OFFSET 6**

5.119.1.22 **#define ETH_INT_MASK_RXE_M_OFFSET 3**

5.119.1.23 **#define ETH_INT_MASK_TXB_M_OFFSET 0**

5.119.1.24 **#define ETH_INT_MASK_TXC_M_OFFSET 5**

5.119.1.25 **#define ETH_INT_MASK_TXE_M_OFFSET 1**

5.119.1.26 **#define ETH_INT_SOURCE (4 * 0x01)**

5.119.1.27 **#define ETH_INT_SOURCE_BUSY_OFFSET 4**

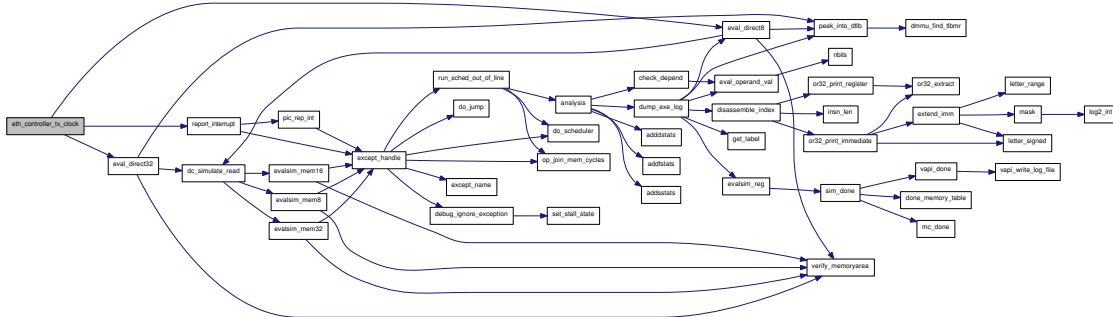
5.119.1.28 **#define ETH_INT_SOURCE_RXB_OFFSET 2**

5.119.1.29 **#define ETH_INT_SOURCE_RXC_OFFSET 6**

5.119.1.30 **#define ETH_INT_SOURCE_RXE_OFFSET 3**

5.119.3.4 static void eth_controller_tx_clock (void * dat) [static]

Here is the call graph for this function:



5.119.3.5 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

5.119.3.6 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

5.119.3.7 static void eth_irq (union param_val val, void * dat) [static]

Set the Ethernet IRQ

Parameters:

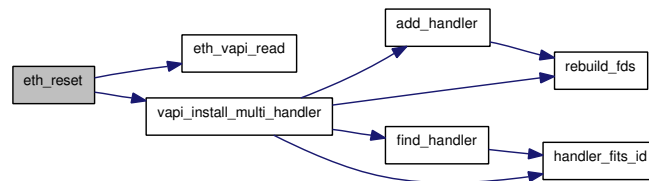
- ← *val* The value to use
- ← *dat* The [config](#) data structure

5.119.3.8 `static uint32_t eth_read32 (oraddr_t addr, void * dat)` [static]

5.119.3.9 `static ssize_t eth_read_rx_file (struct eth_device * eth, void * buf, size_t count)`
[static]

5.119.3.10 `static void eth_reset (void * dat)` [static]

Here is the call graph for this function:



5.119.3.11 `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

5.119.3.12 `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

5.119.3.13 `static void eth_rx_next_packet (struct eth_device * eth)` [static]

Here is the call graph for this function:



5.119.3.14 `static void eth_rxfile (union param_val val, void * dat)` [`static`]

Set the Ethernet DMA Rx file

Free any previously allocated value.

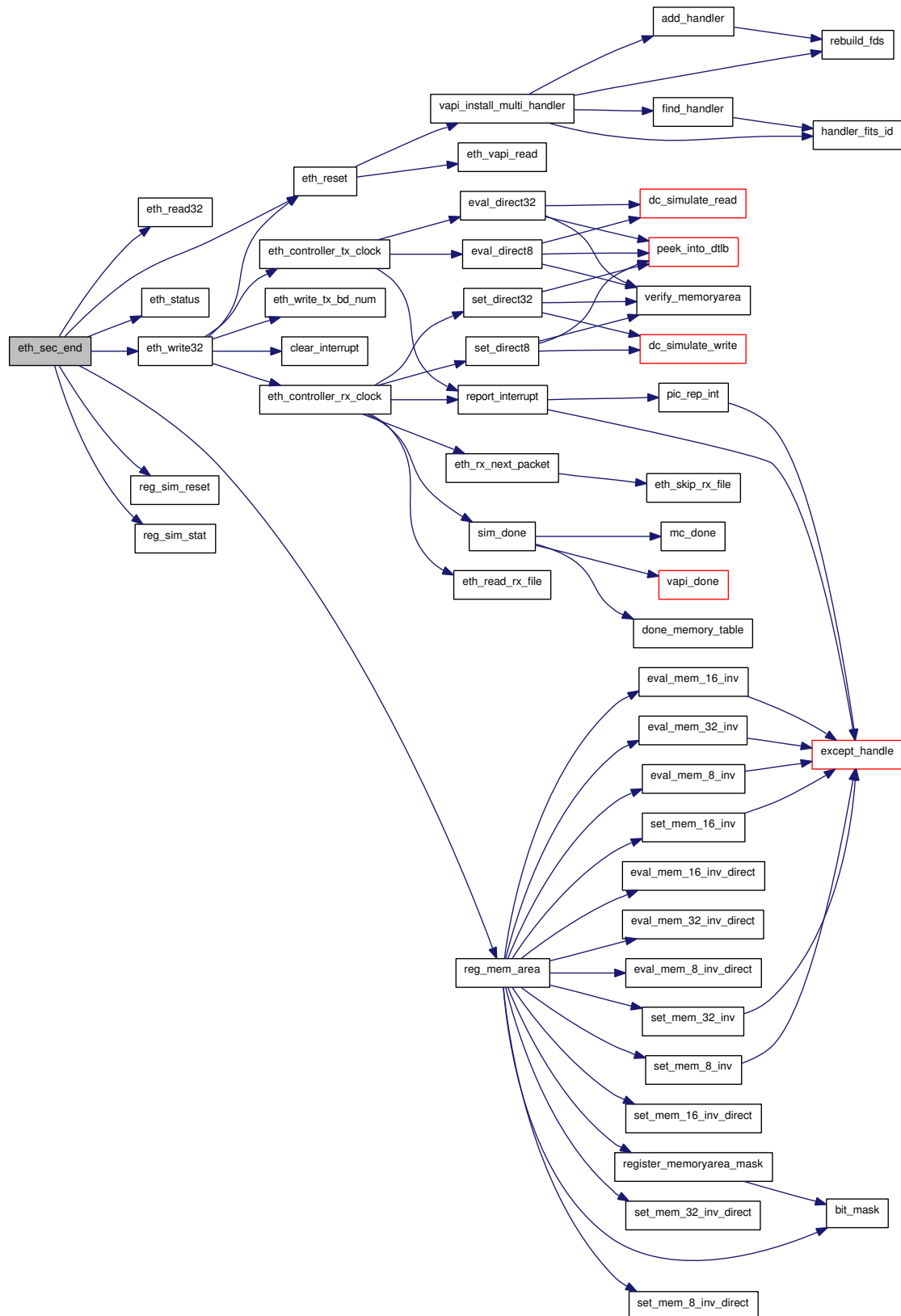
Parameters:

← *val* The value to use

← *dat* The [config](#) data structure

5.119.3.15 static void eth_sec_end (void * dat) [static]

Here is the call graph for this function:



5.119.3.16 static void* eth_sec_start (void) [static]

Initialize a new Ethernet configuration

ALL parameters are set explicitly to default values.

5.119.3.17 static void eth_skip_rx_file (struct eth_device * eth, off_t count) [static]**5.119.3.18 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

5.119.3.19 static void eth_status (void * dat) [static]**5.119.3.20 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

5.119.3.21 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

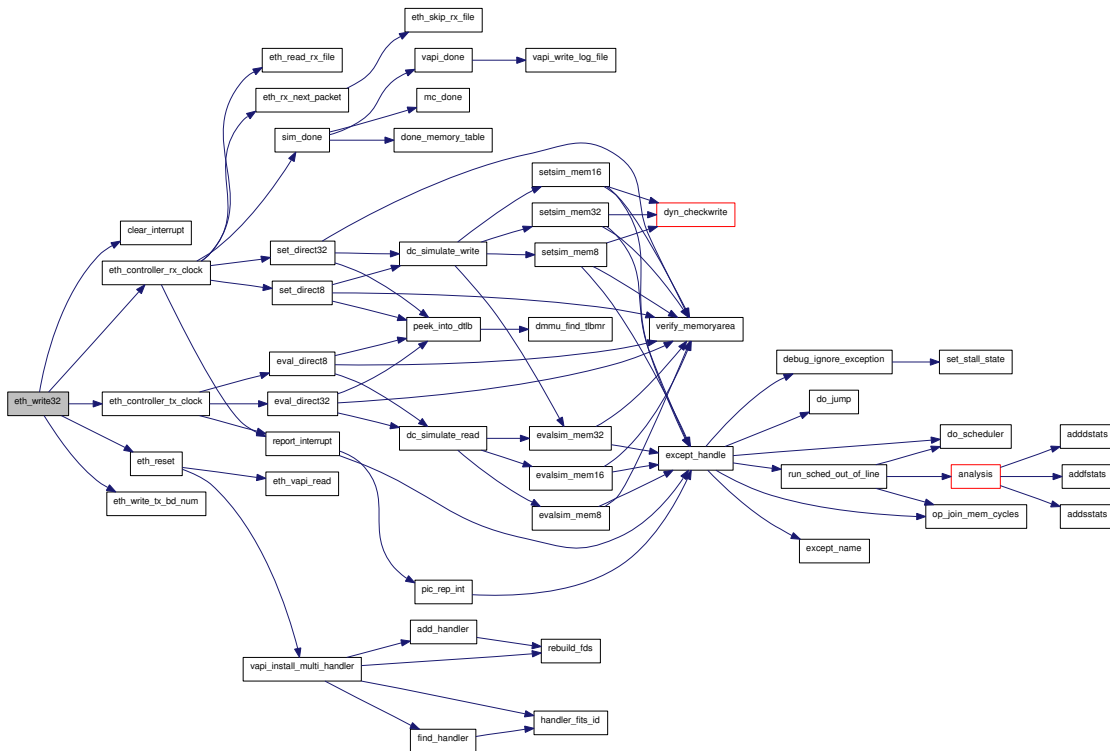
← *dat* The [config](#) data structure

5.119.3.22 `static void eth_vapi_id (union param_val val, void * dat)` [static]

5.119.3.23 `static void eth_vapi_read (unsigned long id, unsigned long data, void * dat)`
[static]

5.119.3.24 `static void eth_write32 (oraddr_t addr, uint32_t value, void * dat)` [static]

Here is the call graph for this function:

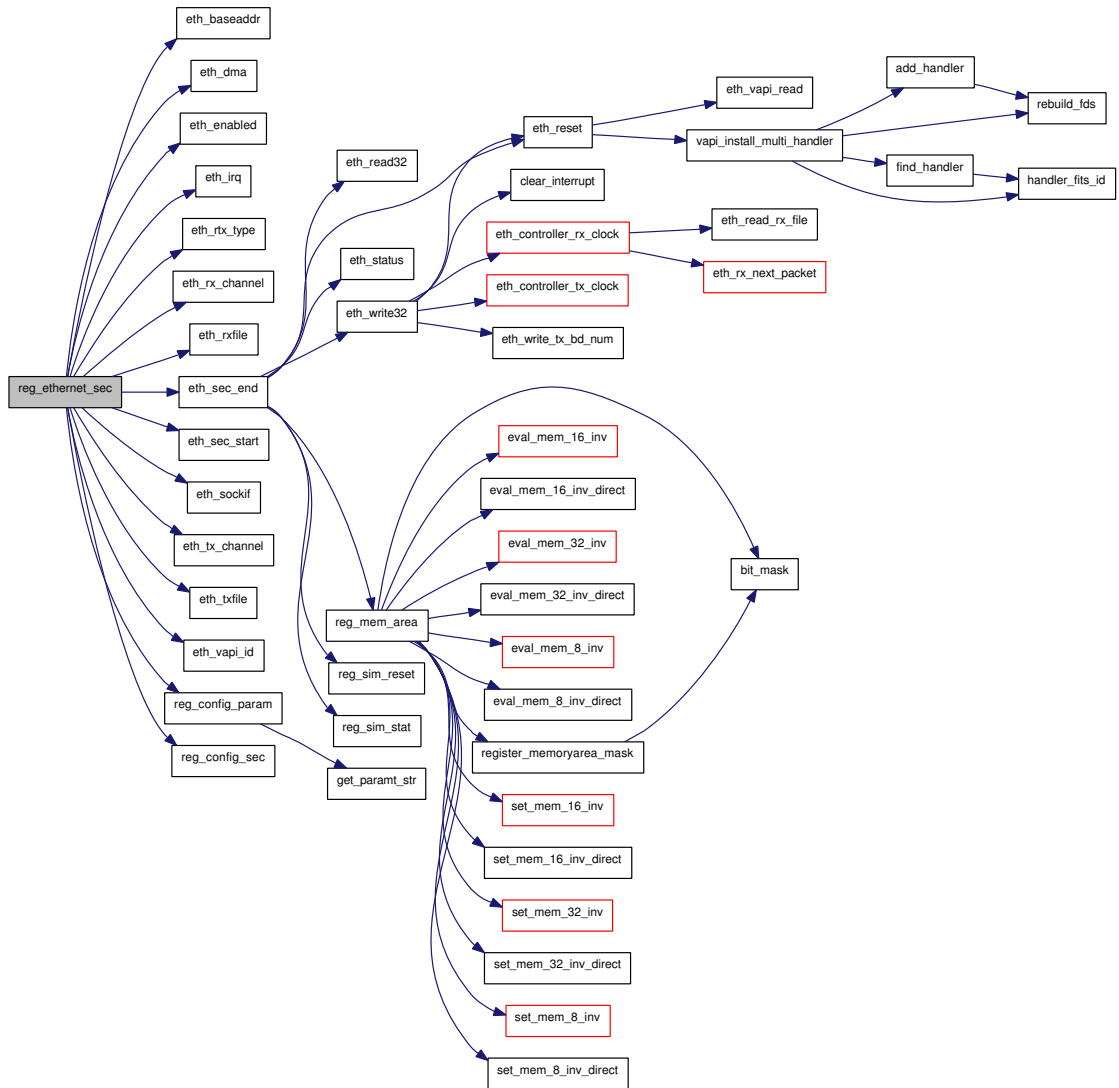


5.119.3.25 `static void eth_write_tx_bd_num (struct eth_device * eth, unsigned long value)`
[static]

5.119.3.26 `void reg_ethernet_sec ()`

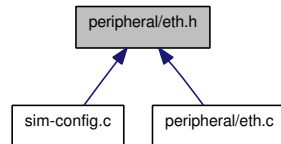
Register a new Ethernet configuration

Here is the call graph for this function:



5.120 peripheral/eth.h File Reference

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



Functions

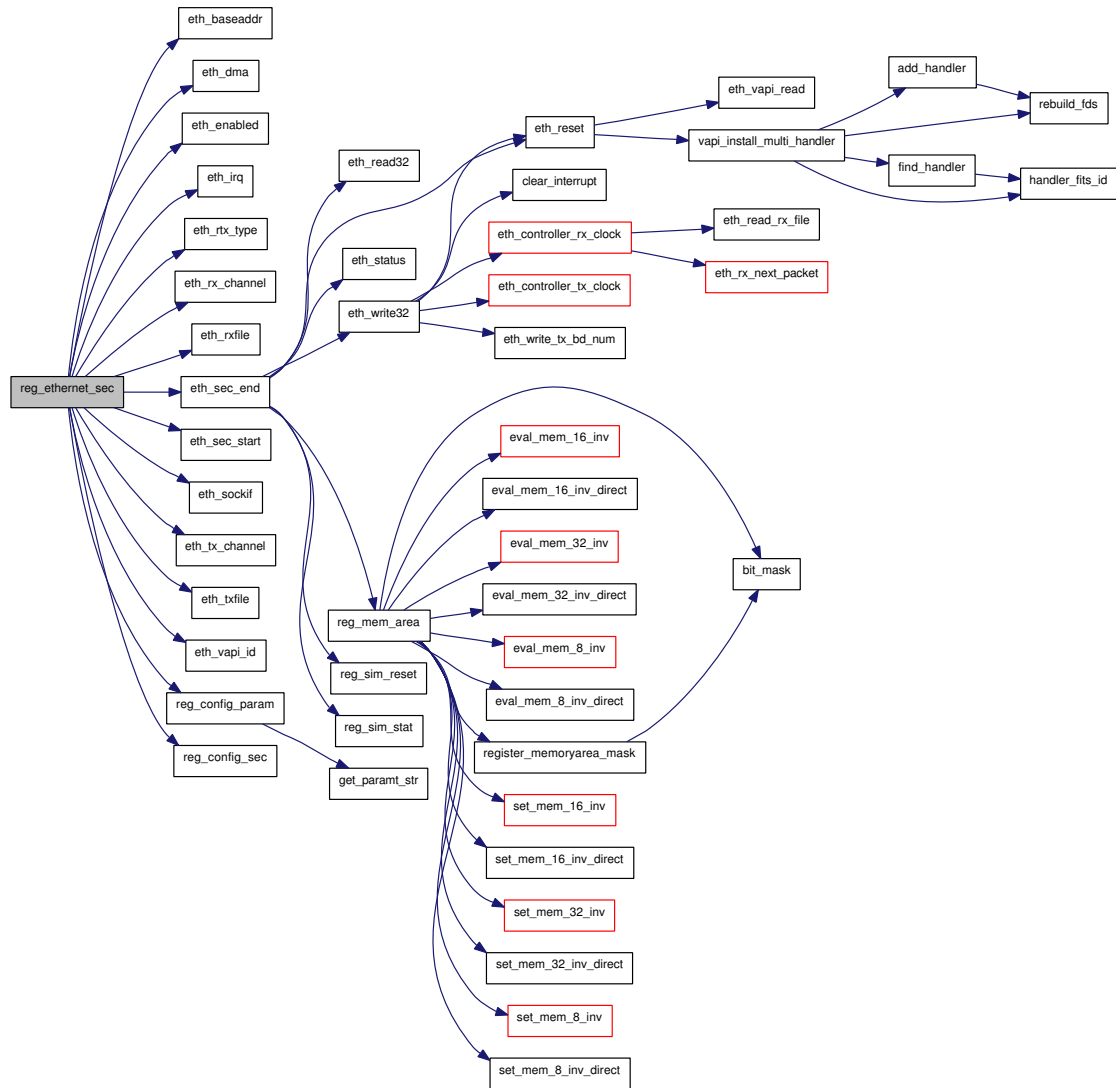
- void [reg_ethernet_sec \(\)](#)

5.120.1 Function Documentation

5.120.1.1 void [reg_ethernet_sec \(\)](#)

Register a new Ethernet configuration

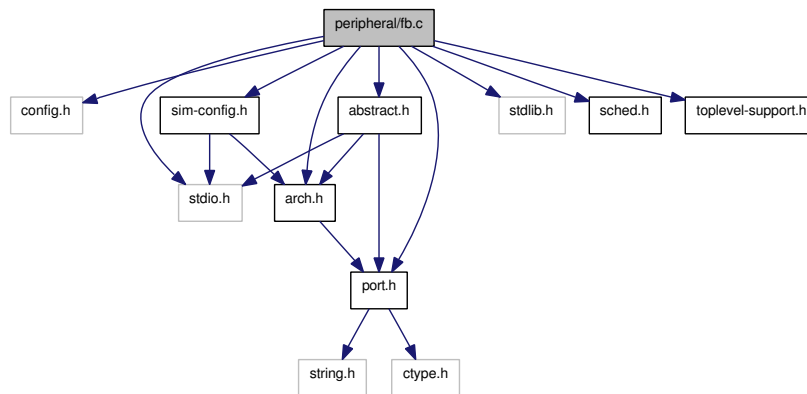
Here is the call graph for this function:



5.121 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` ()

5.121.1 Define Documentation

5.121.1.1 `#define CAM_SIZEX 352`

5.121.1.2 `#define CAM_SIZEY 288`

5.121.1.3 `#define CNV16(x) (x)`

5.121.1.4 `#define CNV32(x) (x)`

5.121.1.5 `#define FB_BUFADDR 0x0004`

5.121.1.6 `#define FB_CAMBUFADDR 0x0008`

5.121.1.7 `#define FB_CAMPOSADDR 0x000c`

5.121.1.8 `#define FB_CTRL 0x0000`

5.121.1.9 `#define FB_PAL 0x0400`

5.121.1.10 `#define FB_SIZEX 640`

5.121.1.11 `#define FB_SIZEY 480`

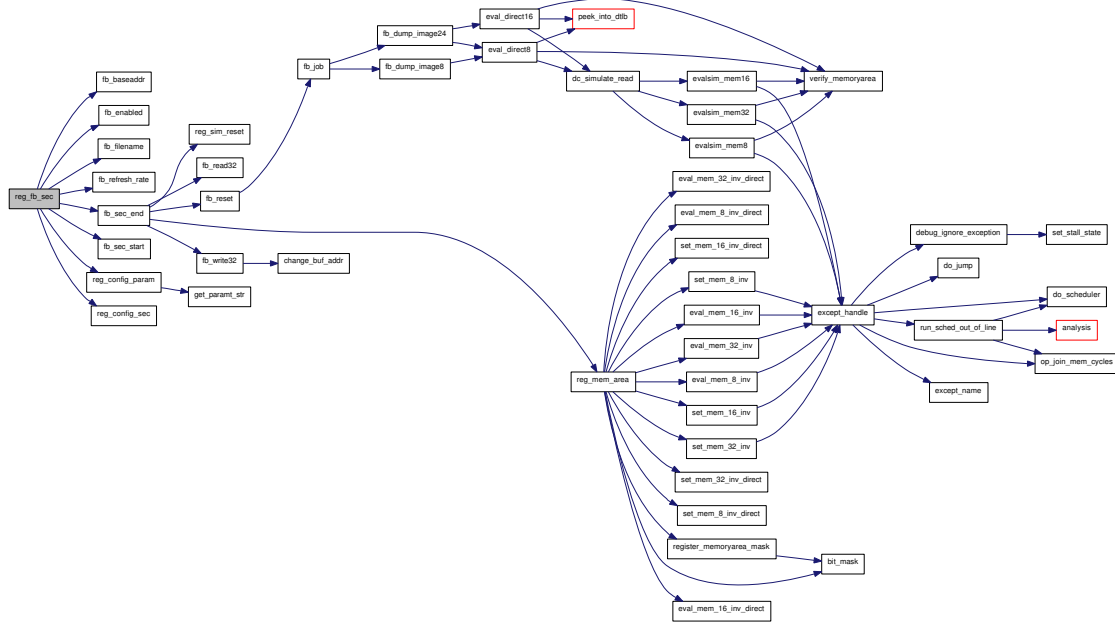
5.121.1.12 `#define FB_WRAP (512*1024)`

5.121.1.13 `#define REFRESH_DIVIDER 20`

Relative amount of time spent in refresh

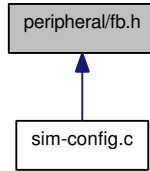
5.121.2.14 void reg_fb_sec ()

Here is the call graph for this function:



5.122 peripheral/fb.h File Reference

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



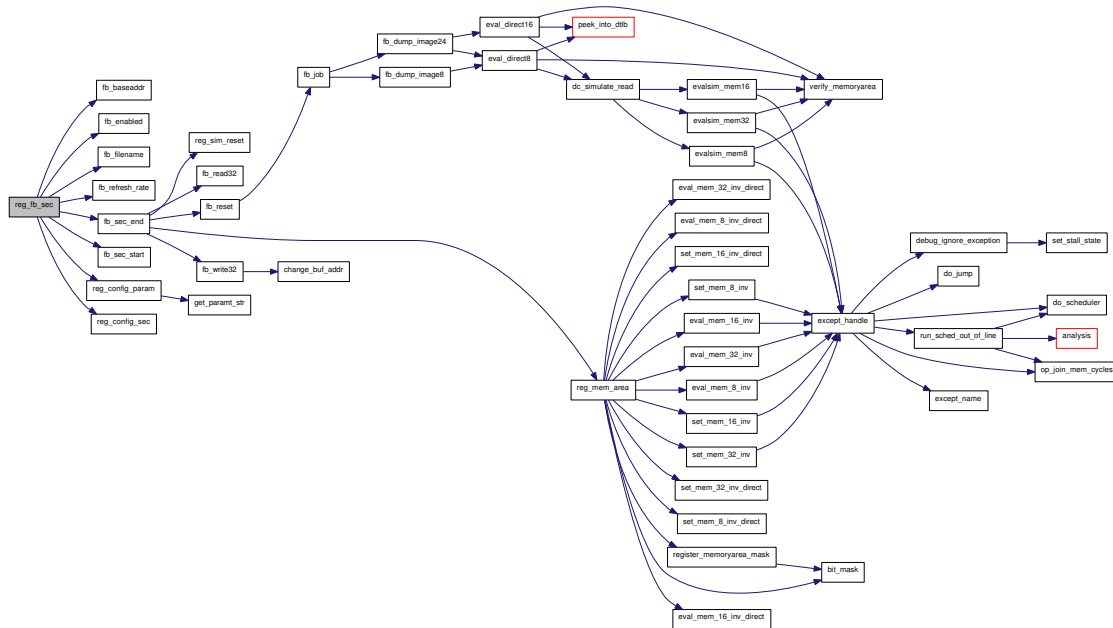
Functions

- void [reg_fb_sec \(\)](#)

5.122.1 Function Documentation

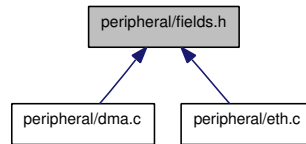
5.122.1.1 void reg_fb_sec ()

Here is the call graph for this function:



5.123 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)

5.123.1 Define Documentation

5.123.1.1 #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) & ~
  
```

5.123.1.2 **#define CLEAR_FLAG(reg_value, reg_name, flag_name) { (reg_value) &= ~(1LU << reg_name##_##flag_name##_OFFSET); }**

5.123.1.3 **#define FIELD_MASK(reg_name, field_name) ((~(~0LU << reg_name##_##field_name##_WIDTH)) << reg_name##_##field_name##_OFFSET)**

5.123.1.4 **#define FIELD_SHIFT(reg_name, field_name) (reg_name##_##field_name##_OFFSET)**

5.123.1.5 **#define FLAG_MASK(reg_name, flag_name) (1LU << reg_name##_##flag_name##_OFFSET)**

5.123.1.6 **#define FLAG_SHIFT(reg_name, flag_name) (reg_name##_##flag_name##_OFFSET)**

5.123.1.7 **#define GET_FIELD(reg_value, reg_name, field_name) (((reg_value) >> reg_name##_##field_name##_OFFSET) & (~(~0LU << reg_name##_##field_name##_WIDTH)))**

5.123.1.8 **#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); }
```

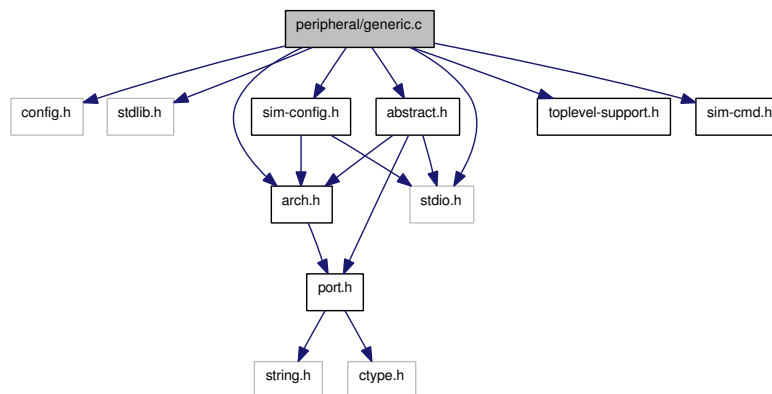
5.123.1.9 **#define SET_FLAG(reg_value, reg_name, flag_name) { (reg_value) |= 1LU << reg_name##_##flag_name##_OFFSET; }**

5.123.1.10 **#define TEST_FLAG(reg_value, reg_name, flag_name) (((reg_value) >> reg_name##_##flag_name##_OFFSET) & 1LU)**

5.124 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)

5.124.1 Function Documentation

5.124.1.1 static unsigned long int [ext_read](#) (unsigned long int *addr*, unsigned long int *mask*)
[static]

5.124.1.2 static void [ext_write](#) (unsigned long int *addr*, unsigned long int *mask*, unsigned long int *value*) [static]

5.124.1.3 static void [generic_baseaddr](#) (union [param_val](#) *val*, void * *dat*) [static]

5.124.1.4 static void [generic_byte_enabled](#) (union [param_val](#) *val*, void * *dat*) [static]

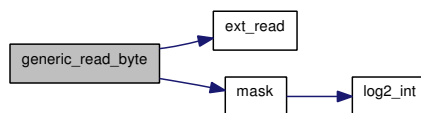
5.124.1.5 static void [generic_enabled](#) (union [param_val](#) *val*, void * *dat*) [static]

5.124.1.6 static void [generic_hw_enabled](#) (union [param_val](#) *val*, void * *dat*) [static]

5.124.1.7 static void [generic_name](#) (union [param_val](#) *val*, void * *dat*) [static]

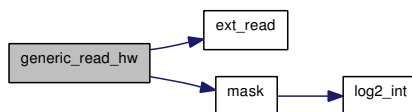
5.124.1.8 static uint8_t [generic_read_byte](#) (oraddr_t *addr*, void * *dat*) [static]

Here is the call graph for this function:



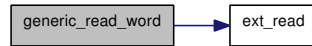
5.124.1.9 static uint16_t [generic_read_hw](#) (oraddr_t *addr*, void * *dat*) [static]

Here is the call graph for this function:



5.124.1.10 static uint32_t generic_read_word (oraddr_t addr, void * dat) [static]

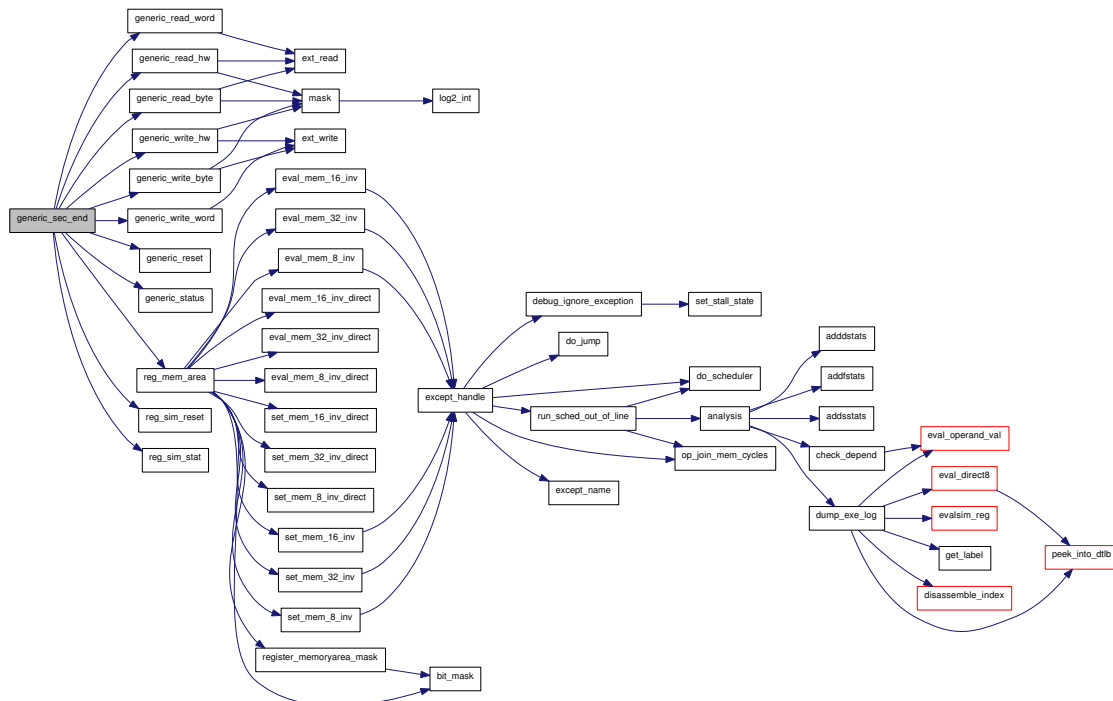
Here is the call graph for this function:



5.124.1.11 static void generic_reset (void * dat) [static]

5.124.1.12 static void generic_sec_end (void * dat) [static]

Here is the call graph for this function:



5.124.1.13 `static void* generic_sec_start ()` [static]

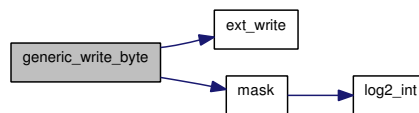
5.124.1.14 `static void generic_size (union param_val val, void * dat)` [static]

5.124.1.15 `static void generic_status (void * dat)` [static]

5.124.1.16 `static void generic_word_enabled (union param_val val, void * dat)` [static]

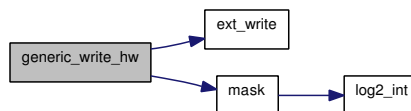
5.124.1.17 `static void generic_write_byte (oraddr_t addr, uint8_t value, void * dat)` [static]

Here is the call graph for this function:



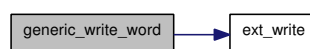
5.124.1.18 `static void generic_write_hw (oraddr_t addr, uint16_t value, void * dat)` [static]

Here is the call graph for this function:



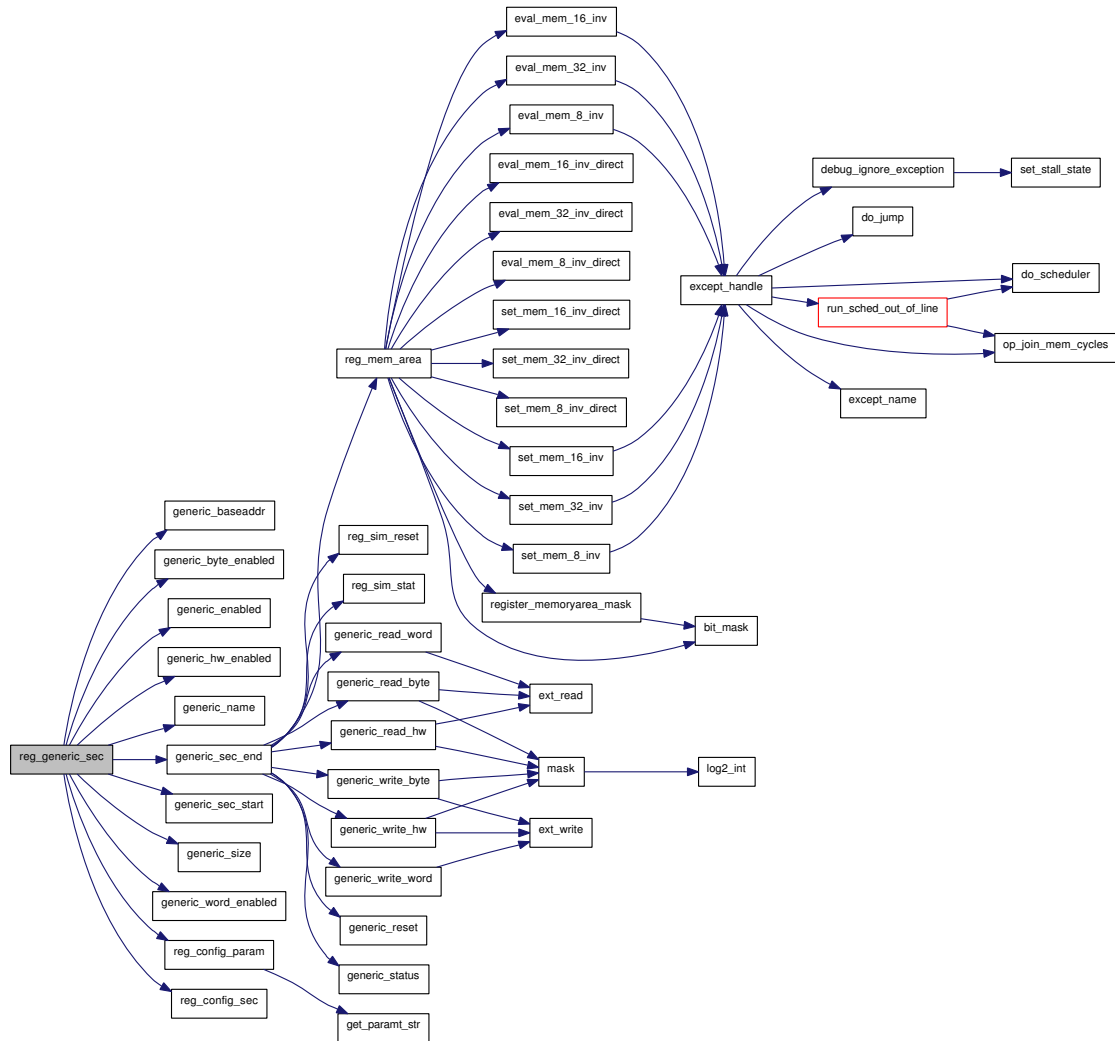
5.124.1.19 `static void generic_write_word (oraddr_t addr, uint32_t value, void * dat)` [static]

Here is the call graph for this function:



5.124.1.20 void reg_generic_sec (void)

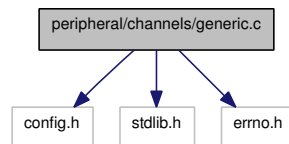
Here is the call graph for this function:



5.125 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)

5.125.1 Function Documentation

5.125.1.1 void [generic_close](#) (void * *data*)

5.125.1.2 void [generic_free](#) (void * *data*)

5.125.1.3 int [generic_open](#) (void * *data*)

5.126 peripheral/generic.h File Reference

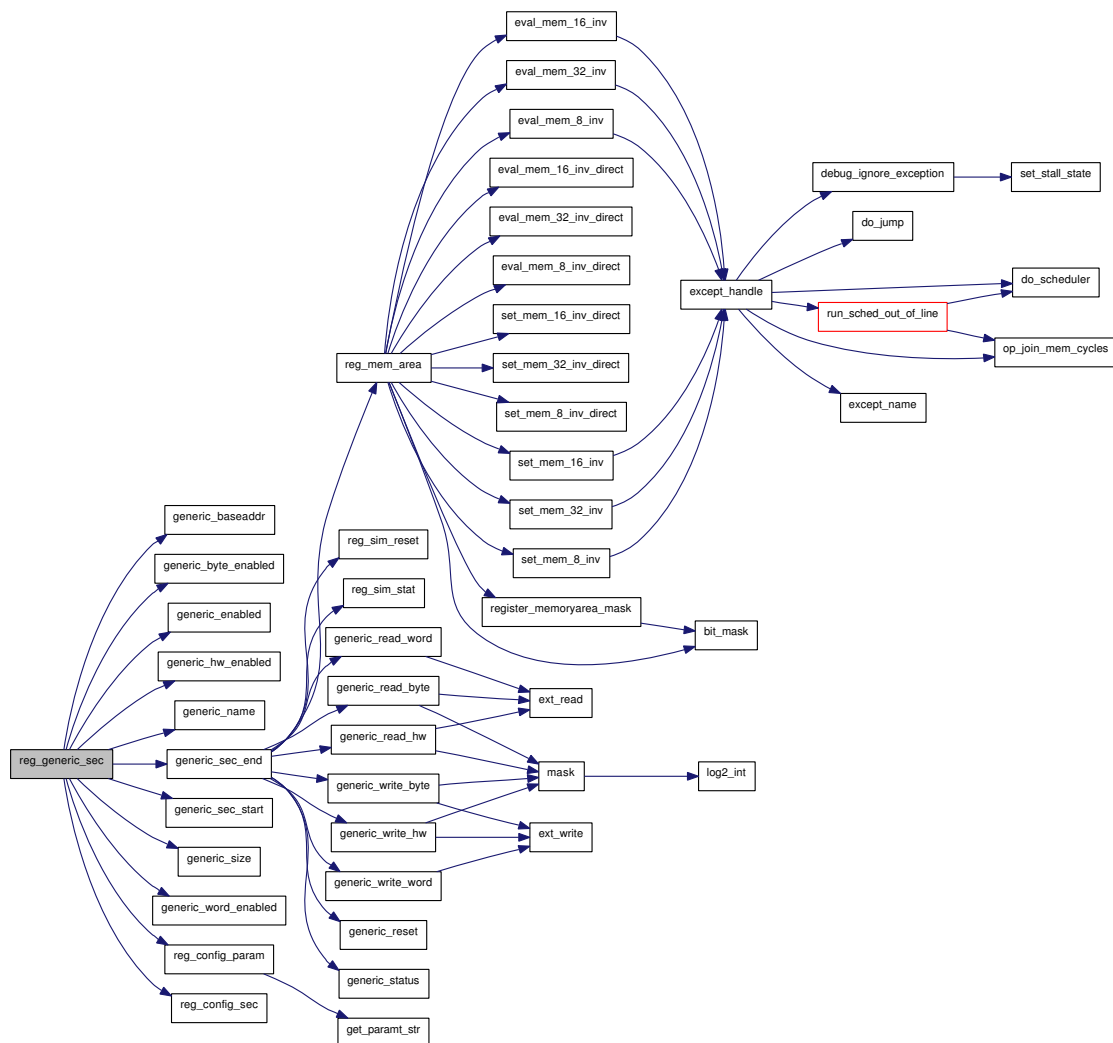
Functions

- void [reg_generic_sec](#) ()

5.126.1 Function Documentation

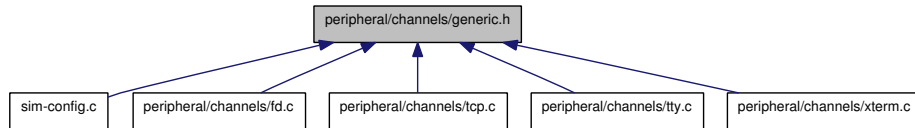
5.126.1.1 void [reg_generic_sec](#) ()

Here is the call graph for this function:



5.127 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)

5.127.1 Function Documentation

5.127.1.1 void [generic_close](#) (void * *data*)

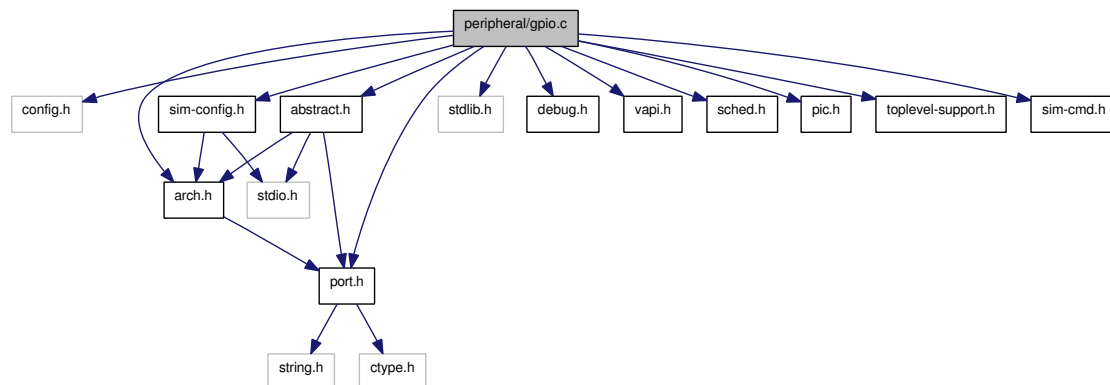
5.127.1.2 void [generic_free](#) (void * *data*)

5.127.1.3 int [generic_open](#) (void * *data*)

5.128 peripheral/gpio.c File Reference

```
#include "config.h"
#include "port.h"
#include <stdlib.h>
#include "sim-config.h"
#include "arch.h"
#include "debug.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

- [DEFAULT_DEBUG_CHANNEL](#) (gpio)
- 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)

5.128.1 Define Documentation

- 5.128.1.1 `#define GPIO_ADDR_SPACE 0x20`
- 5.128.1.2 `#define RGPIO_AUX 0x14`
- 5.128.1.3 `#define RGPIO_CTRL 0x18`
- 5.128.1.4 `#define RGPIO_CTRL_ECLK 0x00000001`
- 5.128.1.5 `#define RGPIO_CTRL_INTE 0x00000004`
- 5.128.1.6 `#define RGPIO_CTRL_INTS 0x00000008`
- 5.128.1.7 `#define RGPIO_CTRL_NEC 0x00000002`
- 5.128.1.8 `#define RGPIO_IN 0x00`
- 5.128.1.9 `#define RGPIO_INTE 0x0C`
- 5.128.1.10 `#define RGPIO_INTS 0x1C`
- 5.128.1.11 `#define RGPIO_OE 0x08`
- 5.128.1.12 `#define RGPIO_OUT 0x04`
- 5.128.1.13 `#define RGPIO_PTRIG 0x10`

5.128.2 Enumeration Type Documentation

5.128.2.1 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

5.128.3 Function Documentation

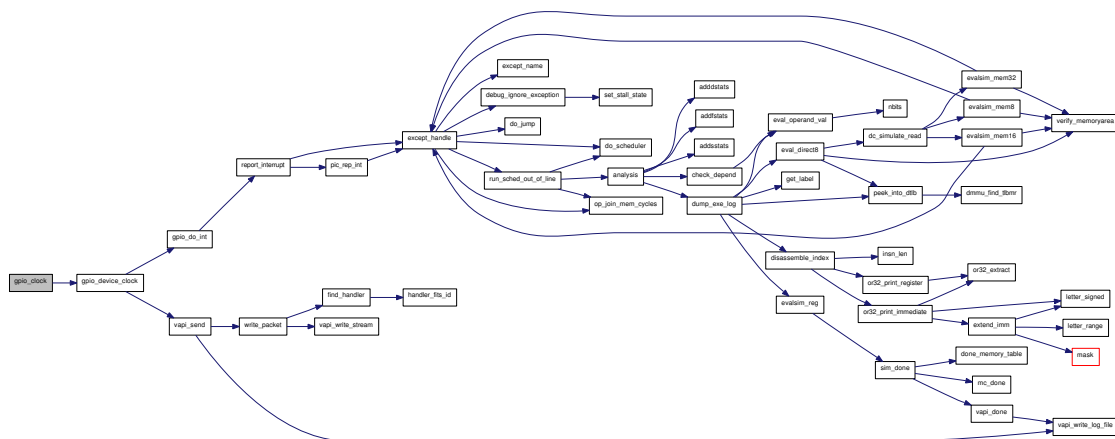
5.128.3.1 DEFAULT_DEBUG_CHANNEL (gpio)

5.128.3.2 static void gpio_base_vapi_id (union param_val val, void * dat) [static]

5.128.3.3 static void gpio_baseaddr (union param_val val, void * dat) [static]

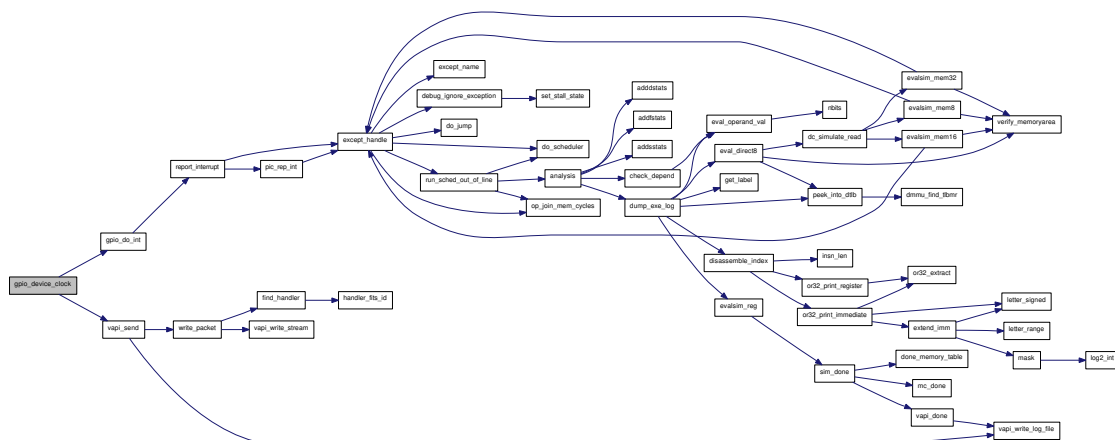
5.128.3.4 static void gpio_clock (void * dat) [static]

Here is the call graph for this function:



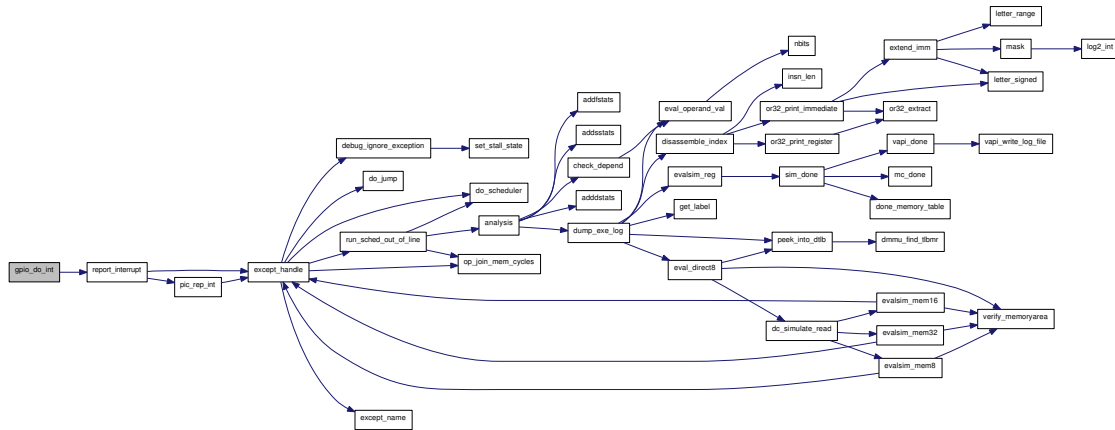
5.128.3.5 static void gpio_device_clock (struct gpio_device * gpio) [static]

Here is the call graph for this function:



5.128.3.6 static void gpio_do_int (void * dat) [static]

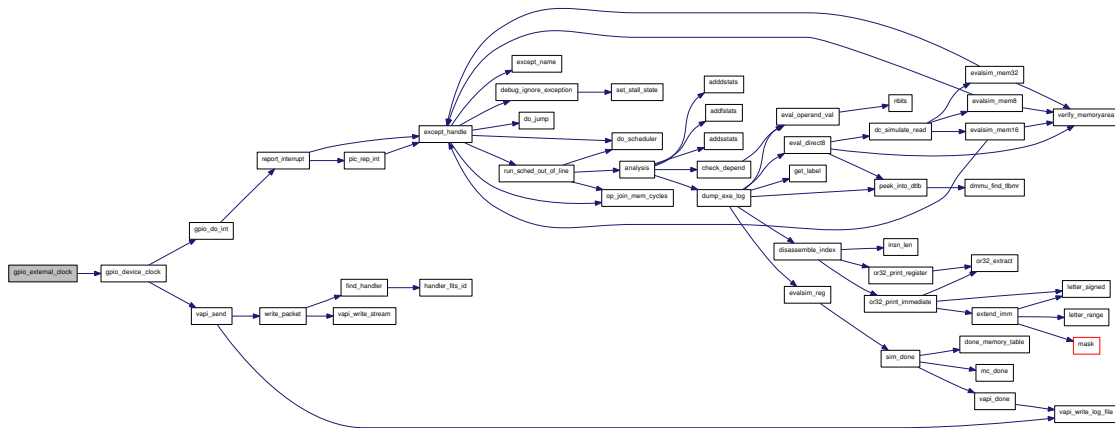
Here is the call graph for this function:



5.128.3.7 static void gpio_enabled (union param_val val, void * dat) [static]

5.128.3.8 static void gpio_external_clock (unsigned long value, struct gpio_device * gpio) [static]

Here is the call graph for this function:

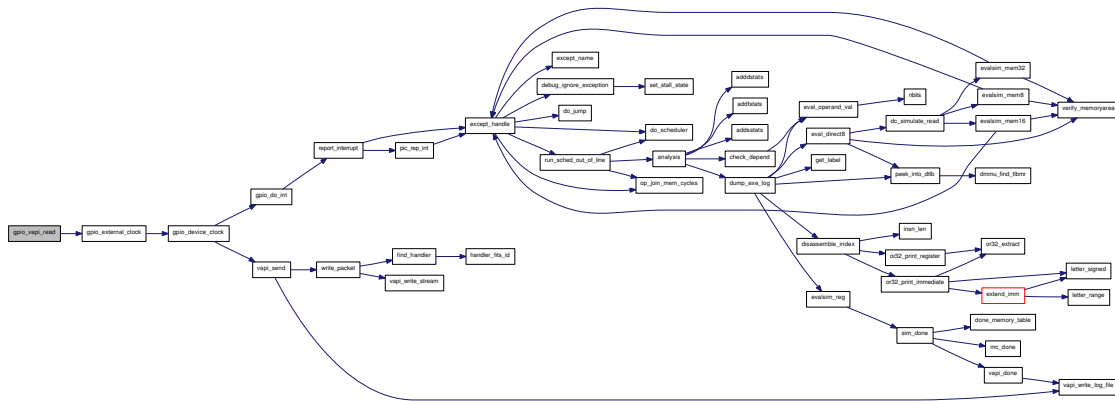


5.128.3.13 `static void* gpio_sec_start (void)` [static]

5.128.3.14 `static void gpio_status (void *dat)` [static]

5.128.3.15 `static void gpio_vapi_read (unsigned long id, unsigned long data, void * dat)`
[static]

Here is the call graph for this function:



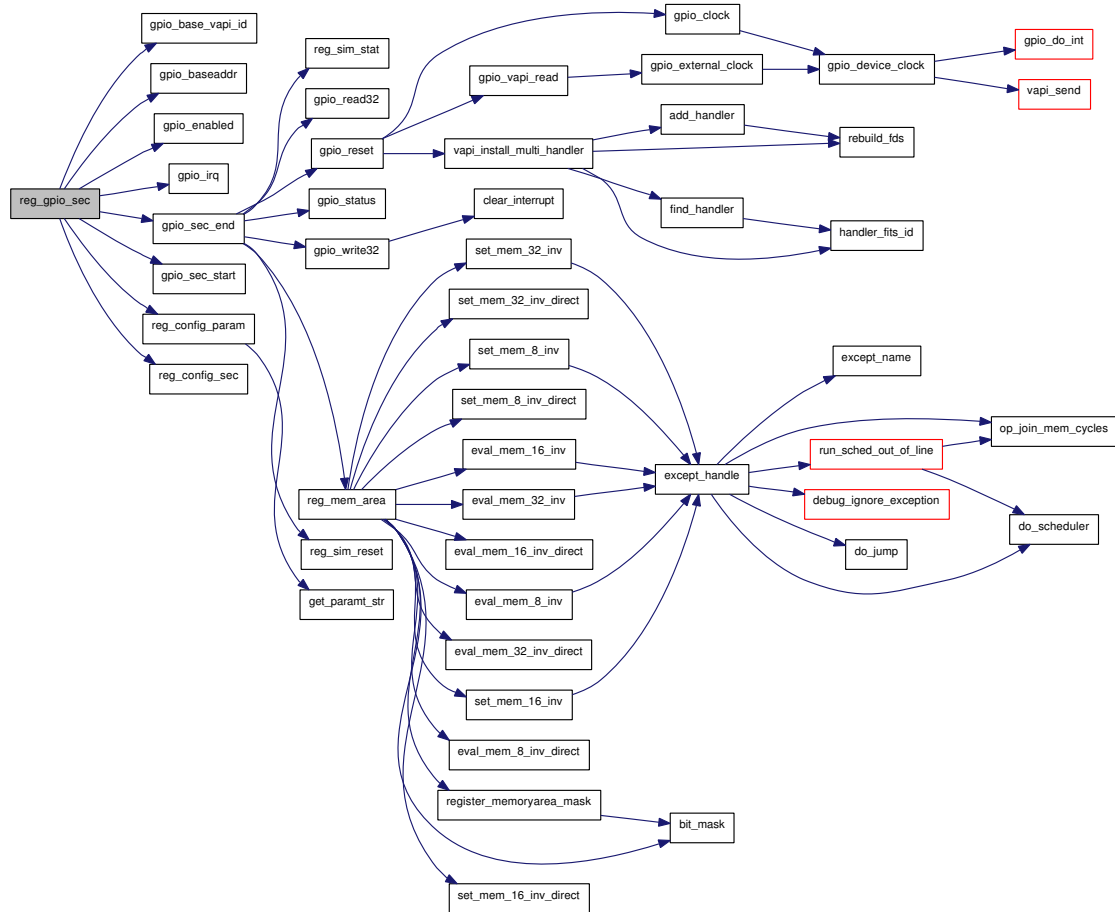
5.128.3.16 `static void gpio_write32 (oraddr_t addr, uint32_t value, void * dat)` [static]

Here is the call graph for this function:



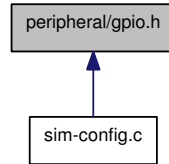
5.128.3.17 void reg_gpio_sec (void)

Here is the call graph for this function:



5.129 peripheral/gpio.h File Reference

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



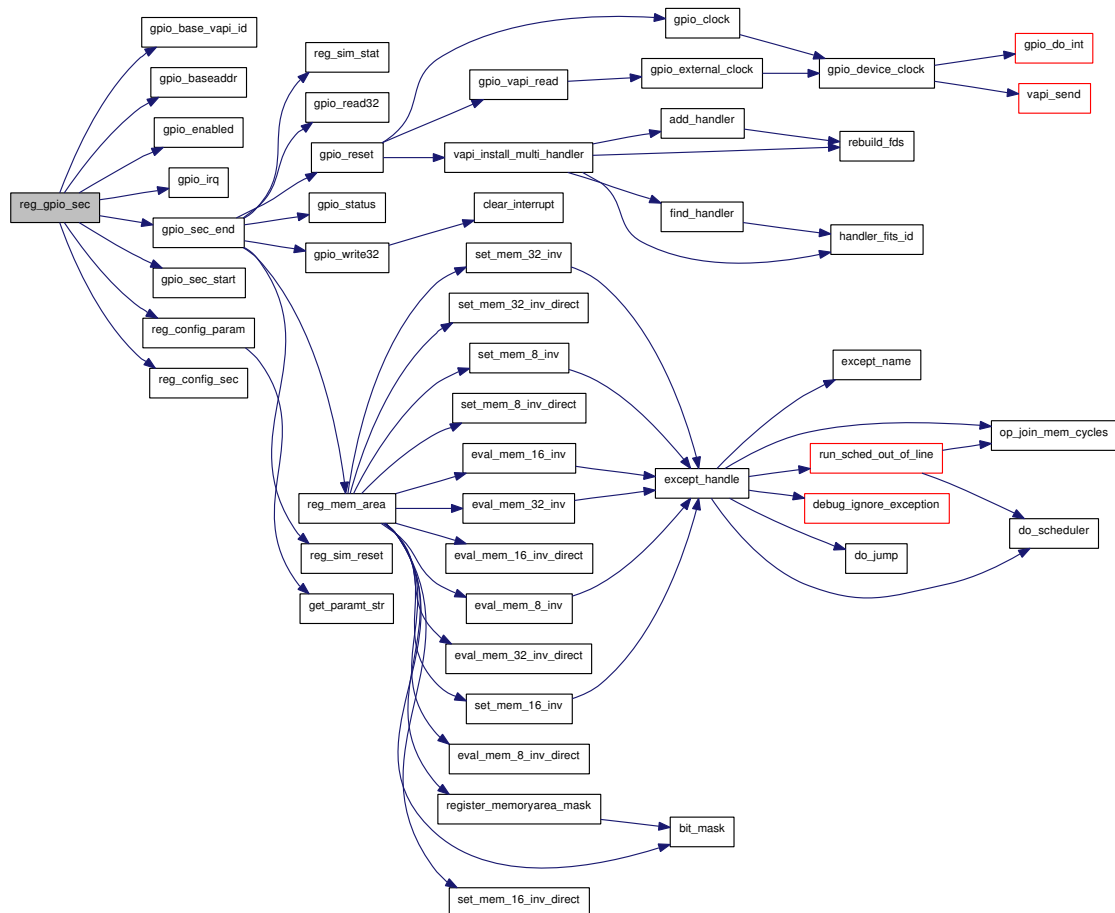
Functions

- void [reg_gpio_sec](#) ()

5.129.1 Function Documentation

5.129.1.1 void [reg_gpio_sec](#) ()

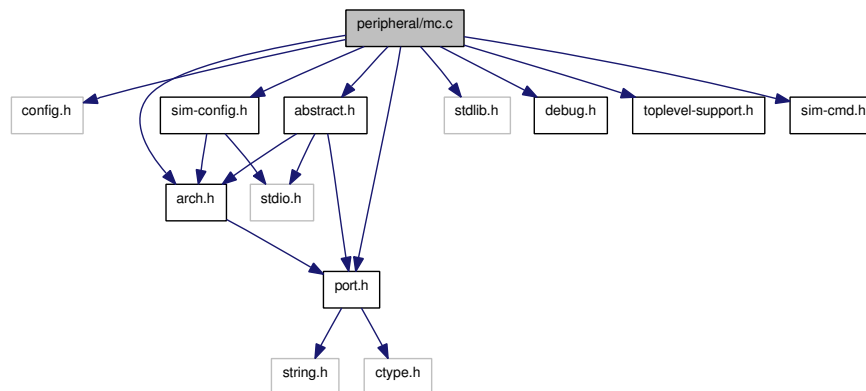
Here is the call graph for this function:



5.130 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 "debug.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 0x0FFF83FFLU
- #define MC_TMS_SSRAM_VALID 0x00000000LU
- #define MC_TMS_ASYNC_VALID 0x03FFFFFFLU
- #define MC_TMS_SYNC_VALID 0x01FFFFFFLU
- #define MC_TMS_VALID 0xFFFFFFFFLU
- #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

- `DEFAULT_DEBUG_CHANNEL` (`mc`)
- 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

5.130.1 Define Documentation

5.130.1.1 **#define MC_ADDR_SPACE (MC_CSC(N_CE))**

5.130.1.2 **#define MC_BA_MASK 0x08**

5.130.1.3 **#define MC_BA_MASK_VALID 0x000003FFLU**

5.130.1.4 **#define MC_CSC(i) (0x10 + (i) * 8)**

5.130.1.5 **#define MC_CSC_BAS_OFFSET 9**

5.130.1.6 **#define MC_CSC_BW_OFFSET 4**

5.130.1.7 **#define MC_CSC_BW_WIDTH 2**

5.130.1.8 **#define MC_CSC_EN_OFFSET 0**

5.130.1.9 **#define MC_CSC_KRO_OFFSET 10**

5.130.1.10 **#define MC_CSC_MEMTYPE_ASYNC 2**

5.130.1.11 **#define MC_CSC_MEMTYPE_OFFSET 1**

5.130.1.12 **#define MC_CSC_MEMTYPE_SDRAM 0**

5.130.1.13 **#define MC_CSC_MEMTYPE_SSRAM 1**

5.130.1.14 **#define MC_CSC_MEMTYPE_SYNC 3**

5.130.1.15 **#define MC_CSC_MEMTYPE_WIDTH 2**

5.130.1.16 **#define MC_CSC_MS_OFFSET 6**

5.130.1.17 **#define MC_CSC_MS_WIDTH 2**

5.130.1.18 **#define MC_CSC_PEN_OFFSET 11**

5.130.1.19 **#define MC_CSC_SEL_OFFSET 16**

5.130.1.20 **#define MC_CSC_SEL_WIDTH 8**

5.130.1.21 **#define MC_CSC_VALID 0x00FF0FFFLLU**

5.130.1.22 **#define MC_CSC_WP_OFFSET 8**

5.130.1.23 **#define MC_CSR 0x00**

5.130.1.24 **#define MC_CSR_VALID 0xFF000703LU**

5.130.1.25 **#define MC_POC 0x04**

5.130.1.26 **#define MC_POC_EN_BW_OFFSET 0**

5.130.1.27 **#define MC_POC_EN_BW_WIDTH 2**

5.130.1.28 **#define MC_POC_EN_MEMTYPE_OFFSET 2**

5.130.1.29 **#define MC_POC_EN_MEMTYPE_WIDTH 2**

5.130.1.30 **#define MC_POC_VALID 0x0000000FLU**

5.130.2.4 `static void mc_enabled (union param_val val, void * dat) [static]`

5.130.2.5 `static void mc_index (union param_val val, void * dat) [static]`

5.130.2.6 `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

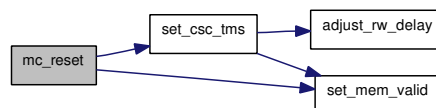
← *dat* The [config](#) data structure

5.130.2.7 `uint32_t mc_read_word (oraddr_t addr, void * dat)`

5.130.2.8 `void mc_reg_mem_area (struct dev_memarea * mem, unsigned int cs, int mc)`

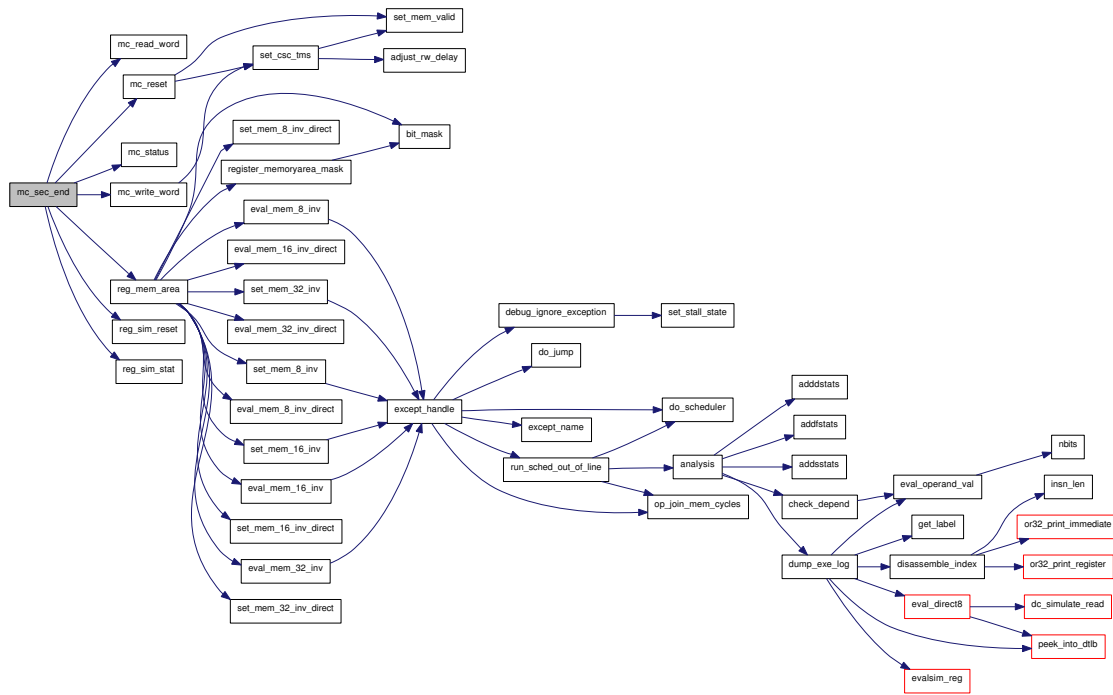
5.130.2.9 `void mc_reset (void * dat)`

Here is the call graph for this function:



5.130.2.10 static void mc_sec_end (void * dat) [static]

Here is the call graph for this function:



5.130.2.11 static void* mc_sec_start () [static]

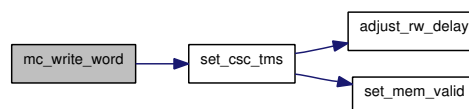
Initialize a new memory controller configuration

ALL parameters are set explicitly to default values.

5.130.2.12 void mc_status (void * dat)

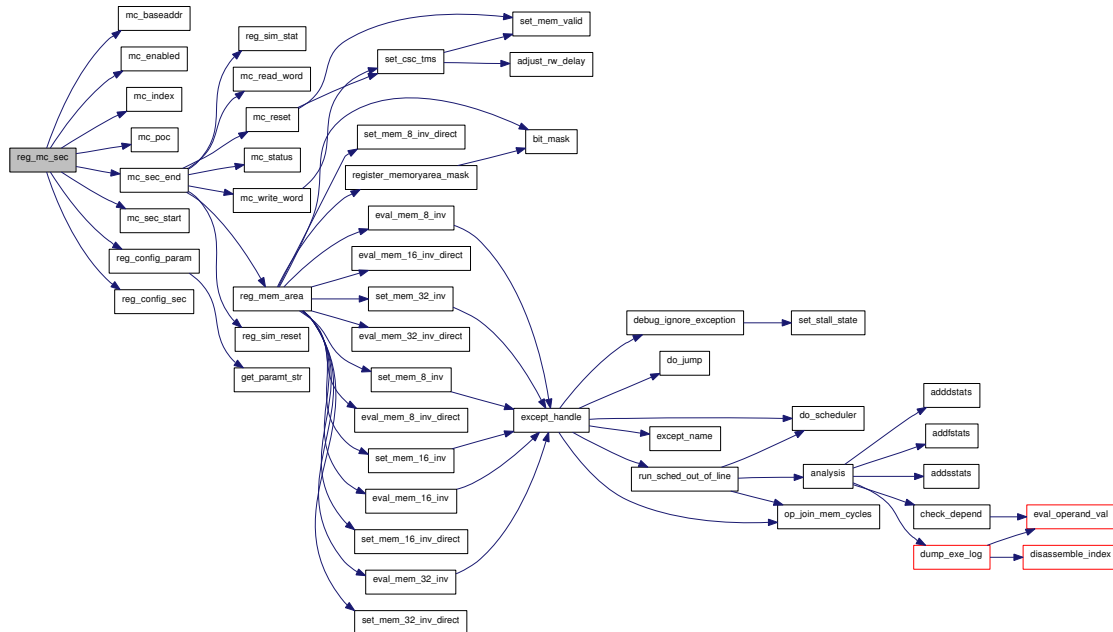
5.130.2.13 void mc_write_word (oraddr_t addr, uint32_t value, void * dat)

Here is the call graph for this function:



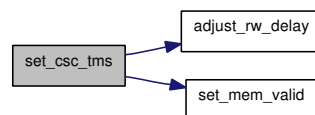
5.130.2.14 void reg_mc_sec (void)

Here is the call graph for this function:



5.130.2.15 void set_csc_tms (int cs, uint32_t csc, uint32_t tms, struct mc * mc)

Here is the call graph for this function:



5.130.3 Variable Documentation

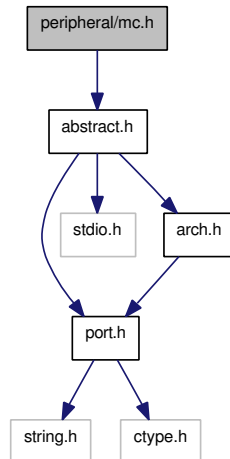
5.130.3.1 struct mc_area* mc_areas = NULL [static]

5.130.3.2 struct mc* mcs = NULL [static]

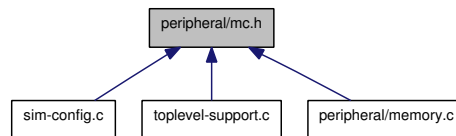
5.131 peripheral/mc.h File Reference

```
#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)

5.131.1 Function Documentation

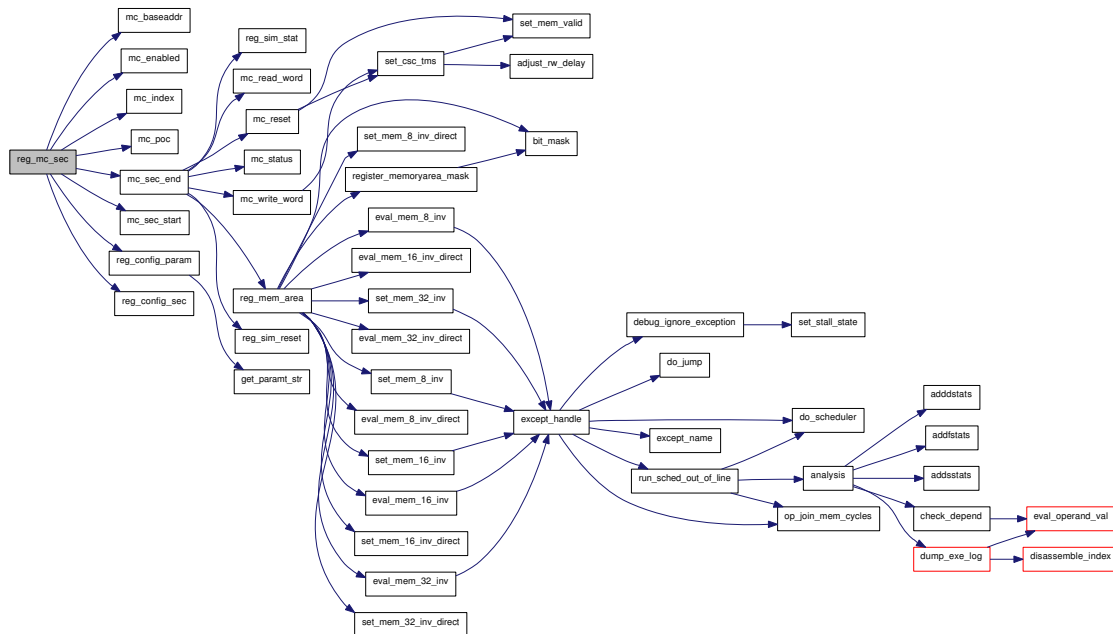
5.131.1.1 void mc_done ()

Free all allocated memory

5.131.1.2 void mc_reg_mem_area (struct dev_memarea * mem, unsigned int cs, int mc)

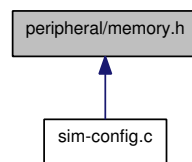
5.131.1.3 void reg_mc_sec ()

Here is the call graph for this function:



5.132 peripheral/memory.h File Reference

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



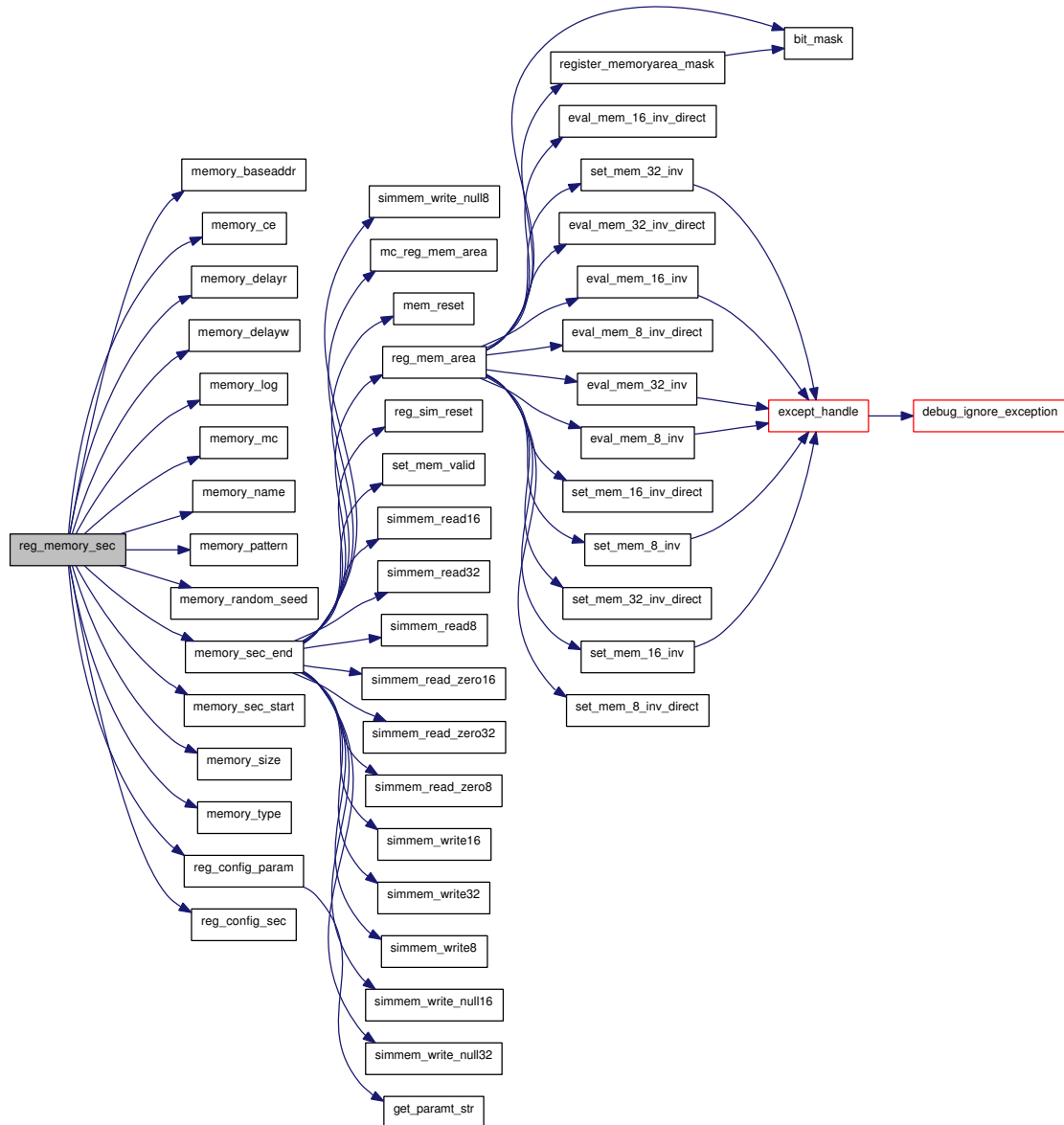
Functions

- void [reg_memory_sec](#) ()

5.132.1 Function Documentation

5.132.1.1 void reg_memory_sec ()

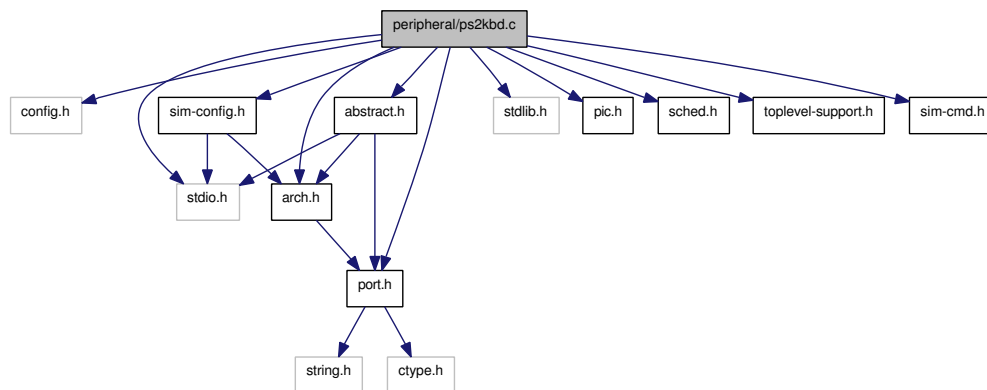
Here is the call graph for this function:



5.133 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]

5.133.1 Define Documentation

- 5.133.1.1 `#define KBD_BAUD_RATE 1200`
- 5.133.1.2 `#define KBD_CCMD_DKI 0xAD`
- 5.133.1.3 `#define KBD_CCMD_EKI 0xAE`
- 5.133.1.4 `#define KBD_CCMD_RCB 0x20`
- 5.133.1.5 `#define KBD_CCMD_ST1 0xAA`
- 5.133.1.6 `#define KBD_CCMD_ST2 0xAB`
- 5.133.1.7 `#define KBD_CCMD_WCB 0x60`
- 5.133.1.8 `#define KBD_CCMDBYTE_EN 0x10`
- 5.133.1.9 `#define KBD_CCMDBYTE_EN2 0x20`
- 5.133.1.10 `#define KBD_CCMDBYTE_INT 0x01`
- 5.133.1.11 `#define KBD_CCMDBYTE_INT2 0x02`
- 5.133.1.12 `#define KBD_CCMDBYTE_SYS 0x04`
- 5.133.1.13 `#define KBD_CCMDBYTE_XLAT 0x40`
- 5.133.1.14 `#define KBD_CTRL 4`
- 5.133.1.15 `#define KBD_DATA 0`
- 5.133.1.16 `#define KBD_KCMD_DK 0xF5`
- 5.133.1.17 `#define KBD_KCMD_ECHO 0xFF`
- 5.133.1.18 `#define KBD_KCMD_EK 0xF4`
- 5.133.1.19 `#define KBD_KCMD_RST 0xFF`
- 5.133.1.20 `#define KBD_KCMD_SRL 0xED`
- 5.133.1.21 `#define KBD_KRESP_ACK 0xFA`
- 5.133.1.22 `#define KBD_KRESP_ECHO 0xEE`
- 5.133.1.23 `#define KBD_KRESP_RSTOK 0xAA`
- 5.133.1.24 `#define KBD_MAX_BUF 0x100`
- 5.133.1.25 `#define KBD_SPACE 8`
- 5.133.1.26 `#define KBD_STATUS_A2 0x08`
- 5.133.1.27 `#define KBD_STATUS_IBF 0x02`

Generated on Sun Oct 12 09:05:03 2008 for OpenRisc. The OpenRISC 1000 Architectural Simulator by Doxygen

- 5.133.1.28 `#define KBD_STATUS_INH 0x10`

- 5.133.1.29 `#define KBD_STATUS_MOBF 0x20`

- 5.133.1.30 `#define KBD_STATUS_OBF 0x01`

5.133.2.6 `static void kbd_put (struct kbd_state * kbd, unsigned char c)` [static]

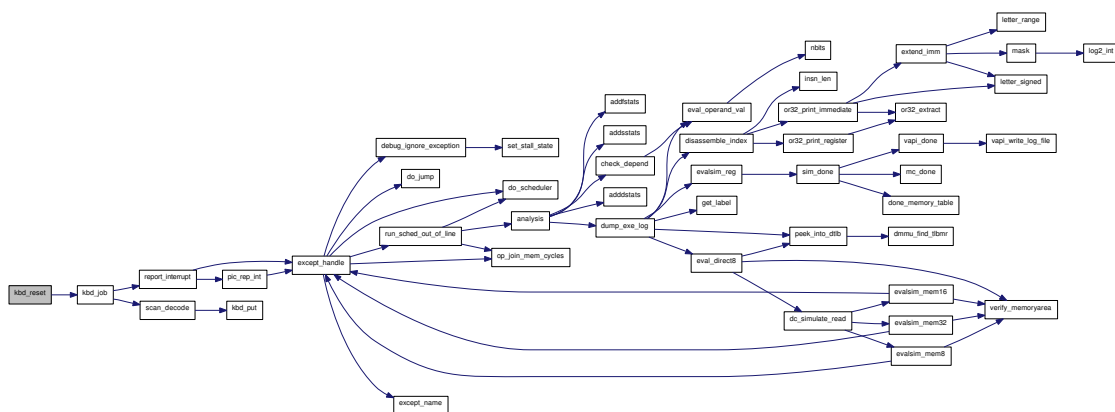
5.133.2.7 `static uint8_t kbd_read8 (oraddr_t addr, void * dat)` [static]

Here is the call graph for this function:



5.133.2.8 `static void kbd_reset (void * dat)` [static]

Here is the call graph for this function:



5.133.2.9 `static void kbd_rxfile (union param_val val, void * dat)` [static]

Set the keyboard input file

Free any previously allocated value.

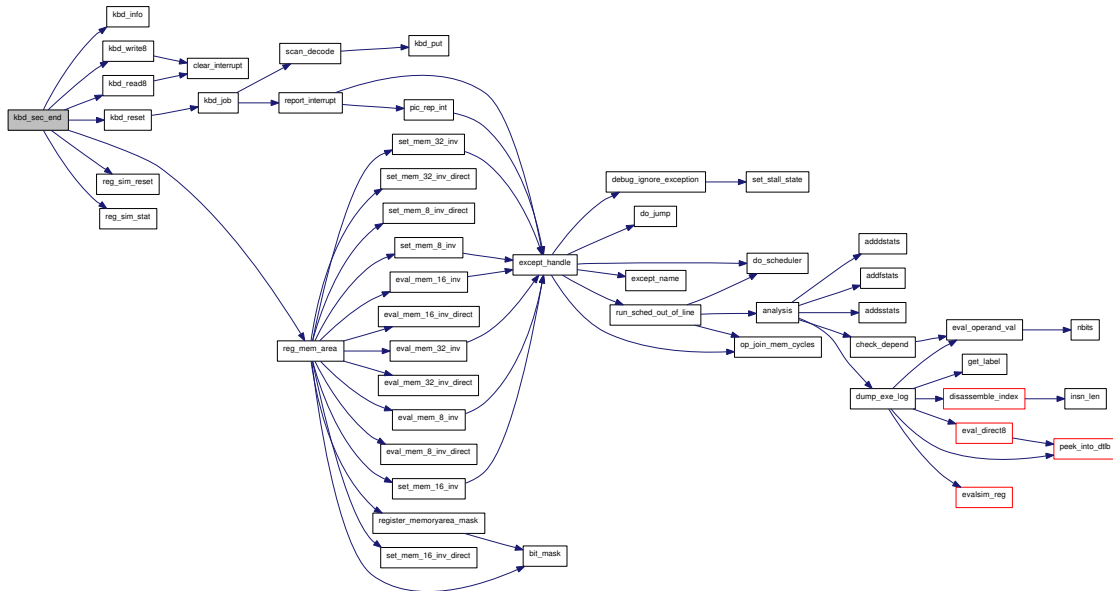
Parameters:

← *val* The value to use

← *dat* The `config` data structure

5.133.2.10 static void kbd_sec_end (void * dat) [static]

Here is the call graph for this function:



5.133.2.11 static void* kbd_sec_start () [static]

Initialize a new keyboard configuration

ALL parameters are set explicitly to default values.

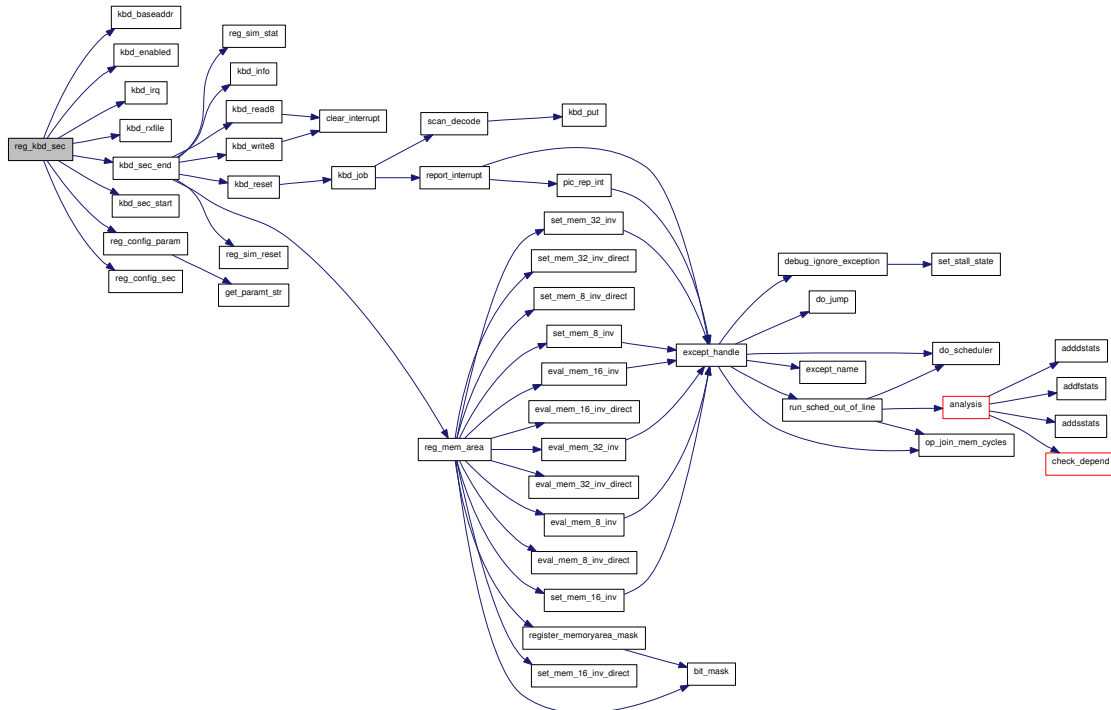
5.133.2.12 static void kbd_write8 (oraddr_t addr, uint8_t value, void * dat) [static]

Here is the call graph for this function:



5.133.2.13 void reg_kbd_sec ()

Here is the call graph for this function:



5.133.2.14 static void scan_decode (struct kbd_state * kbd, unsigned char c) [static]

Here is the call graph for this function:



5.133.3 Variable Documentation

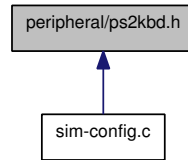
5.133.3.1 unsigned char code

5.133.3.2 const { ... } scan_table[128] [static]

5.133.3.3 unsigned char shift

5.134 peripheral/ps2kbd.h File Reference

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



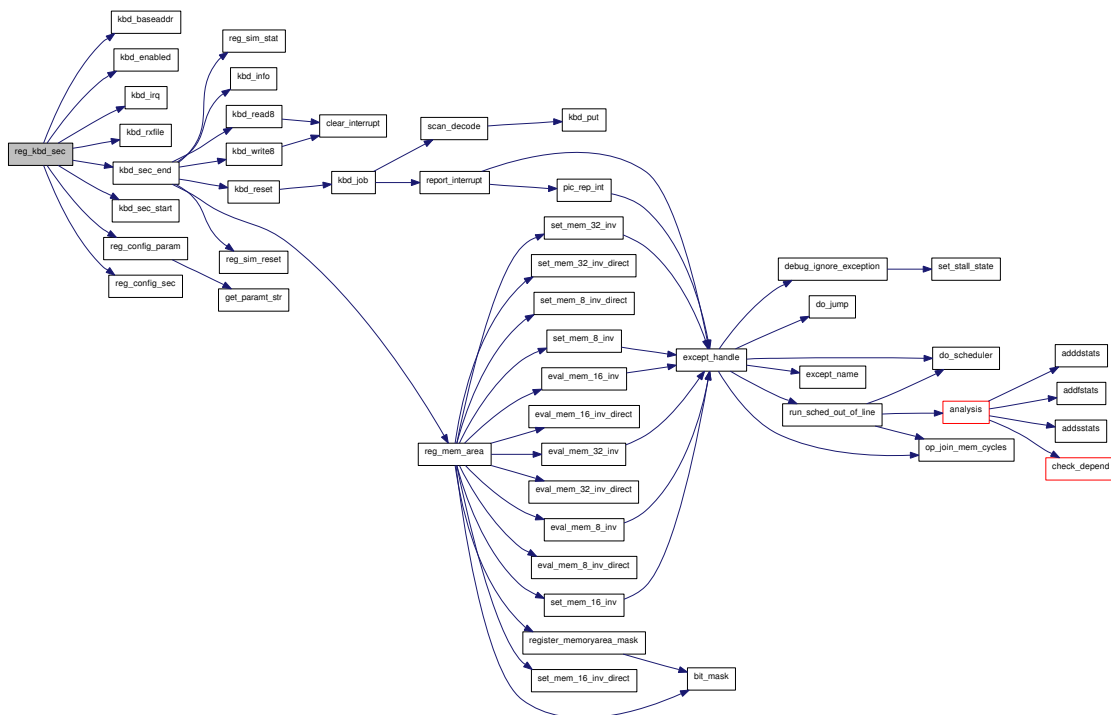
Functions

- void [reg_kbd_sec](#) ()

5.134.1 Function Documentation

5.134.1.1 void [reg_kbd_sec](#) ()

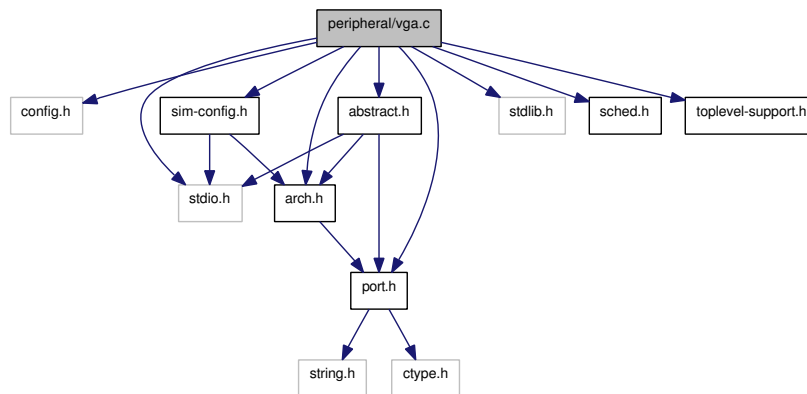
Here is the call graph for this function:



5.135 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` ()

5.135.1 Define Documentation

5.135.1.1 **#define VGA_ADDR_SPACE 1024**

5.135.1.2 **#define VGA_CLUTA 0x800**

5.135.1.3 **#define VGA_CLUTB 0xc00**

5.135.1.4 **#define VGA_CTRL 0x00**

5.135.1.5 **#define VGA_CTRL_CD 0x00000300**

5.135.1.6 **#define VGA_CTRL_PC 0x00000400**

5.135.1.7 **#define VGA_CTRL_VEN 0x00000001**

5.135.1.8 **#define VGA_HTIM 0x08**

5.135.1.9 **#define VGA_HVLEN 0x10**

5.135.1.10 **#define VGA_MASK 0xff**

5.135.1.11 **#define VGA_STAT 0x04**

5.135.1.12 **#define VGA_VBARA 0x14**

5.135.1.13 **#define VGA_VBARB 0x18**

5.135.1.14 **#define VGA_VTIM 0x0c**

5.135.2 Function Documentation

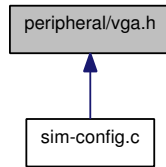
5.135.2.1 **void reg_vga_sec ()**

Create a new VGA configuration

ALL parameters are set explicitly to default values. Alternative naming for file parameter supported.

5.136 peripheral/vga.h File Reference

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



Functions

- void [reg_vga_sec](#) ()

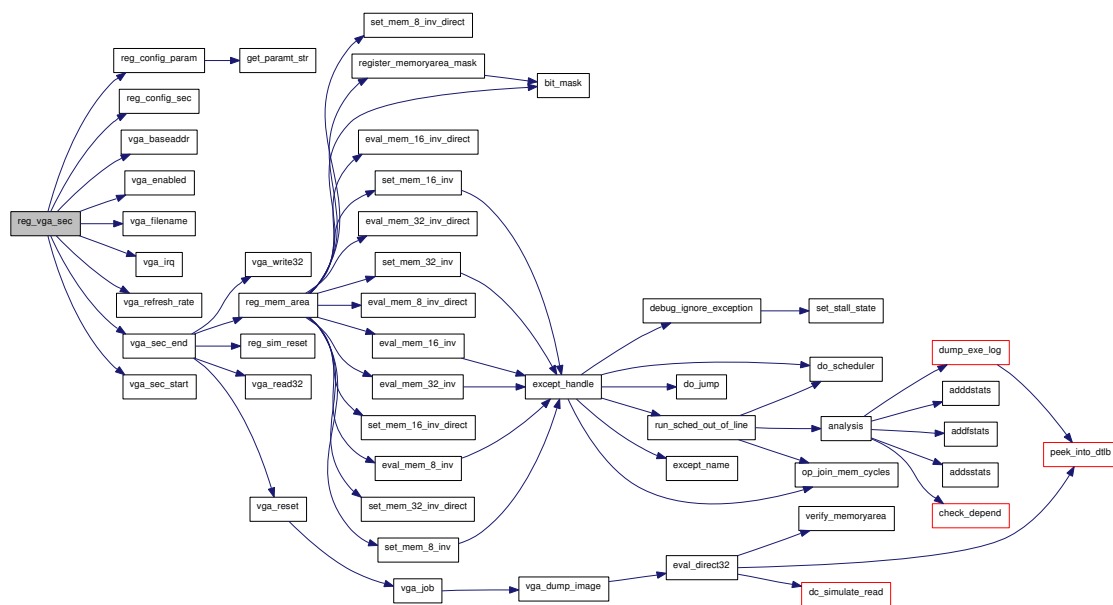
5.136.1 Function Documentation

5.136.1.1 void [reg_vga_sec](#) ()

Create a new VGA configuration

ALL parameters are set explicitly to default values. Alternative naming for file parameter supported.

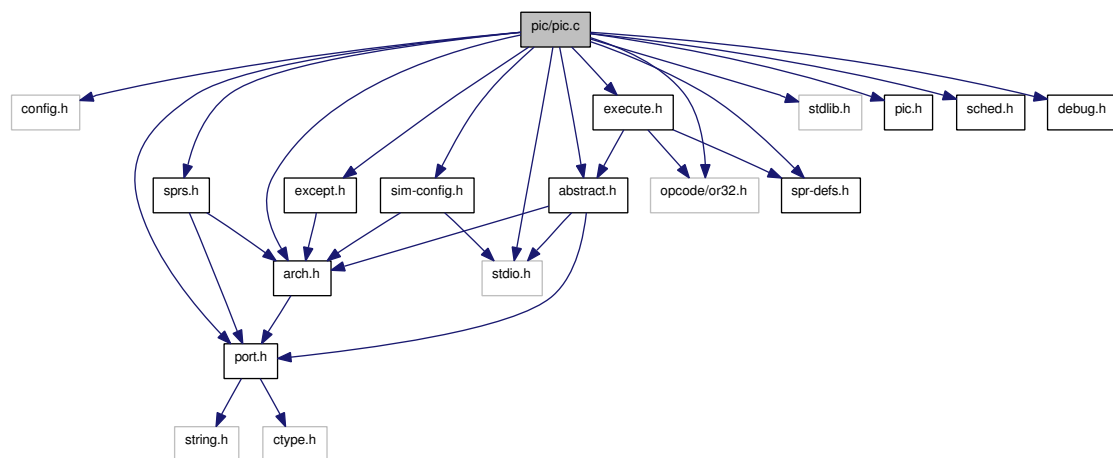
Here is the call graph for this function:



5.137 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 "debug.h"
```

Include dependency graph for pic.c:



Functions

- [DEFAULT_DEBUG_CHANNEL](#) (pic)
- 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](#)

5.137.1 Function Documentation

5.137.1.1 void [clear_interrupt](#) (int *line*)

5.137.1.2 [DEFAULT_DEBUG_CHANNEL](#) (pic)

5.137.1.3 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)

5.137.1.4 static void [pic_enabled](#) (union [param_val](#) *val*, void * *dat*) [*static*]

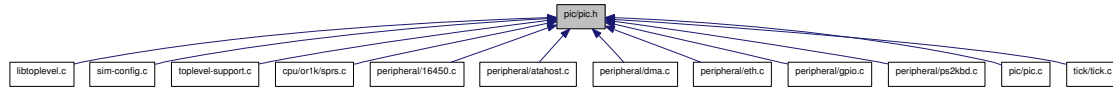
Enable or disable the programmable interrupt controller
Set the corresponding field in the UPR

Parameters:

- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

5.138 pic/pic.h File Reference

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



Functions

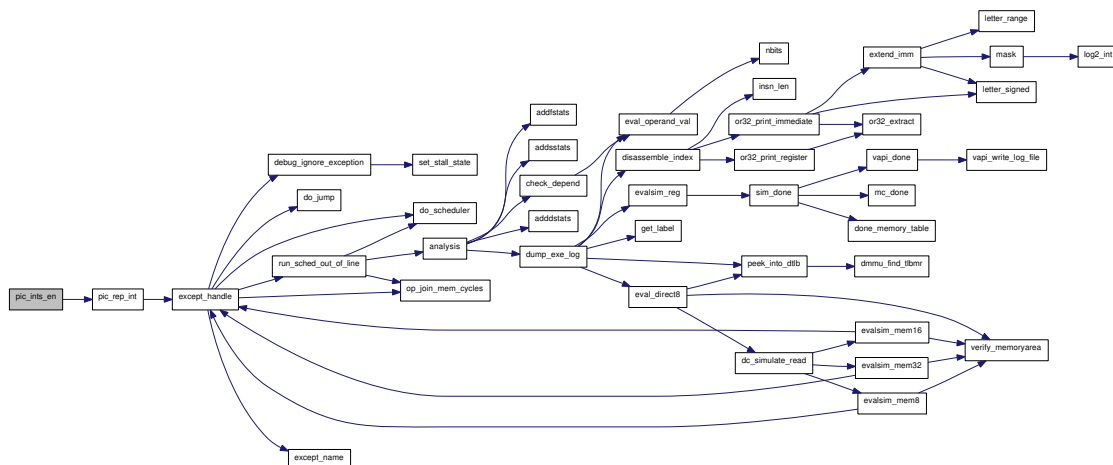
- void [pic_reset](#) ()
- void [report_interrupt](#) (int line)
- void [clear_interrupt](#) (int line)
- void [pic_ints_en](#) ()
- void [reg_pic_sec](#) ()

5.138.1 Function Documentation

5.138.1.1 void clear_interrupt (int line)

5.138.1.2 void pic_ints_en ()

Here is the call graph for this function:



5.138.1.3 void pic_reset ()

5.138.1.4 void reg_pic_sec ()

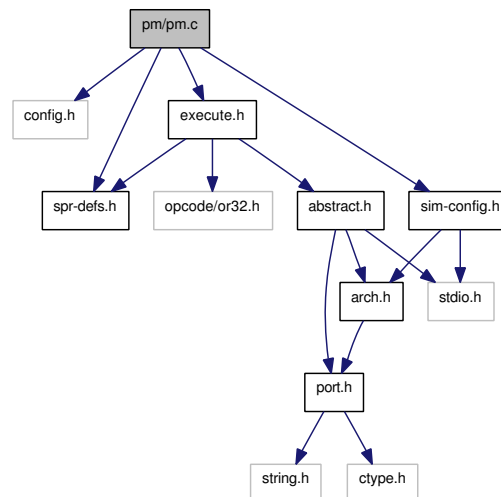
Initialize a new interrupt controller configuration

ALL parameters are set explicitly to default values in [init_defconfig\(\)](#)

5.139 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](#) ()

5.139.1 Function Documentation

5.139.1.1 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)

5.139.1.2 void [pm_reset](#) ()

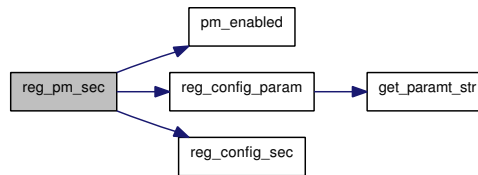
Reset power management

Initializes PMR register by clearing it.

5.139.1.3 void reg_pm_sec ()

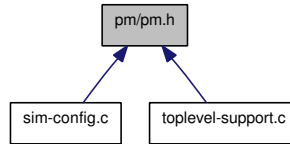
Set up a new power management configuration section

Here is the call graph for this function:



5.140 pm/pm.h File Reference

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



Functions

- void [pm_reset](#) ()
- void [reg_pm_sec](#) ()

5.140.1 Function Documentation

5.140.1.1 void pm_reset ()

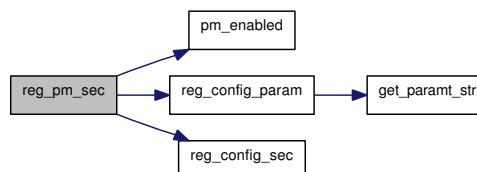
Reset power management

Initializes PMR register by clearing it.

5.140.1.2 void reg_pm_sec ()

Set up a new power management configuration section

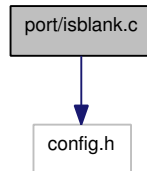
Here is the call graph for this function:



5.141 port/isblank.c File Reference

```
#include "config.h"
```

Include dependency graph for isblank.c:



Functions

- int [isblank](#) (int *c*)

5.141.1 Function Documentation

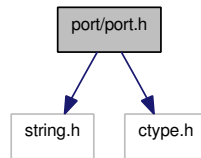
5.141.1.1 int isblank (int *c*)

5.142 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)

5.142.1 Define Documentation

5.142.1.1 #define [PRIx16](#) "x"

5.142.1.2 #define [PRIx8](#) "x"

5.142.2 Function Documentation

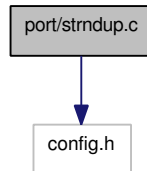
5.142.2.1 int [isblank](#) (int *c*)

5.142.2.2 char* [strndup](#) (const char * *s*, size_t *n*)

5.143 port/strndup.c File Reference

```
#include "config.h"
```

Include dependency graph for strndup.c:



Functions

- char * [strndup](#) (const char *s, size_t n)

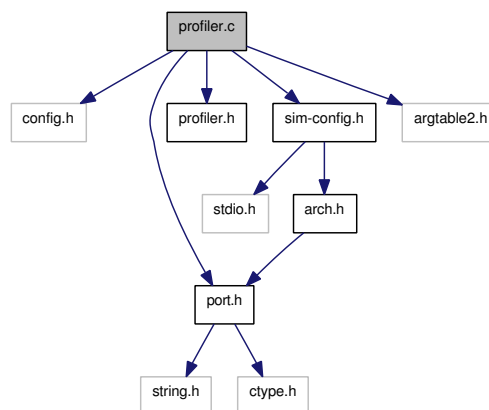
5.143.1 Function Documentation

5.143.1.1 char* strndup (const char * s, size_t n)

5.144 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

5.144.1 Define Documentation

5.144.1.1 #define MAX_STACK 1024

Maximum stack frames that can be profiled

5.144.2 Function Documentation

5.144.2.1 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.

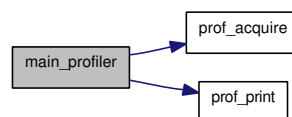
Parameters:

- ← *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:



5.144.2.2 int prof_acquire (const char * fprofname)

Acquire data from profiler file

Parameters:

- ← *fprofname* Data file to analyse

Returns:

0 on success, return code otherwise

5.144.2.3 `static void prof_print ()` [static]

5.144.2.4 `void prof_set (int _quiet, int _cumulative)`

5.144.3 Variable Documentation

5.144.3.1 `int cumulative = 0` [static]

Whether we are in cumulative mode

5.144.3.2 `FILE* fprof = 0` [static]

File to read from

5.144.3.3 `int maxstack = 0` [static]

Max depth

5.144.3.4 `int nfuncalls = 0` [static]

Number of covered calls

5.144.3.5 `int nstack = 0` [static]

Current depth

5.144.3.6 `int ntotcalls = 0` [static]

Number of total calls

5.144.3.7 `int prof_cycles = 0`

Global: current cycles

5.144.3.8 `struct func_struct prof_func[MAX_FUNCS]`

Global: data about functions

5.144.3.9 `int prof_nfuncs = 0`

Global: total number of functions

5.144.3.10 `int quiet = 0` [static]

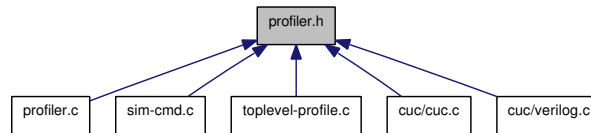
Whether we should not report warnings

5.144.3.11 `struct stack_struct stack[MAX_STACK]` `[static]`

Representation of the stack

5.145 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](#)

5.145.1 Define Documentation

5.145.1.1 #define MAX_FUNCS 1024

Maximum number of functions that can be profiled

5.145.2 Function Documentation

5.145.2.1 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:

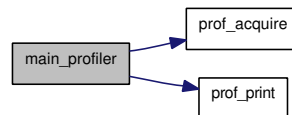
- ← *argc* Number of command args
- ← *argv* Vector of the command args

← *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:

**5.145.2.2 int prof_acquire (const char * *fprofname*)**

Acquire data from profiler file

Parameters:

← *fprofname* Data file to analyse

Returns:

0 on success, return code otherwise

5.145.2.3 void prof_set (int *_quiet*, int *_cumulative*)**5.145.3 Variable Documentation****5.145.3.1 int prof_cycles**

Global: current cycles

5.145.3.2 struct func_struct prof_func[MAX_FUNCS]

Global: data about functions

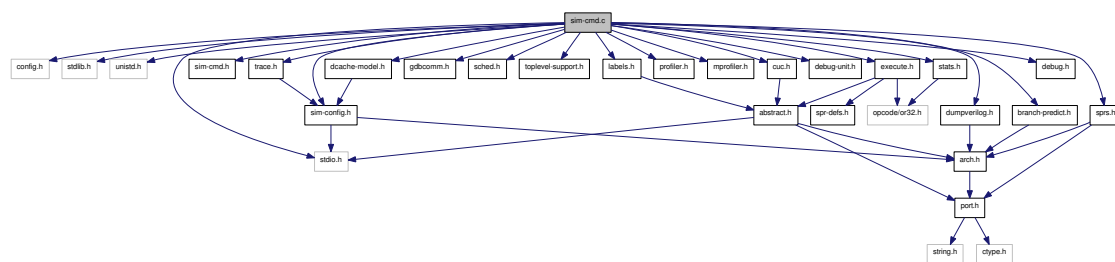
5.145.3.3 int prof_nfuncs

Global: total number of functions

5.146 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 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)

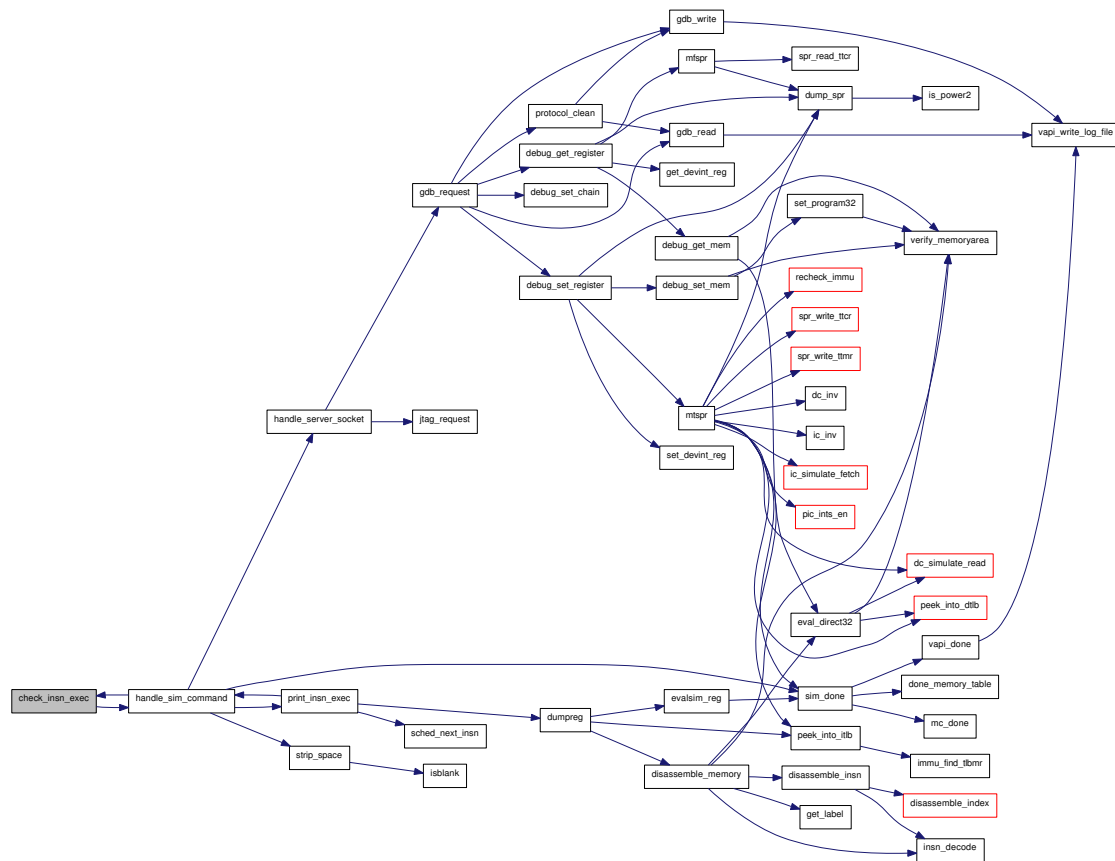
Variables

- static long long `to_insn_num`
- static struct `sim_stat` * `sim_stats` = NULL
- static struct `sim_command` `sim_commands` []

5.146.1 Function Documentation

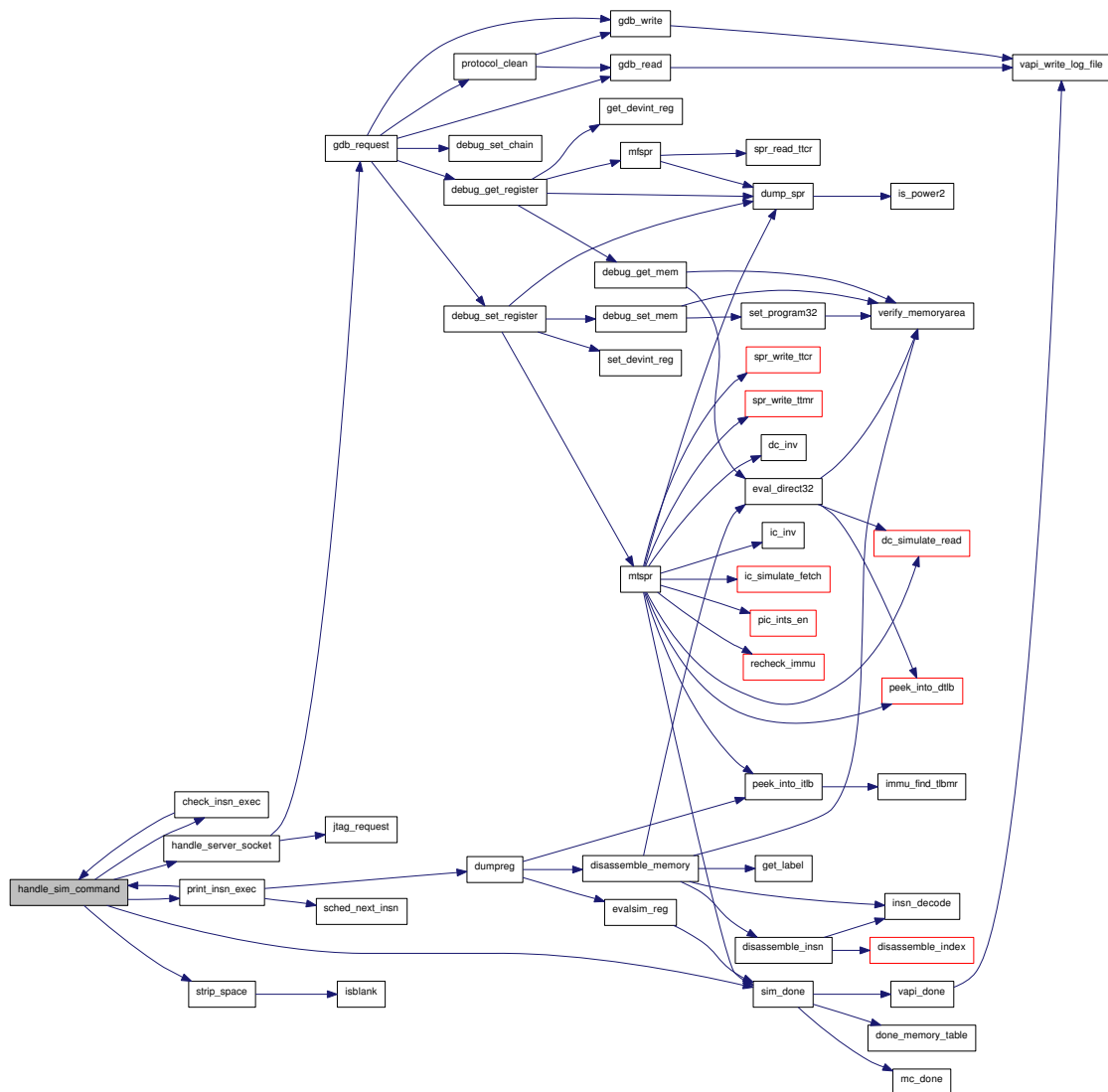
5.146.1.1 void check_insn_exec (void * dat)

Here is the call graph for this function:



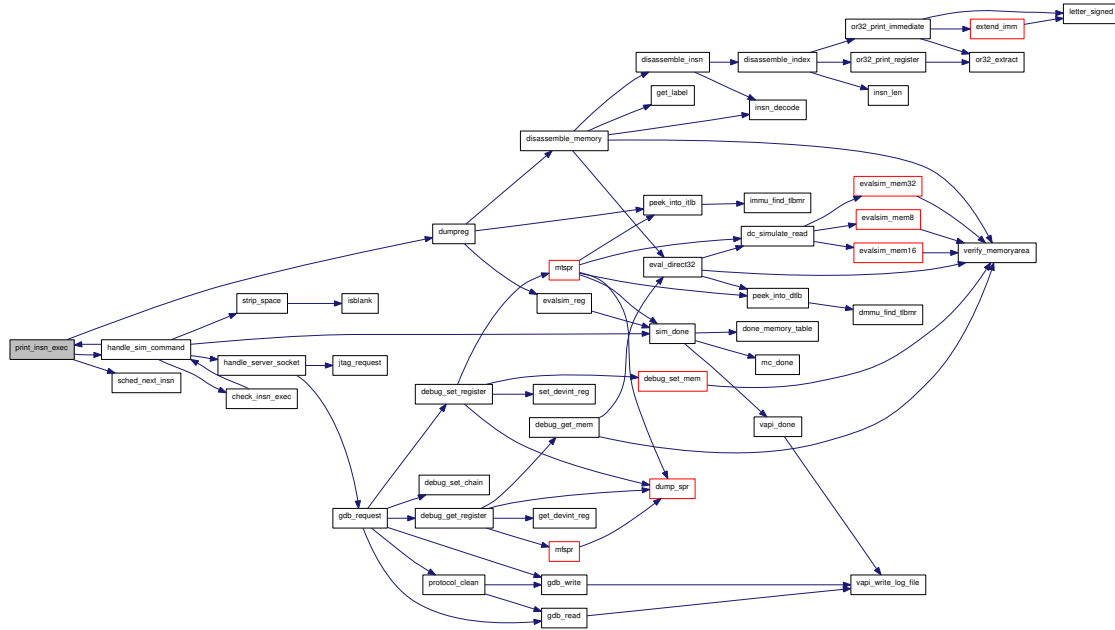
5.146.1.2 void handle_sim_command (void)

Here is the call graph for this function:



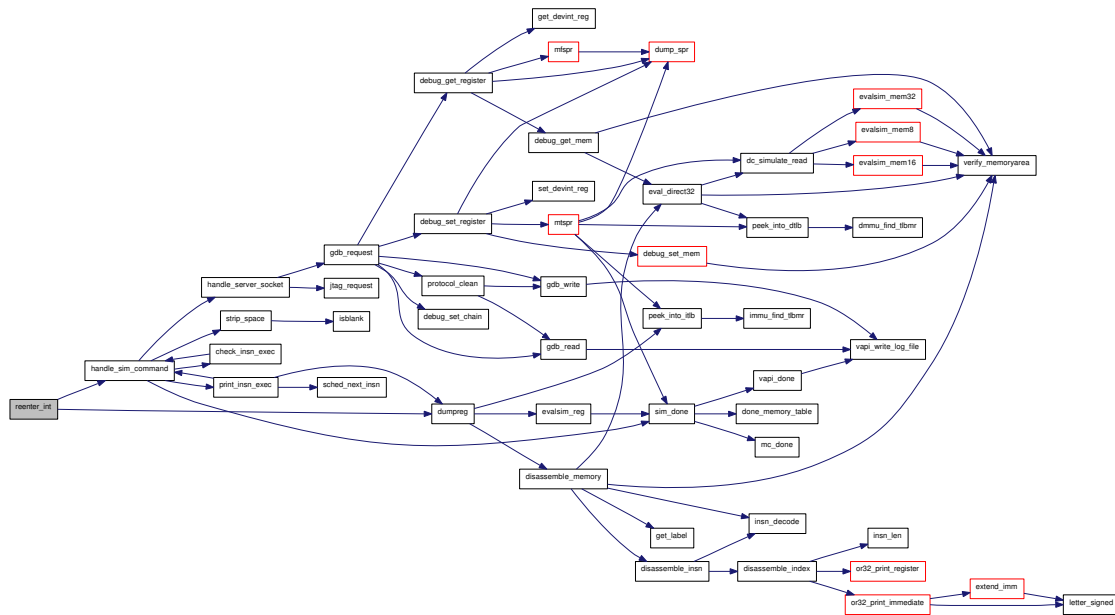
5.146.1.3 void print_insn_exec (void * dat)

Here is the call graph for this function:



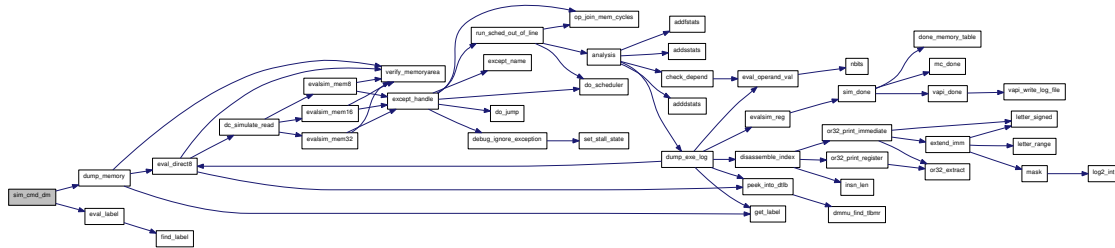
5.146.1.4 void reenter_int (void * dat)

Here is the call graph for this function:



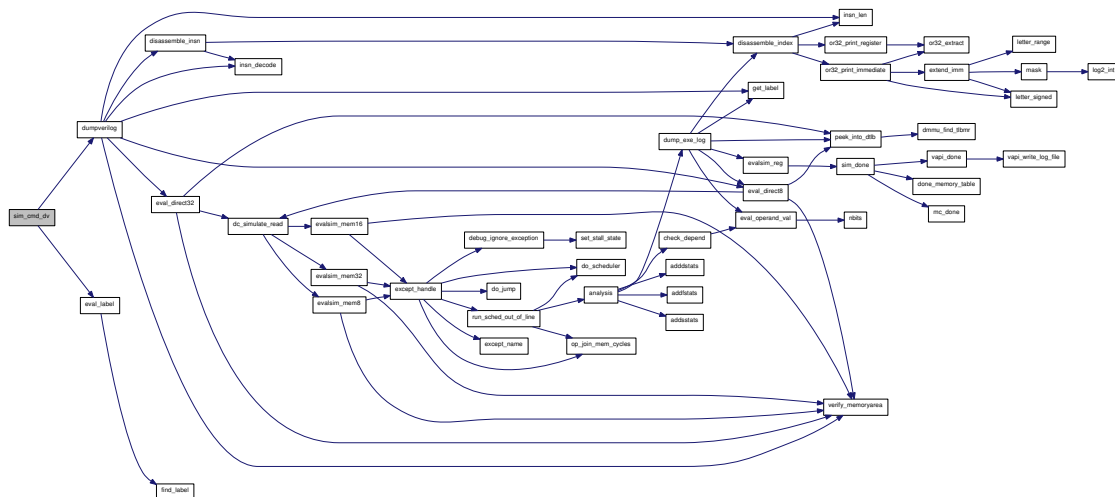
5.146.1.13 static int sim_cmd_dm (int argc, char ** argv) [static]

Here is the call graph for this function:



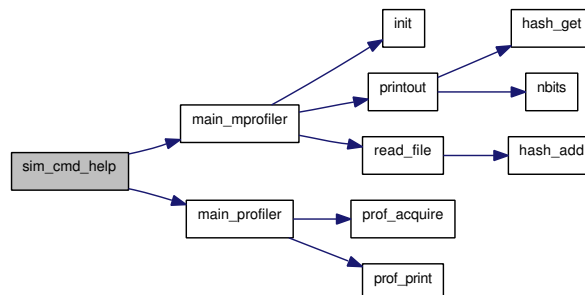
5.146.1.14 static int sim_cmd_dv (int argc, char ** argv) [static]

Here is the call graph for this function:



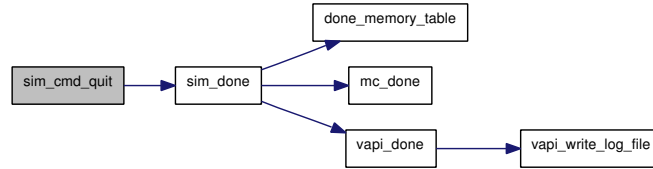
5.146.1.15 static int sim_cmd_help (int argc, char ** argv) [static]

Here is the call graph for this function:



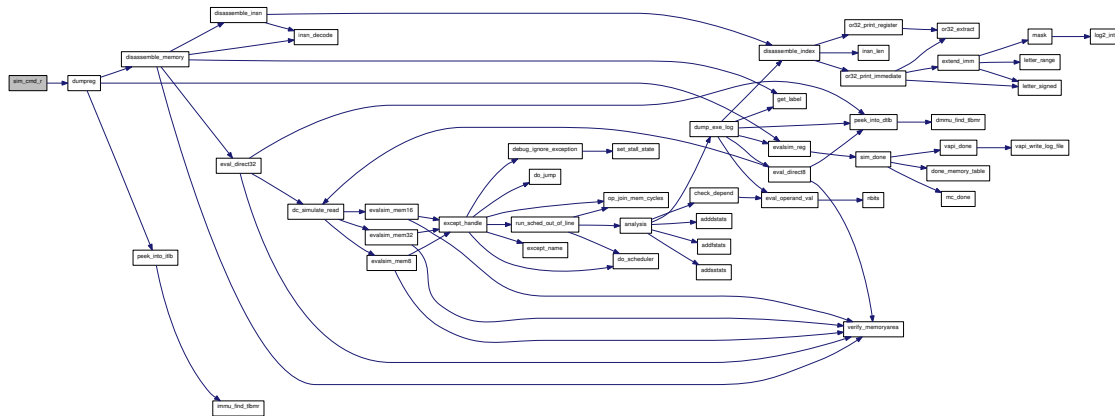
5.146.1.23 static int sim_cmd_quit (int argc, char ** argv) [static]

Here is the call graph for this function:



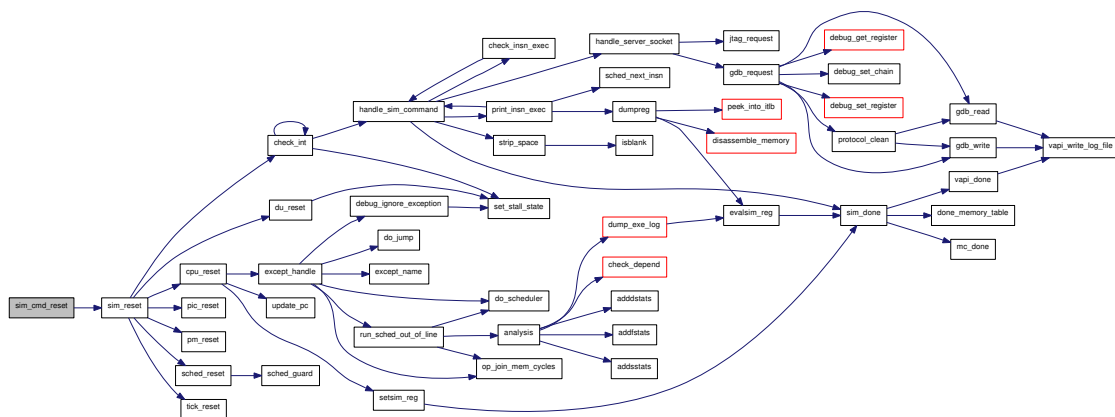
5.146.1.24 static int sim_cmd_r (int argc, char ** argv) [static]

Here is the call graph for this function:



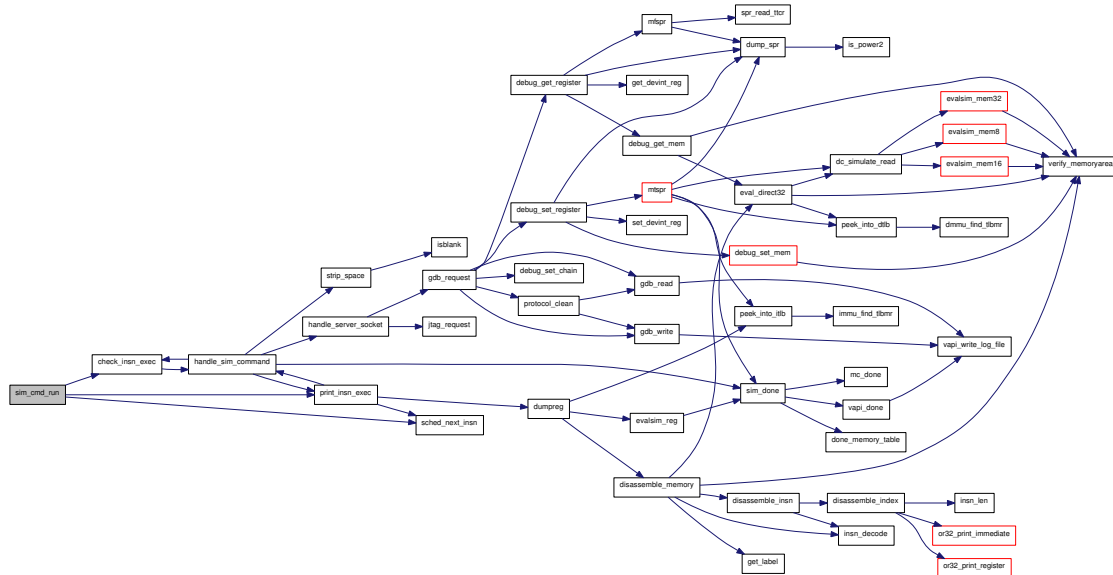
5.146.1.25 static int sim_cmd_reset (int argc, char ** argv) [static]

Here is the call graph for this function:



5.146.1.26 `static int sim_cmd_run (int argc, char ** argv) [static]`

Here is the call graph for this function:



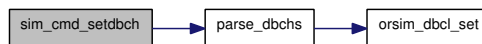
5.146.1.27 `static int sim_cmd_set (int argc, char ** argv) [static]`

Here is the call graph for this function:



5.146.1.28 `static int sim_cmd_setdbch (int argc, char ** argv) [static]`

Here is the call graph for this function:



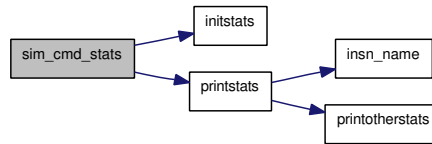
5.146.1.29 `static int sim_cmd_stall (int argc, char ** argv) [static]`

Here is the call graph for this function:



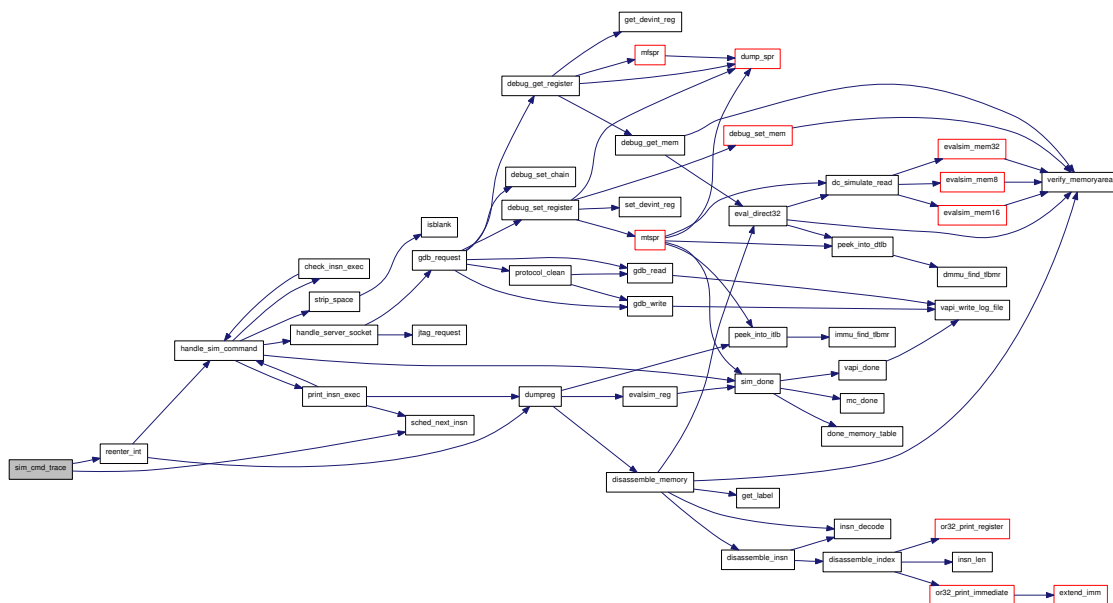
5.146.1.30 static int sim_cmd_stats (int argc, char ** argv) [static]

Here is the call graph for this function:



5.146.1.31 static int sim_cmd_trace (int argc, char ** argv) [static]

Here is the call graph for this function:



5.146.1.32 static int sim_cmd_unstall (int argc, char ** argv) [static]

Here is the call graph for this function:



5.146.1.33 static char* strip_space(char * *str*) [static]

Here is the call graph for this function:



5.146.2 Variable Documentation

5.146.2.1 struct sim_command sim_commands[] [static]

Initial value:

```

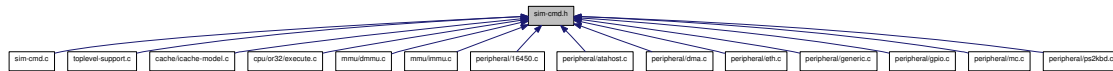
{
  {"q", sim_cmd_quit},
  {"help", sim_cmd_help},
  {"t", sim_cmd_trace},
  {"dm", sim_cmd_dm},
  {"dv", sim_cmd_dv},
  {"dh", sim_cmd_dh},
  {"pm", sim_cmd_pm},
  {"cm", sim_cmd_cm},
  {"pr", sim_cmd_pr},
  {"pc", sim_cmd_pc},
  {"breaks", sim_cmd_breaks},
  {"break", sim_cmd_break},
  {"r", sim_cmd_r},
  {"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}
}
  
```

5.146.2.2 struct sim_stat* sim_stats = NULL [static]

5.146.2.3 long long to_insn_num [static]

5.147 sim-cmd.h File Reference

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



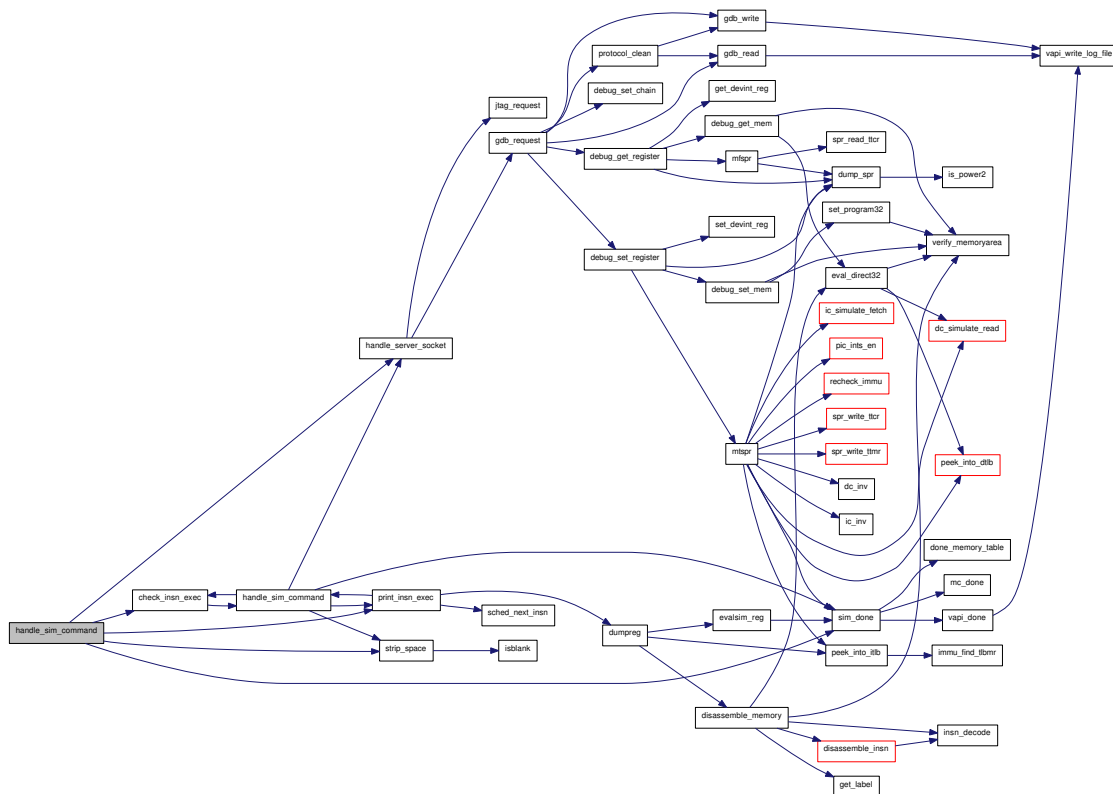
Functions

- void [handle_sim_command](#) (void)
- void [reg_sim_stat](#) (void(*stat_func)(void *dat), void *dat)

5.147.1 Function Documentation

5.147.1.1 void handle_sim_command (void)

Here is the call graph for this function:

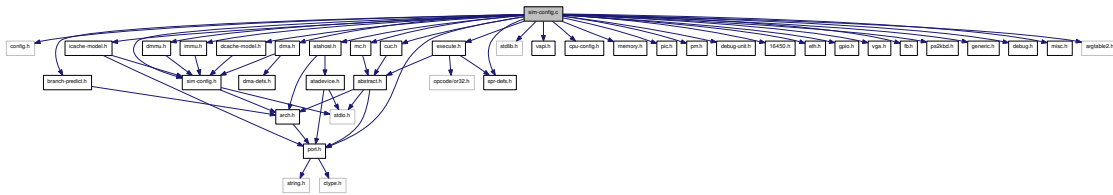


5.147.1.2 void reg_sim_stat (void(*) (void *dat) stat_func, void * dat)

5.148 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](#)

Defines

- #define [WARNING](#)(s) fprintf (stderr, "Warning: config.%s: %s\n", cur_section → [name](#), (s))
- #define [MERROR](#)(s) {fprintf (stderr, "ERROR: %s\n", s); if (runtime.sim.init) exit (1);}

Functions

- [DEFAULT_DEBUG_CHANNEL](#) ([config](#))
- 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)
- static char * [get_paramt_str](#) (enum [param_t](#) type)
- 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

5.148.1 Define Documentation

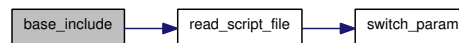
5.148.1.1 `#define MERROR(s) {fprintf(stderr, "ERROR: %s\n", s); if (runtime.sim.init) exit (1);}`

5.148.1.2 `#define WARNING(s) fprintf(stderr, "Warning: config.%s: %s\n", cur_section → name, (s))`

5.148.2 Function Documentation

5.148.2.1 `void base_include (union param_val val, void * dat)`

Here is the call graph for this function:



5.148.2.2 `DEFAULT_DEBUG_CHANNEL (config)`

5.148.2.3 `static char* get_paramt_str (enum param_t type) [static]`

5.148.2.4 `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:



5.148.2.5 `int parse_args (int argc, char * argv[])`

Parse the arguments for the standalone simulator

Updated by Jeremy Bennett to use argtable2.

Parameters:

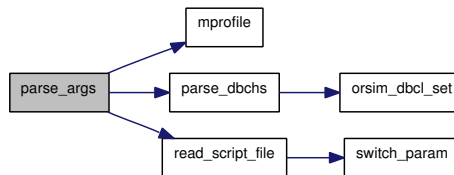
← `argc` Number of command args

← *argv* Vector of the command args

Returns:

0 on success, 1 on failure

Here is the call graph for this function:



5.148.2.6 void print_config (void)

Print the current configuration

Here is the call graph for this function:



5.148.2.7 static void read_script_file (const char *filename) [static]

Here is the call graph for this function:



5.148.2.8 void reg_config_param (struct config_section *sec, const char *param, enum param_t type, void(*) (union param_val, void *) param_cb)

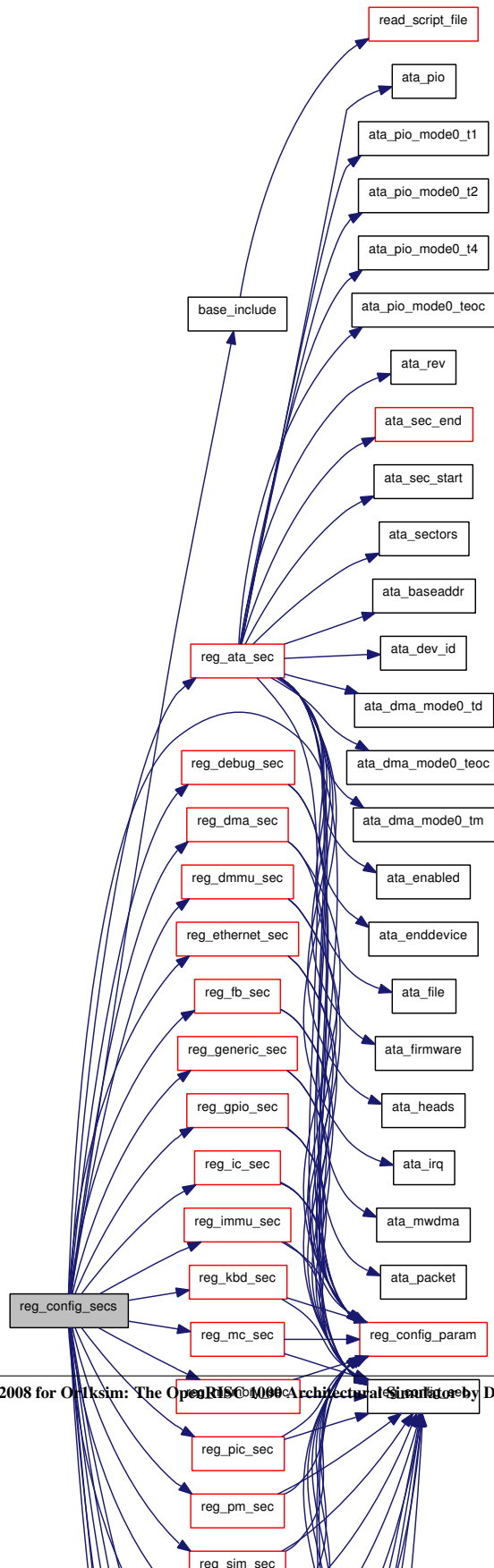
Here is the call graph for this function:



5.148.2.9 struct config_section* reg_config_sec (const char * section, void (*)(*)(void) sec_start, void (*)(void *) sec_end) [read]

5.148.2.10 void reg_config_secs (void)

Here is the call graph for this function:



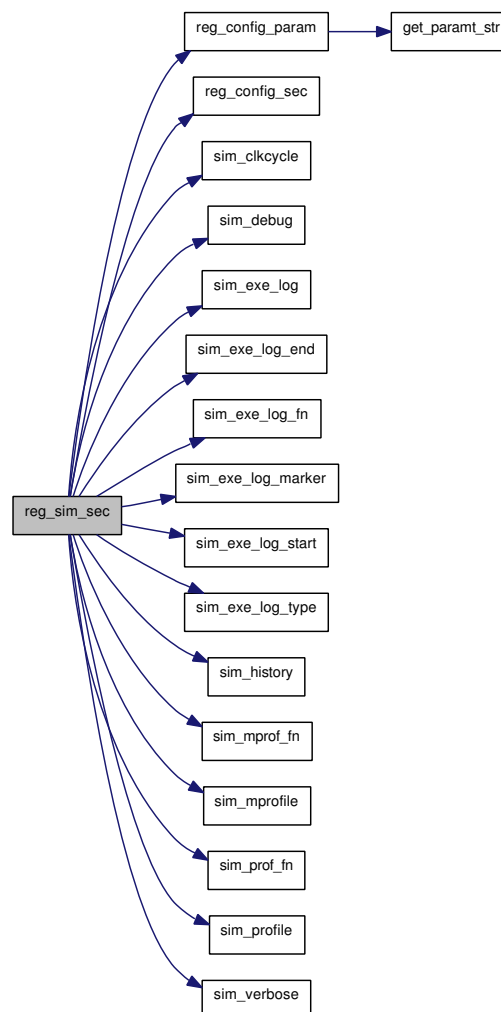
5.148.2.11 `static void reg_sim_sec ()` [static]

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:



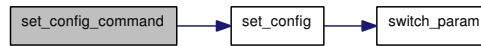
5.148.2.12 `static int set_config (int argc, char ** argv)` [static]

Here is the call graph for this function:



5.148.2.13 void set_config_command (int argc, char ** argv)

Here is the call graph for this function:

**5.148.2.14 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)

5.148.2.15 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)

5.148.2.16 void sim_exe_log (union param_val val, void * dat)**5.148.2.17 void sim_exe_log_end (union param_val val, void * dat)****5.148.2.18 void sim_exe_log_fn (union param_val val, void * dat)****5.148.2.19 void sim_exe_log_marker (union param_val val, void * dat)****5.148.2.20 void sim_exe_log_start (union param_val val, void * dat)****5.148.2.21 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)

5.148.2.22 void sim_history (union param_val val, void * dat)

5.148.2.23 void sim_mprof_fn (union param_val val, void * dat)

5.148.2.24 void sim_mprofile (union param_val val, void * dat)

5.148.2.25 void sim_prof_fn (union param_val val, void * dat)

5.148.2.26 void sim_profile (union param_val val, void * dat)

5.148.2.27 void sim_verbose (union param_val val, void * dat)

5.148.2.28 static void switch_param (char * param, struct config_param * cur_param)
[static]

5.148.3 Variable Documentation

5.148.3.1 struct config config

5.148.3.2 struct config_section* cur_section

5.148.3.3 struct runtime runtime

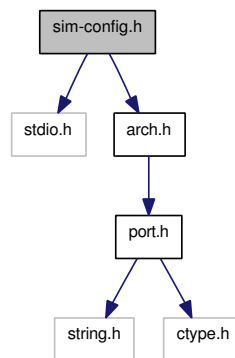
5.148.3.4 struct config_section* sections = NULL

5.149 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)
- #define [CONFIG_ERROR](#)(s) {fprintf (stderr, "ERROR: config.%.s:%.s\n", cur_section → name, s); if (runtime.sim.init) exit (1);}

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`

5.149.1 Define Documentation

5.149.1.1 `#define CHECK_INT_TIME 100000`

5.149.1.2 `#define CONFIG_ERROR(s) {fprintf (stderr, "ERROR: config.%s:%s\n", cur_section → name, s); if (runtime.sim.init) exit (1);}`

5.149.1.3 `#define EXE_LOG_HARDWARE 0`

5.149.1.4 `#define EXE_LOG_SIMPLE 1`

5.149.1.5 `#define EXE_LOG_SOFTWARE 2`

5.149.1.6 `#define MAX_SBUF_LEN 256`

5.149.1.7 `#define PRINTF(x...) fprintf (runtime.sim.fout, x)`

5.149.1.8 `#define STR_SIZE 256`

5.149.2 Enumeration Type Documentation

5.149.2.1 `enum param_t`

Enum of all possible paramter types

Enumerator:

paramt_none

paramt_str

paramt_word

paramt_int

paramt_longlong

paramt_addr

5.149.3 Function Documentation

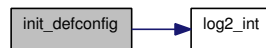
5.149.3.1 void init_defconfig (void)

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:



5.149.3.2 int parse_args (int argc, char * argv[])

Parse the arguments for the standalone simulator

Updated by Jeremy Bennett to use argtable2.

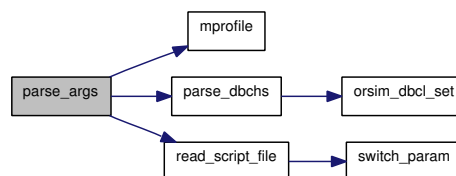
Parameters:

- ← *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:



5.149.3.3 void print_config (void)

Print the current configuration

Here is the call graph for this function:



5.149.3.4 void `reg_config_param` (struct `config_section` * *sec*, const char * *param*, enum `param_t` *type*, void(*)(`union param_val`, void *) *param_cb*)

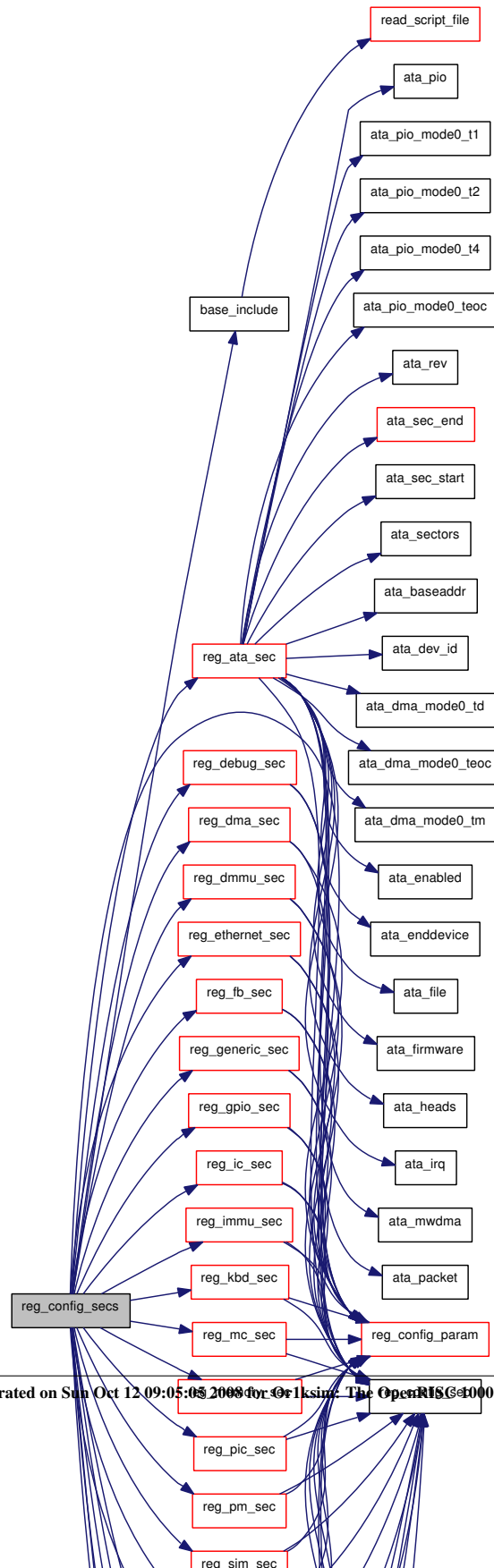
Here is the call graph for this function:



5.149.3.5 `struct config_section* reg_config_sec (const char * section, void (*)(void) sec_start, void (*)(void *) sec_end) [read]`

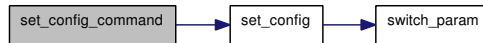
5.149.3.6 `void reg_config_secs ()`

Here is the call graph for this function:



5.149.3.7 void set_config_command (int argc, char ** argv)

Here is the call graph for this function:



5.149.4 Variable Documentation

5.149.4.1 struct config config

5.149.4.2 struct config_section* cur_section

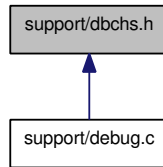
5.149.4.3 int do_stats

Whether we are doing statistical analysis. Globally available

5.149.4.4 struct runtime runtime

5.150 support/dbchs.h File Reference

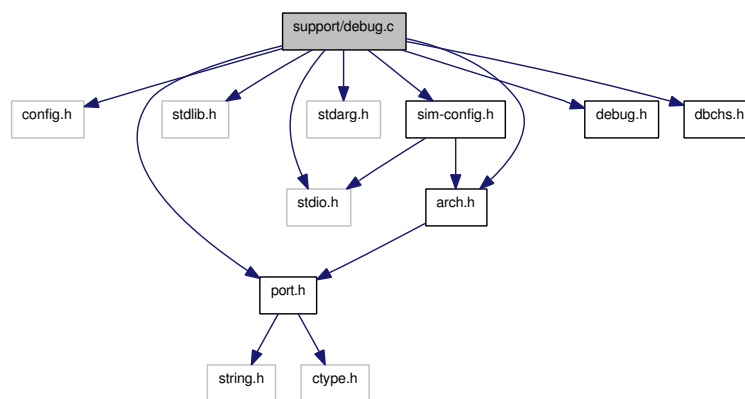
This graph shows which files directly or indirectly include this file:



5.151 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" }

5.151.1 Define Documentation

5.151.1.1 `#define __ORSIM_NO_DEC_DBCH`

5.151.1.2 `#define DECLARE_DEBUG_CHANNEL(dbch) __orsim_dbch_##dbch,`

5.151.1.3 `#define DECLARE_DEBUG_CHANNEL(dbch) char __orsim_dbch_##dbch[] =
"\0"#dbch;`

5.151.2 Function Documentation

5.151.2.1 `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

5.151.2.2 `void orsim_dbcl_set (enum __ORSIM_DEBUG_CLASS dbcl, char * dbch, int on)`

5.151.2.3 `void orsim_dbcl_set_name (enum __ORSIM_DEBUG_CLASS dbcl, const char * dbch,
int on)`

Here is the call graph for this function:



5.151.2.4 `void orsim_dbg_log (enum __ORSIM_DEBUG_CLASS dbcl, const char * dbch, const
char * function, const char * format, ...)`

5.151.2.5 `void parse_dbchs (const char * str)`

Here is the call graph for this function:



5.151.3 Variable Documentation

5.151.3.1 `char* __orsim_dbchs[] [static]`

5.151.3.2 `const char* debug_classes[] = { "trace", "fixme", "warn", "err" } [static]`

5.152 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,...)

5.152.1 Define Documentation

5.152.1.1 **#define** `__ORSIM_DBG_USE_FUNC __FUNCTION__`

5.152.1.2 **#define** `__ORSIM_DEBUG_LOG(args...) orsim_dbg_log(__dbcl, __dbch, __ORSIM_DBG_USE_FUNC, args); } } while(0)`

5.152.1.3 **#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
```

5.152.1.4 **#define** `__ORSIM_GET_DEBUGGING(dbcl, dbch) __ORSIM_GET_DEBUGGING##dbcl(dbch)`

5.152.1.5 **#define** `__ORSIM_GET_DEBUGGING_ERR(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_ERR))`

5.152.1.6 **#define** `__ORSIM_GET_DEBUGGING_FIXME(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_FIXME))`

5.152.1.7 **#define** `__ORSIM_GET_DEBUGGING_TRACE(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_TRACE))`

5.152.1.8 **#define** `__ORSIM_GET_DEBUGGING_WARN(dbch) ((dbch)[0] & (1 << __ORSIM_DBCL_WARN))`

5.152.1.9 **#define** `DECLARE_DEBUG_CHANNEL(dbch) extern char __orsim_dbch_##dbch[];`

5.152.1.10 **#define** `DEFAULT_DEBUG_CHANNEL(dbch)`

Value:

```
extern char __orsim_dbch_##dbch[ ]; \
static char * const __orsim_dbch___default = __orsim_dbch_##dbch;
```

- 5.152.1.11 `#define ERR __ORSIM_DPRINTF(_ERR, __orsim_dbch__default)`
- 5.152.1.12 `#define ERR_(ch) __ORSIM_DPRINTF(_ERR, __orsim_dbch_##ch)`
- 5.152.1.13 `#define ERR_ON(ch) __ORSIM_GET_DEBUGGING(_ERR, __orsim_dbch_##ch)`
- 5.152.1.14 `#define FIXME __ORSIM_DPRINTF(_FIXME, __orsim_dbch__default)`
- 5.152.1.15 `#define FIXME_(ch) __ORSIM_DPRINTF(_FIXME, __orsim_dbch_##ch)`
- 5.152.1.16 `#define FIXME_ON(ch) __ORSIM_GET_DEBUGGING(_FIXME, __orsim_dbch_-##ch)`
- 5.152.1.17 `#define TRACE __ORSIM_DPRINTF(_TRACE, __orsim_dbch__default)`
- 5.152.1.18 `#define TRACE_(ch) __ORSIM_DPRINTF(_TRACE, __orsim_dbch_##ch)`
- 5.152.1.19 `#define TRACE_ON(ch) __ORSIM_GET_DEBUGGING(_TRACE, __orsim_dbch_-##ch)`
- 5.152.1.20 `#define WARN __ORSIM_DPRINTF(_WARN, __orsim_dbch__default)`
- 5.152.1.21 `#define WARN_(ch) __ORSIM_DPRINTF(_WARN, __orsim_dbch_##ch)`
- 5.152.1.22 `#define WARN_ON(ch) __ORSIM_GET_DEBUGGING(_WARN, __orsim_dbch_##ch)`

5.152.2 Enumeration Type Documentation

5.152.2.1 `enum __ORSIM_DEBUG_CLASS`

Enumerator:

- `__ORSIM_DBCL_TRACE`
- `__ORSIM_DBCL_FIXME`
- `__ORSIM_DBCL_WARN`
- `__ORSIM_DBCL_ERR`

5.152.3 Function Documentation

5.152.3.1 `void debug(int level, const char *format, ...)`

Internal debug function

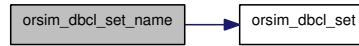
Print the message if the level is greater than or equal to that specified in the configuration.

Parameters:

- ← *level* The debug level of this message
- ← *format* Varargs format string
- ← ... The varargs required by the string

5.152.3.2 void void orsim_dbcl_set_name (enum __ORSIM_DEBUG_CLASS *dbcl*, const char * *dbch*, int *on*)

Here is the call graph for this function:

**5.152.3.3 void orsim_dbg_log (enum __ORSIM_DEBUG_CLASS *dbcl*, const char * *dbch*, const char * *function*, const char * *format*, ...)****5.152.3.4 void parse_dbchs (const char * *str*)**

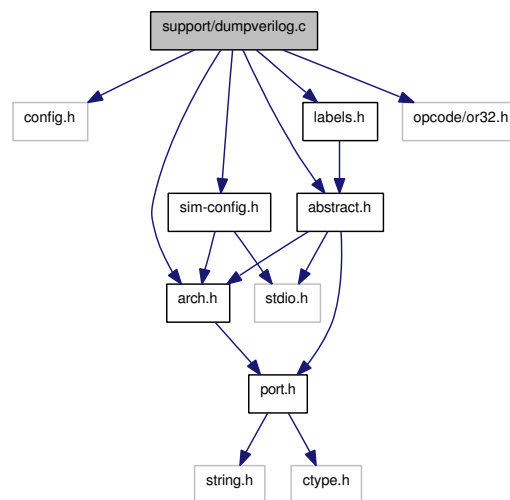
Here is the call graph for this function:



5.153 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\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"
- #define [LABELEND_CHAR](#) ":"

Functions

- void [dumpverilog](#) (char *verilog_modname, [oraddr_t](#) from, [oraddr_t](#) to)
- void [dumphex](#) ([oraddr_t](#) from, [oraddr_t](#) to)

5.153.1 Define Documentation

5.153.1.1 **#define DISWIDTH 25**

5.153.1.2 **#define DW 32**

5.153.1.3 **#define DWQ (DW/8)**

5.153.1.4 **#define LABELEND_CHAR ""**

5.153.1.5 **#define OR1K_MEM_VERILOG_FOOTER ""**
begin\n\n **if** (ce && ~we) **begin**\n\n dataout <= #1 mem[addr];\n\n disout <= #1
dis[addr];\n\n \$display("\nor1k_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("\nor1k_mem: writing mem[%0d]:%0h
dis: %0s", addr, mem[addr], dis[addr]);\n\n **end**\n\n **endmodule**\n\n"

5.153.1.6 **#define OR1K_MEM_VERILOG_HEADER(MODNAME, FROMADDR, TOADDR,
DISWIDTH)**

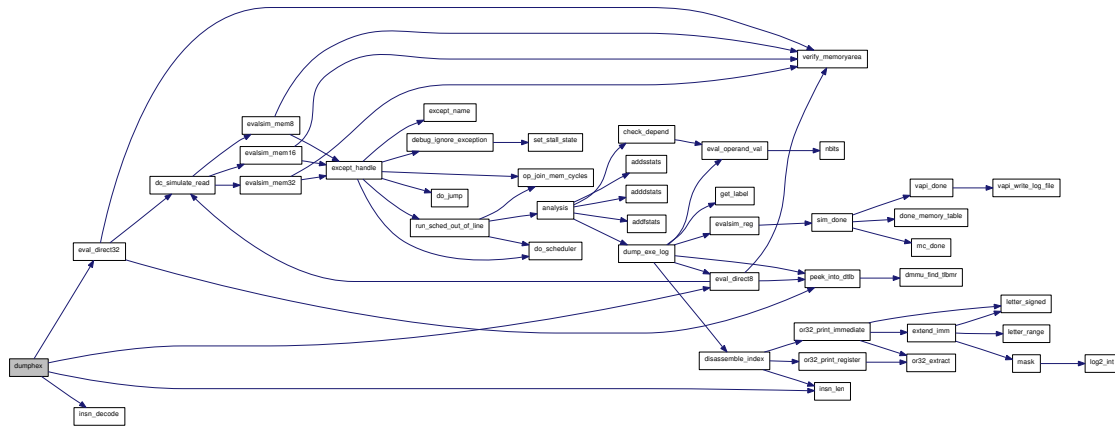
Value:

```
"\n\n"
#include "general.h"\n\n\n"
`timescale 1ns/100ps\n\n\n"
// Simple dw-wide Sync SRAM with initial content generated by orlksim.\n\n\n"
// All control, data in and addr signals are sampled at rising clock edge \n\n\n"
// Data out is not registered. Address bits specify dw-word (narrowest \n\n\n"
// addressed data is not byte but dw-word !). \n\n\n"
// There are still some bugs in generated output (dump word aligned regions)\n\n\n"
module %s(clk, data, addr, ce, we, disout);\n\n\n"
parameter dw = 32;\n\n\n"
parameter amin = %d;\n\n\n"
parameter amax = %d;\n\n\n"
input clk;\n\n\n"
inout [dw-1:0] data;\n\n\n"
input [31:0] addr;\n\n\n"
input ce;\n\n\n"
input we;\n\n\n"
output [%d:0] disout;\n\n\n"
reg [%d:0] disout;\n\n\n"
reg [dw-1:0] mem [amax:amin];\n\n\n"
reg [%d:0] dis [amax:amin];\n\n\n"
reg [dw-1:0] dataout;\n\n\n"
tri [dw-1:0] data = (ce && ~we) ? dataout : 'bz;\n\n\n"
initial begin\n\n", MODNAME, FROMADDR, TOADDR, DISWIDTH-1, DISWIDTH-1,
```

5.153.2 Function Documentation

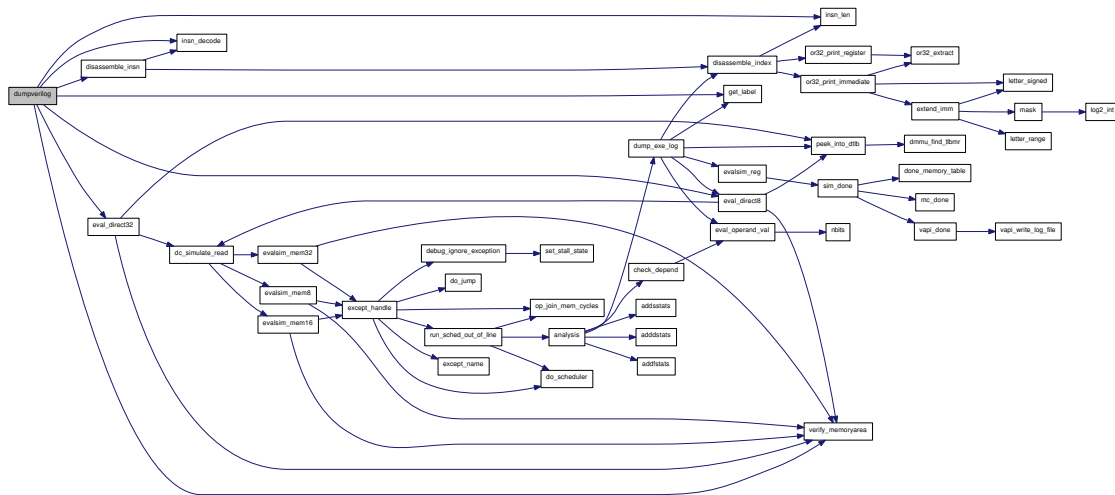
5.153.2.1 void dumphex (oraddr_t from, oraddr_t to)

Here is the call graph for this function:



5.153.2.2 void dumpverilog (char * verilog_modname, oraddr_t from, oraddr_t to)

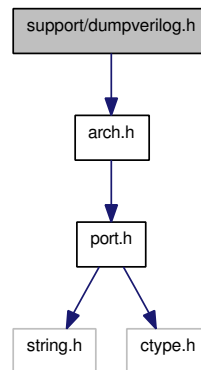
Here is the call graph for this function:



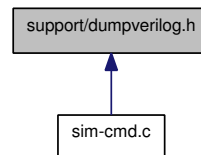
5.154 support/dumpverilog.h File Reference

```
#include "arch.h"
```

Include dependency graph for dumpverilog.h:



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



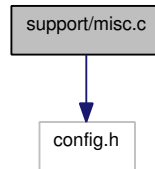
Functions

- void `dumpverilog` (char *verilog_modname, `oraddr_t` from, `oraddr_t` to)
- void `dumphex` (`oraddr_t` from, `oraddr_t` to)

5.155 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*)

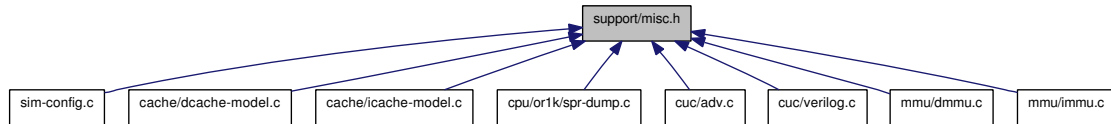
5.155.1 Function Documentation

5.155.1.1 [int is_power2](#) (int *x*)

5.155.1.2 [int log2_int](#) (unsigned long *x*)

5.156 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)

5.156.1 Function Documentation

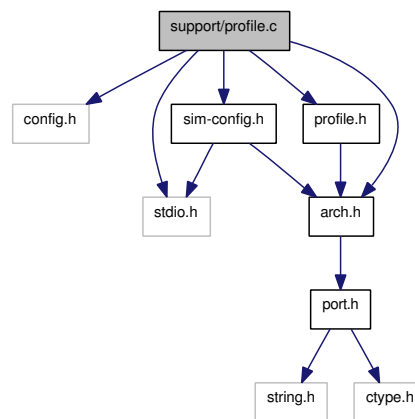
5.156.1.1 int is_power2 (int x)

5.156.1.2 int log2_int (unsigned long x)

5.157 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)

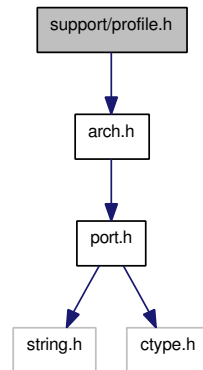
5.157.1 Function Documentation

5.157.1.1 void `mprofile` (`oraddr_t memaddr`, unsigned char type)

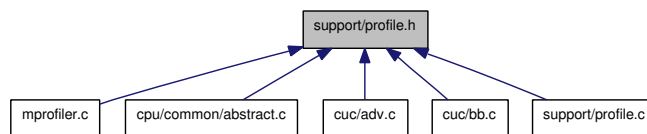
5.158 support/profile.h File Reference

```
#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)

5.158.1 Define Documentation

5.158.1.1 `#define MPROF_16 16`

5.158.1.2 `#define MPROF_32 32`

5.158.1.3 `#define MPROF_8 8`

5.158.1.4 `#define MPROF_FETCH 4`

5.158.1.5 `#define MPROF_READ 1`

5.158.1.6 `#define MPROF_WRITE 2`

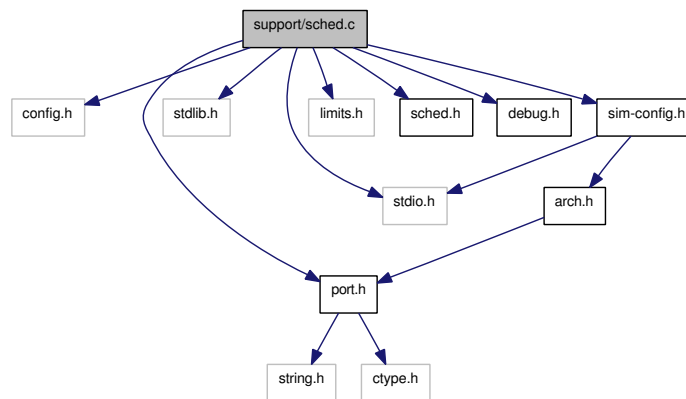
5.158.2 Function Documentation

5.158.2.1 `void mprofile (oraddr_t memaddr, unsigned char type)`

5.159 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 [SCHED_HEAP_SIZE](#) 128
- #define [SCHED_TIME_MAX](#) INT32_MAX

Functions

- [DEFAULT_DEBUG_CHANNEL](#) (sched)
- [DECLARE_DEBUG_CHANNEL](#) (sched_jobs)
- void [sched_guard](#) (void *dat)
- void [sched_reset](#) (void)
- void [sched_init](#) (void)
- void [do_scheduler](#) (void)
- static void [sched_print_jobs](#) (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, const char *func)
- void [sched_next_insn](#) (void(*func)(void *), void *dat)

Variables

- struct [scheduler_struct](#) scheduler

5.159.1 Define Documentation

5.159.1.1 `#define SCHED_HEAP_SIZE 128`

5.159.1.2 `#define SCHED_TIME_MAX INT32_MAX`

5.159.2 Function Documentation

5.159.2.1 `DECLARE_DEBUG_CHANNEL (sched_jobs)`

5.159.2.2 `DEFAULT_DEBUG_CHANNEL (sched)`

5.159.2.3 `void do_scheduler (void)`

5.159.2.4 `void sched_add (void(*) (void *) job_func, void * job_param, int32_t job_time, const char * func)`

Here is the call graph for this function:



5.159.2.5 `void sched_find_remove (void(*) (void *) job_func, void * dat, const char * func)`

5.159.2.6 `void sched_guard (void * dat)`

5.159.2.7 `void sched_init (void)`

Here is the call graph for this function:



5.159.2.8 `void sched_next_insn (void(*) (void *) func, void * dat)`

5.159.2.9 `static void sched_print_jobs (void) [static]`

5.159.2.10 `void sched_reset (void)`

Here is the call graph for this function:

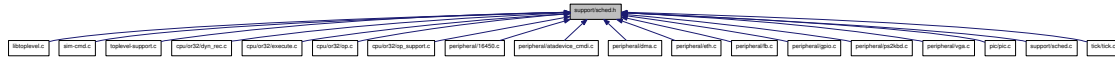


5.159.3 Variable Documentation

5.159.3.1 `struct scheduler_struct scheduler`

5.160 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, __FUNCTION__)

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, const char *func)
- 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](#)

5.160.1 Define Documentation

5.160.1.1 #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

5.160.1.2 #define [SCHED_FIND_REMOVE](#)(f, p) sched_find_remove(f, p, __FUNCTION__)

Macro to remove a job from the scheduler

5.160.2 Function Documentation

5.160.2.1 void do_scheduler ()

5.160.2.2 void sched_add (void(*) (void *) job_func, void * job_param, int32_t job_time, const char * func)

Here is the call graph for this function:



5.160.2.3 void sched_find_remove (void(*) (void *) job_func, void * dat, const char * func)

5.160.2.4 void sched_init ()

Here is the call graph for this function:



5.160.2.5 void sched_next_insn (void(*) (void *) func, void * dat)

5.160.2.6 void sched_reset ()

Here is the call graph for this function:



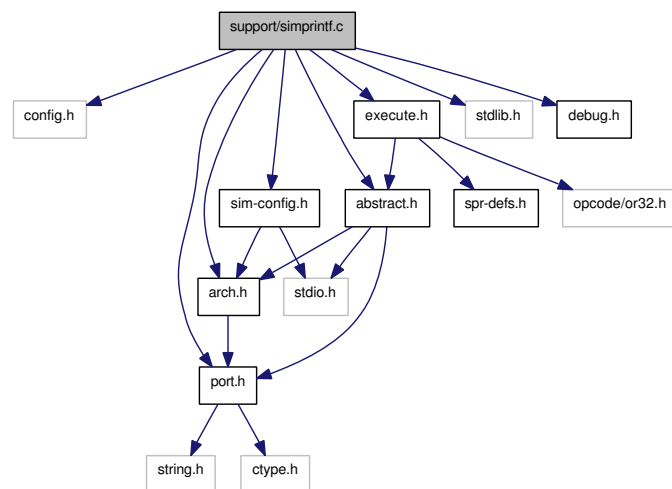
5.160.3 Variable Documentation

5.160.3.1 struct scheduler_struct scheduler

5.161 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

- [DEFAULT_DEBUG_CHANNEL](#) (simprintf)
- static char * [simgetstr](#) (oraddr_t stackaddr, unsigned long regparam)
- void [simprintf](#) (oraddr_t stackaddr, unsigned long regparam)

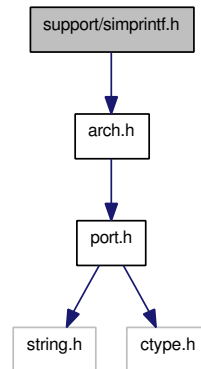
Variables

- char [fmtstr](#) [FMTLEN]

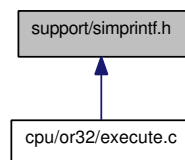
5.162 support/simprintf.h File Reference

```
#include "arch.h"
```

Include dependency graph for simprintf.h:



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



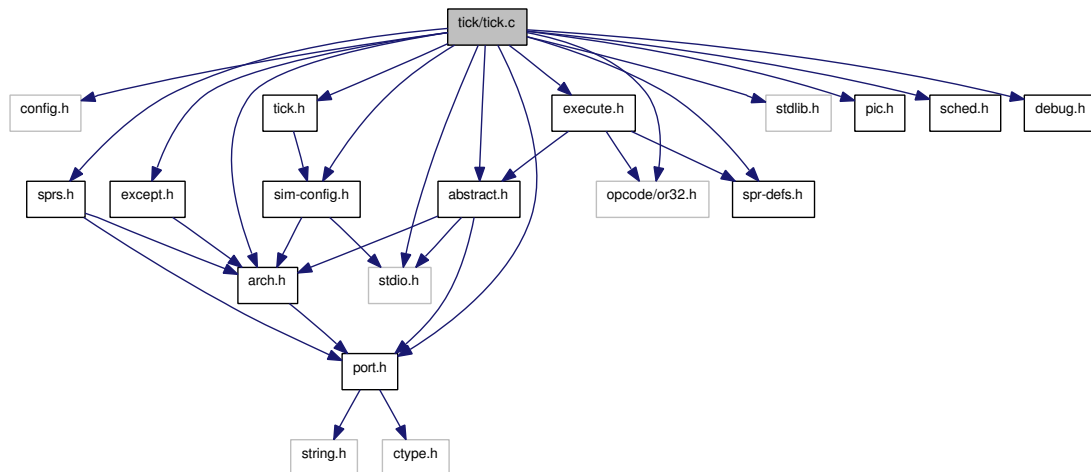
Functions

- void [simprintf](#) ([oraddr_t](#) stackaddr, unsigned long regparam)

5.163 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 "debug.h"
```

Include dependency graph for tick.c:



Functions

- [DEFAULT_DEBUG_CHANNEL](#) (tick)
- 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`)
- `uorreg_t spr_read_ttc` ()

Variables

- static long long `cycles_start` = 0
- int `tick_count`

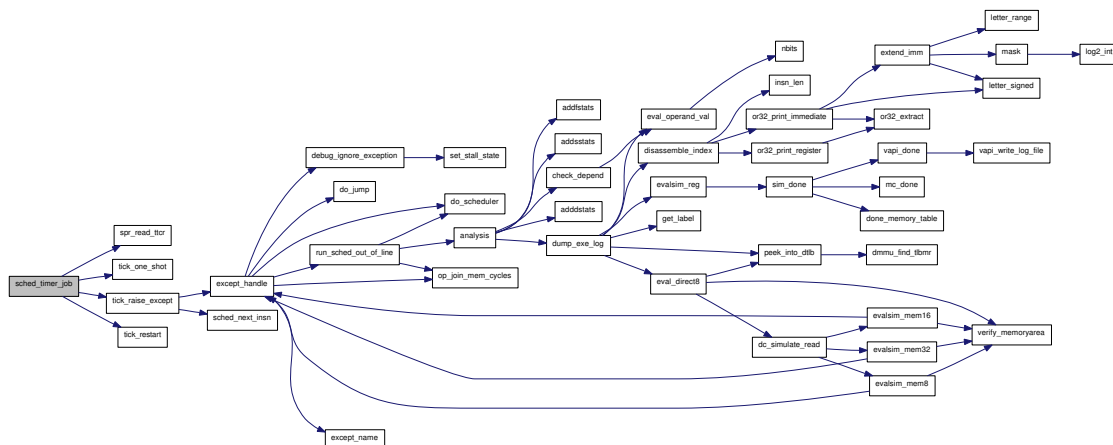
5.163.1 Function Documentation

5.163.1.1 DEFAULT_DEBUG_CHANNEL (tick)

5.163.1.2 static void `sched_timer_job` (`uorreg_t prev_ttmr`) [static]

Schedules the timer jobs

Here is the call graph for this function:

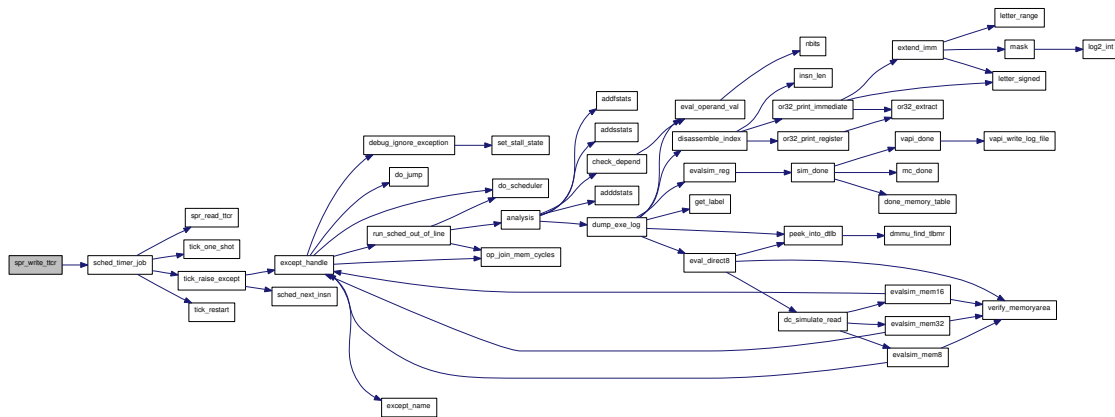


5.163.1.3 `uorreg_t spr_read_ttc` ()

5.163.1.4 void `spr_write_ttc` (`uorreg_t value`)

Handles a write to the ttc spr

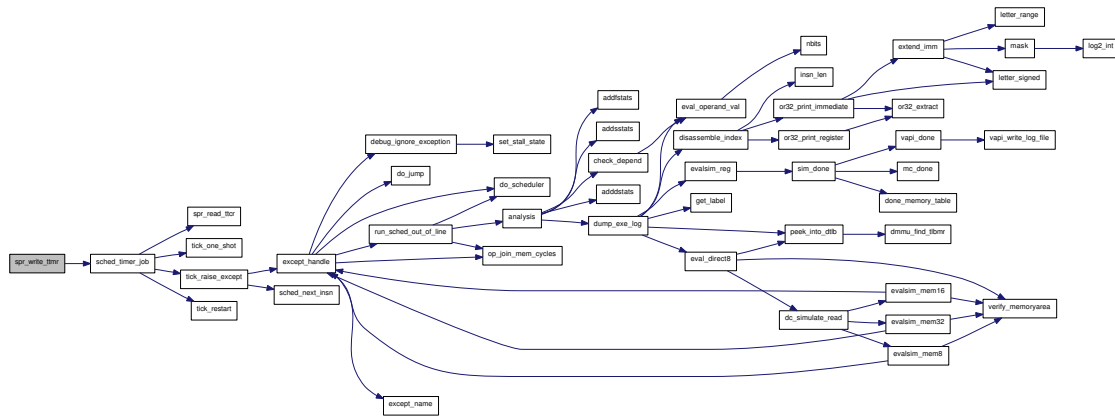
Here is the call graph for this function:



5.163.1.5 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:



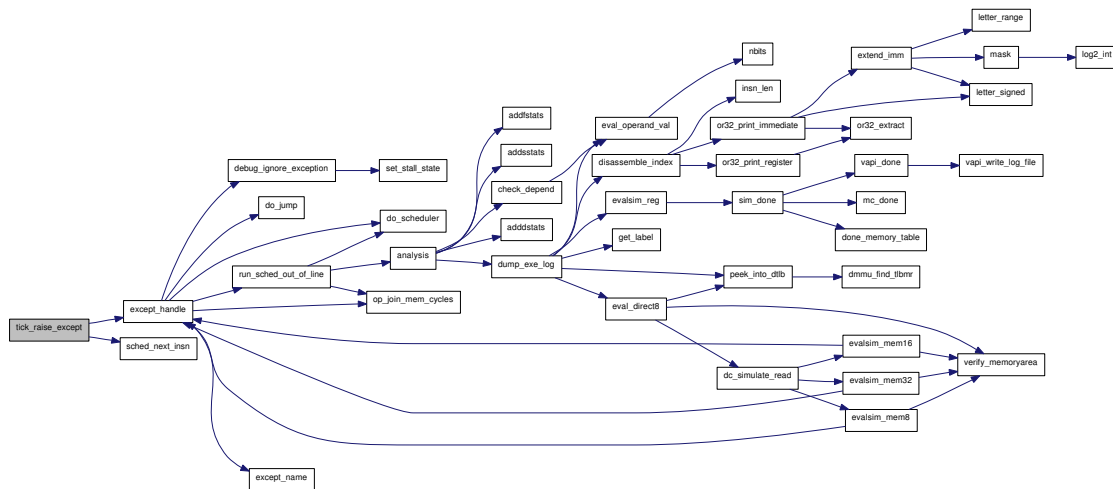
5.163.1.6 static void tick_one_shot (void * dat) [static]

Stops the timer

5.163.1.7 static void tick_raise_except (void * dat) [static]

Raises a timer exception

Here is the call graph for this function:



5.163.1.8 void tick_reset (void)

Reset. It initializes TTCR register.

5.163.1.9 static void tick_restart (void * *dat*) [static]

Restarts the tick timer

5.163.2 Variable Documentation

5.163.2.1 long long cycles_start = 0 [static]

When did the timer start to count

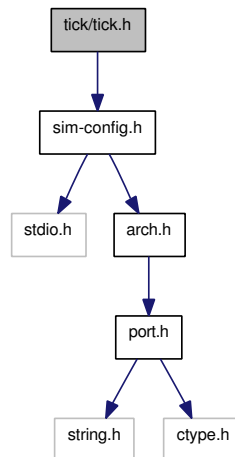
5.163.2.2 int tick_count

Indicates if the timer is actually counting. Needed to simulate one-shot mode correctly

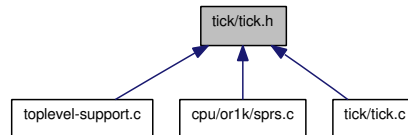
5.164 tick/tick.h File Reference

```
#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](#) ()

5.164.1 Function Documentation

5.164.1.1 [uorreg_t spr_read_ttc](#) ()

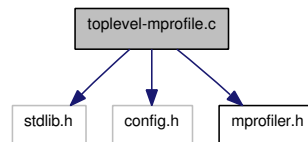
5.164.1.2 [void spr_write_ttc](#) ([uorreg_t value](#))

Handles a write to the ttc spr

5.165 toplevel-mprofile.c File Reference

```
#include <stdlib.h>
#include "config.h"
#include "mprofiler.h"
```

Include dependency graph for toplevel-mprofile.c:



Functions

- `int main (int argc, char *argv[])`

5.165.1 Function Documentation

5.165.1.1 int main (int argc, char * argv[])

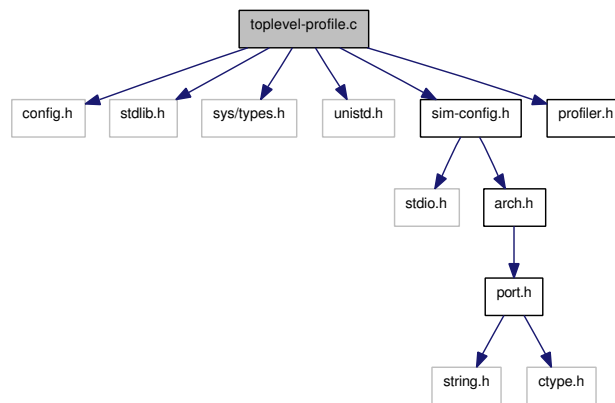
Here is the call graph for this function:



5.166 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:



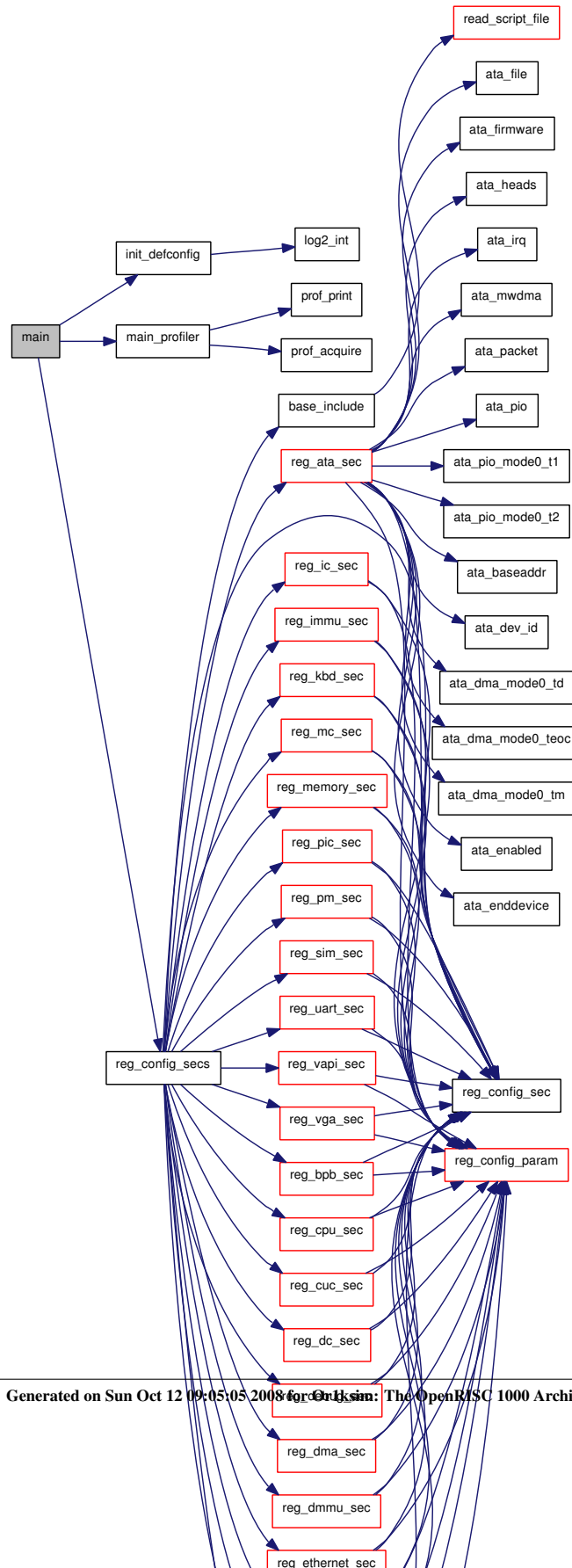
Functions

- `int main (int argc, char *argv[])`

5.166.1 Function Documentation

5.166.1.1 `int main (int argc, char * argv[])`

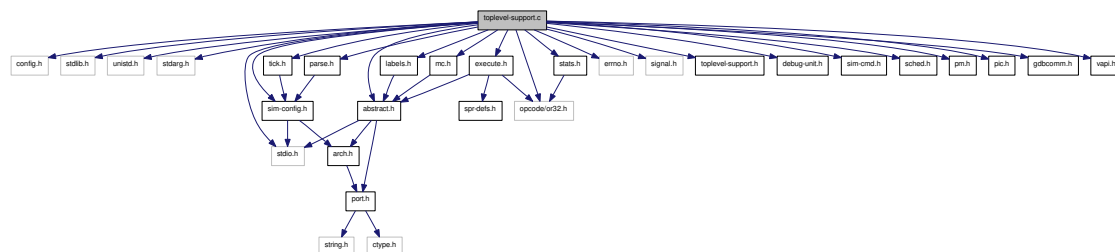
Here is the call graph for this function:



5.167 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 "vapi.h"
#include "abstract.h"
#include "mc.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

5.167.1 Function Documentation

5.167.1.1 void `check_int` (void * *dat*)

Routine poll to see if interaction is needed

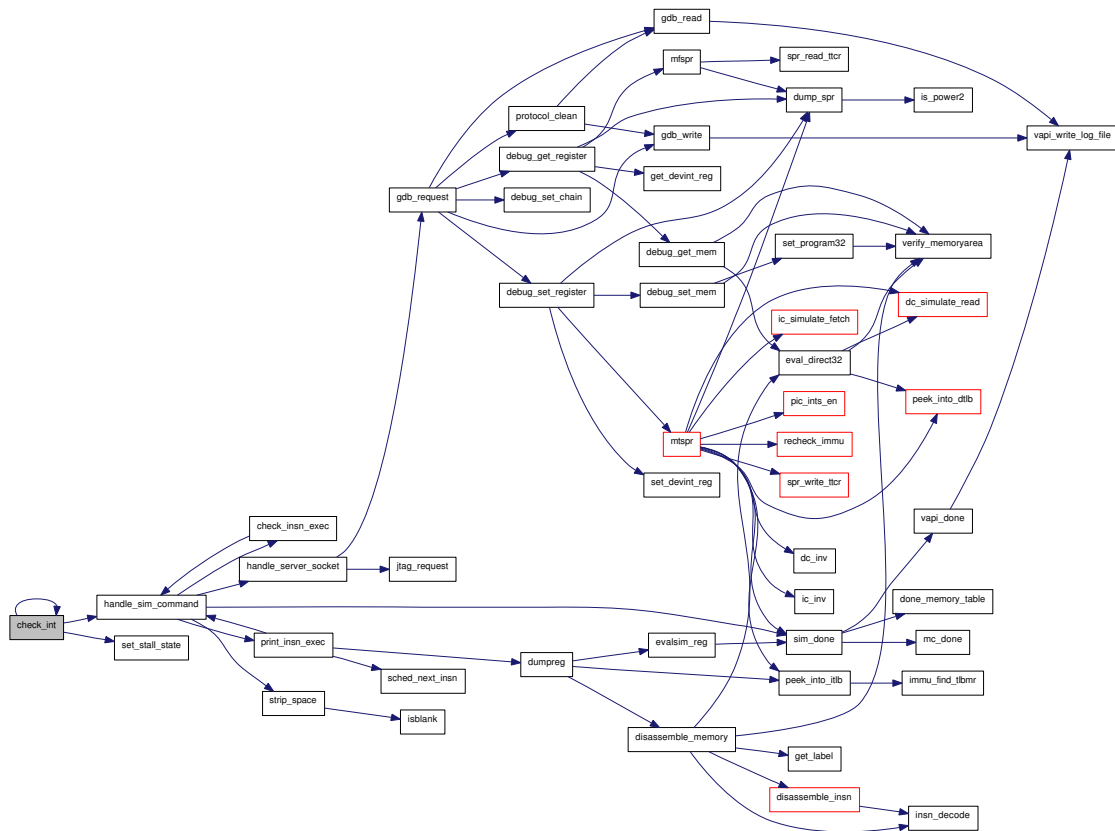
This is most likely to happen due to a ctrl-C. However when the -i flag is specified, the simulator starts up ready for interaction.

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:



5.167.1.2 void ctrl_c (int *signum*)

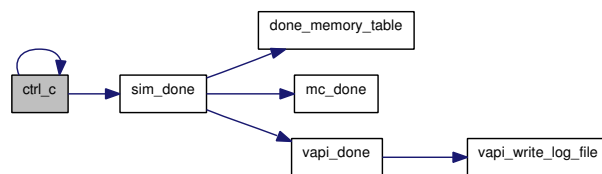
Signal handler for ctrl-C

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:



5.167.1.3 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

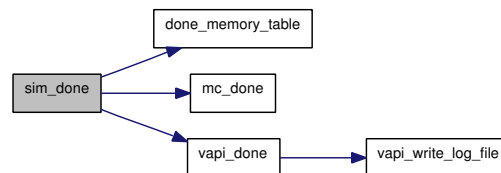
← *dat* The data structure to be passed as argument when the reset_hook function is called.

5.167.1.4 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:



5.167.1.5 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.

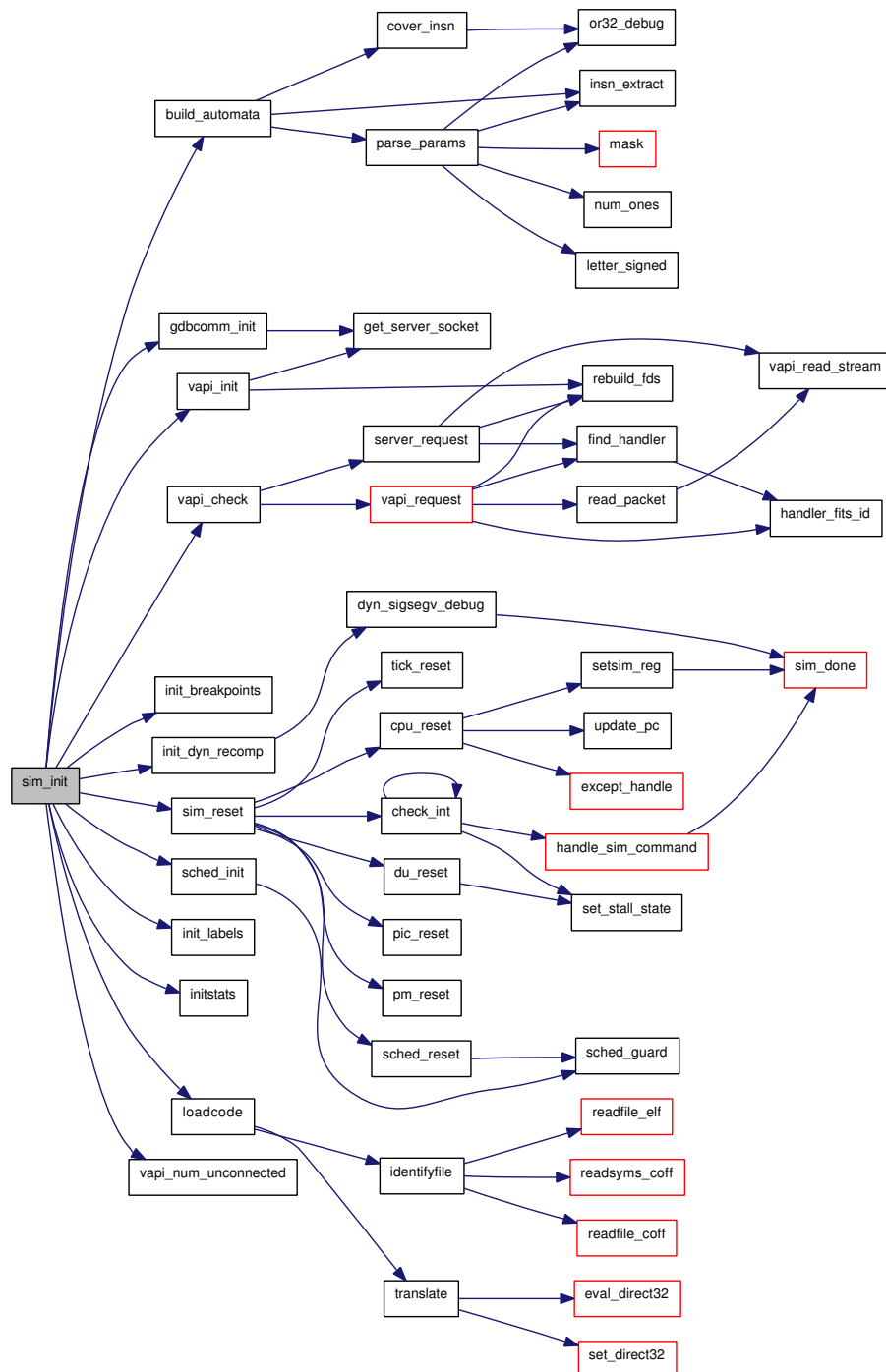
Open the various logs and statistics files requested by the configuration and/or command arguments.

Initialize GDB and VAPI connections.

Reset the simulator.

Wait for VAPI to connect if configured.

Here is the call graph for this function:



5.167.1.6 void sim_reset ()

Reset the simulator

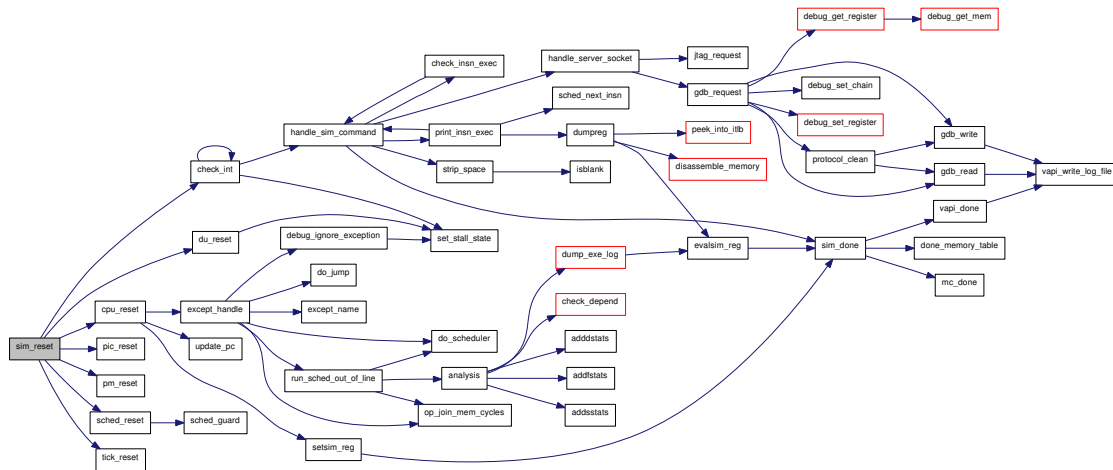
The scheduler is reset, then all reset functions on the reset hook list (i.e. peripherals) are reset. Then stan-

ard 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:



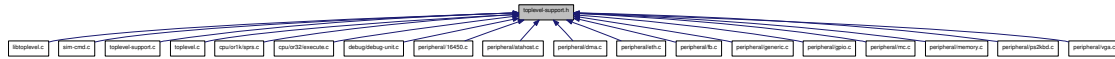
5.167.2 Variable Documentation

5.167.2.1 struct sim_reset_hook* sim_reset_hooks = NULL [static]

The list of reset hooks. Local to this source file

5.168 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](#) ()

5.168.1 Function Documentation

5.168.1.1 void [check_int](#) (void * *dat*)

Routine poll to see if interaction is needed

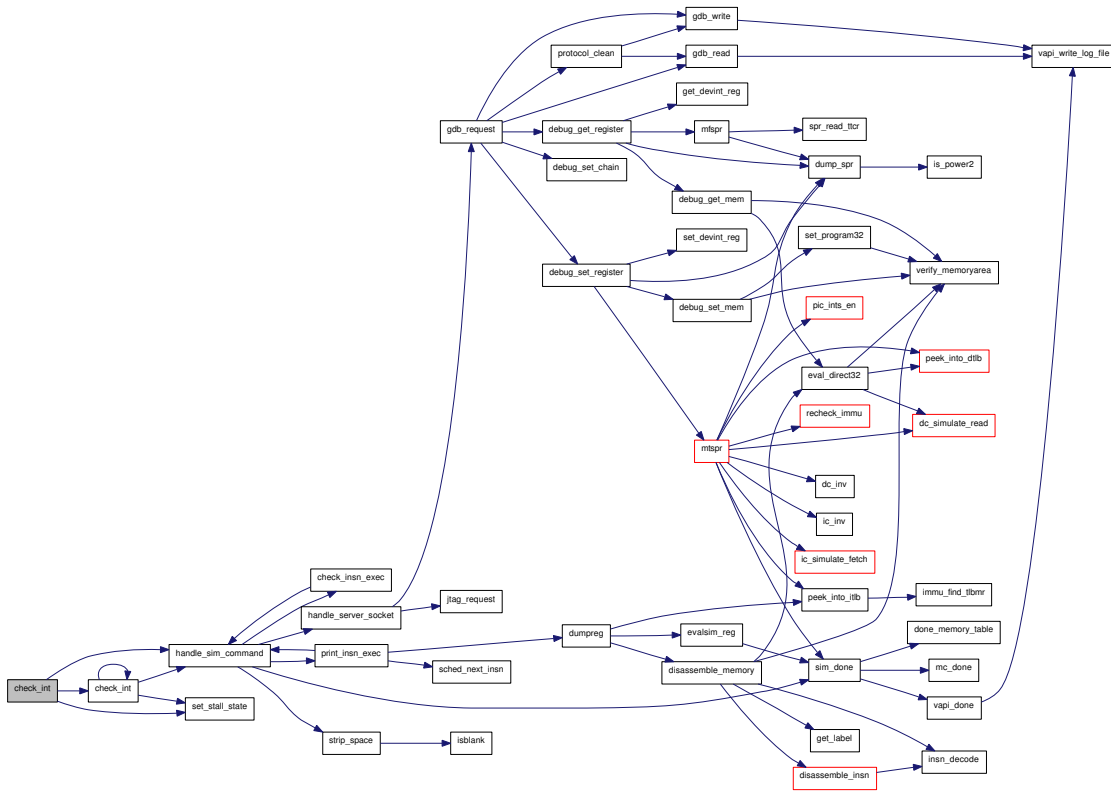
This is most likely to happen due to a ctrl-C. However when the -i flag is specified, the simulator starts up ready for interaction.

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:



5.168.1.2 void ctrl_c (int *signum*)

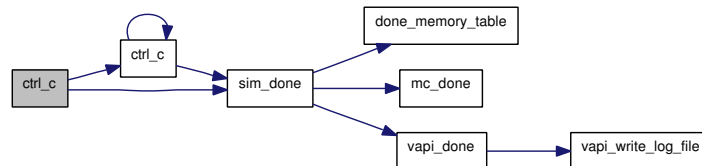
Signal handler for ctrl-C

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:



5.168.1.3 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

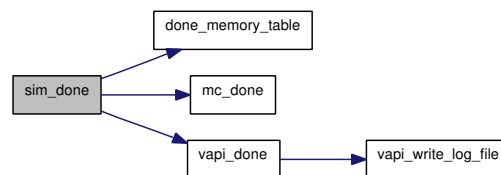
← *dat* The data structure to be passed as argument when the `reset_hook` function is called.

5.168.1.4 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:

**5.168.1.5 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.

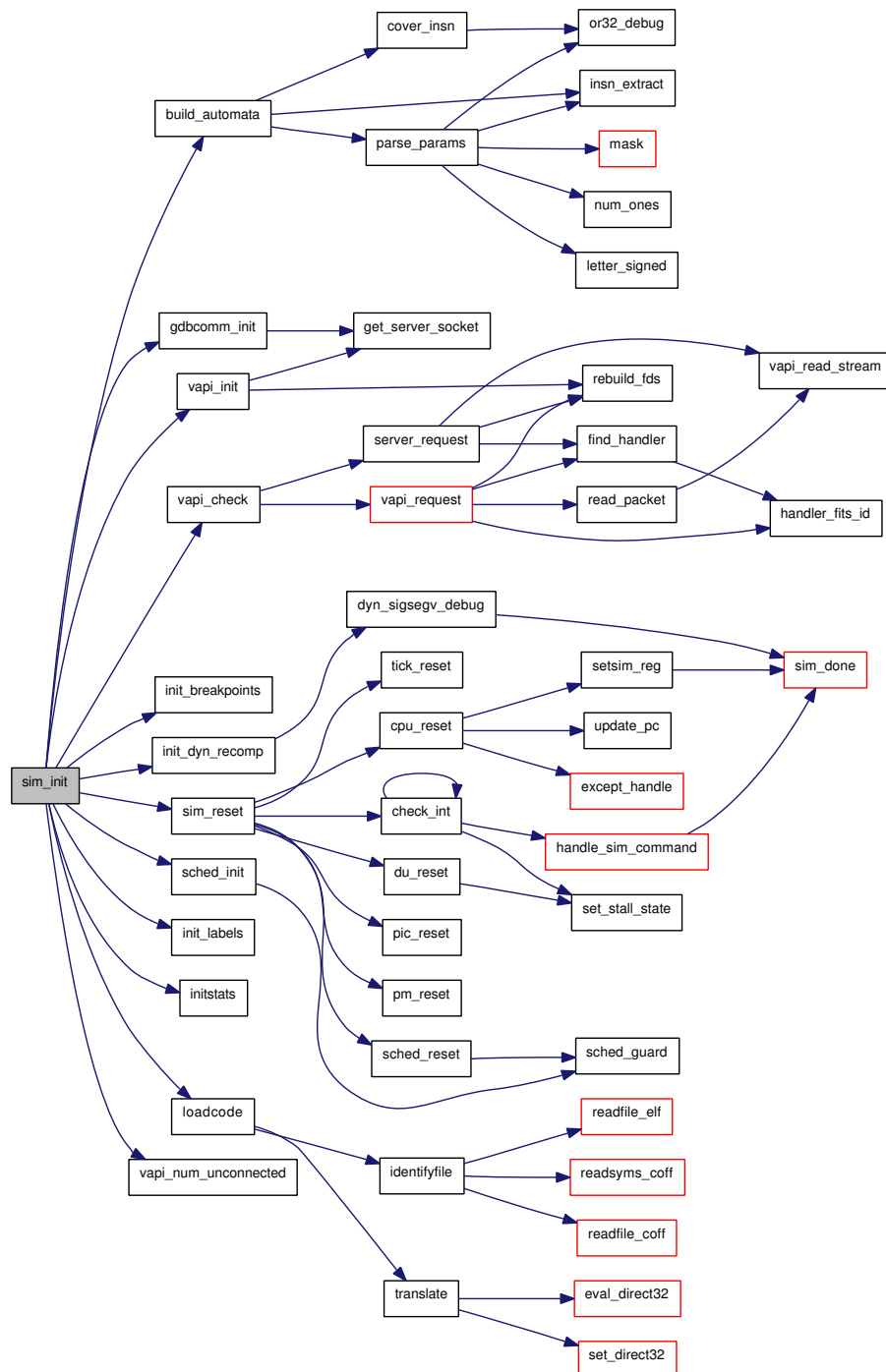
Open the various logs and statistics files requested by the configuration and/or command arguments.

Initialize GDB and VAPI connections.

Reset the simulator.

Wait for VAPI to connect if configured.

Here is the call graph for this function:



5.168.1.6 void sim_reset ()

Reset the simulator

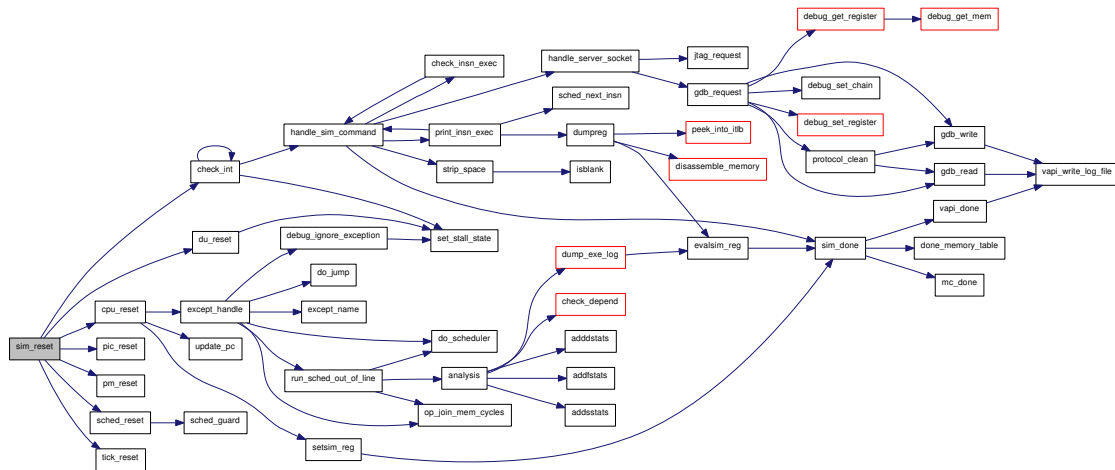
The scheduler is reset, then all reset functions on the reset hook list (i.e. peripherals) are reset. Then stan-

ard 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.

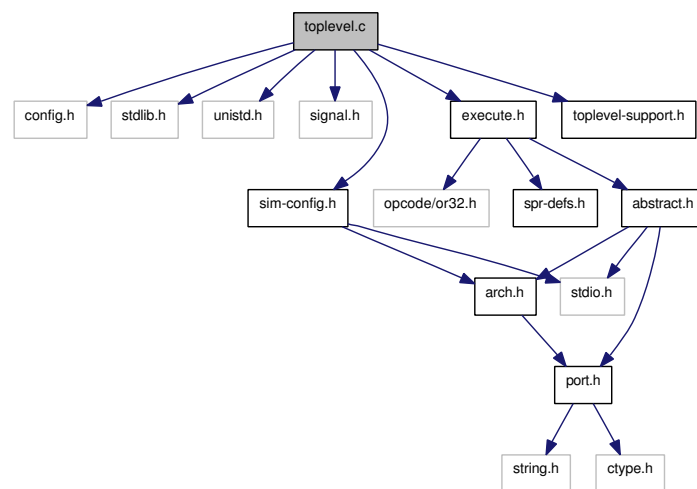
Here is the call graph for this function:



5.169 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[])`

5.169.1 Function Documentation

5.169.1.1 `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:

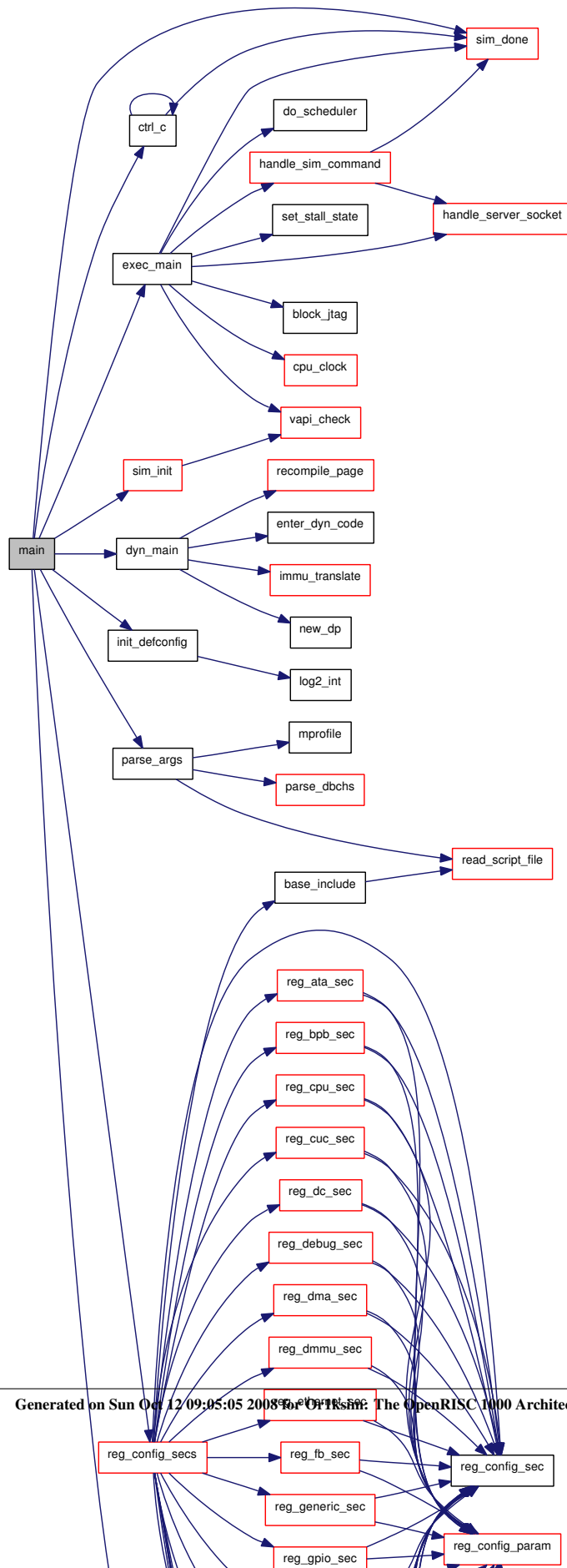
- ← *argc* The number of arguments to the command

← *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.

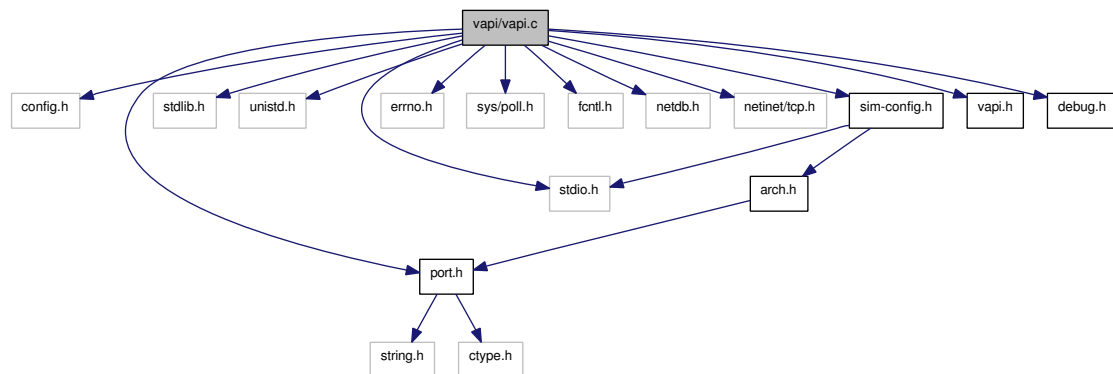
Here is the call graph for this function:



5.170 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 <fcntl.h>
#include <netdb.h>
#include <netinet/tcp.h>
#include "sim-config.h"
#include "vapi.h"
#include "debug.h"
```

Include dependency graph for vapi.c:



Data Structures

- struct [vapi_handler](#)

Functions

- [DEFAULT_DEBUG_CHANNEL](#) (vapi)
- 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

5.170.1 Function Documentation

5.170.1.1 static struct [vapi_handler](#)* [add_handler](#) (unsigned long *base_id*, unsigned long *num_ids*)
[static, read]

Here is the call graph for this function:



5.170.1.2 [DEFAULT_DEBUG_CHANNEL](#) ([vapi](#))

5.170.1.3 static struct [vapi_handler](#)* [find_handler](#) (unsigned long *id*) [static, read]

Here is the call graph for this function:



5.170.1.4 `int get_server_socket (const char * name, const char * proto, int port)`

5.170.1.5 `static int handler_fits_id (const struct vapi_handler * t, unsigned long id) [static]`

5.170.1.6 `static int read_packet (int fd, unsigned long * id, unsigned long * data) [static]`

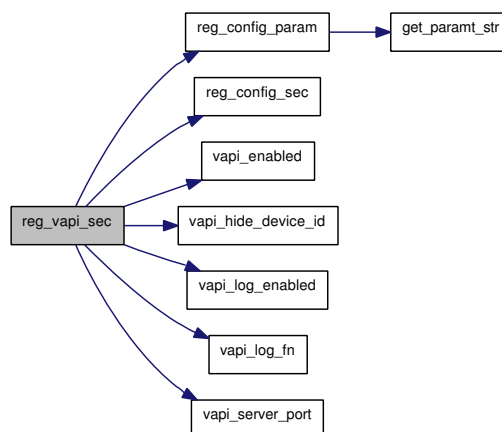
Here is the call graph for this function:



5.170.1.7 `void rebuild_fds ()`

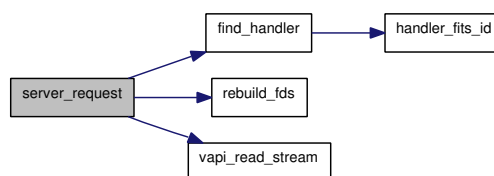
5.170.1.8 `void reg_vapi_sec (void)`

Here is the call graph for this function:



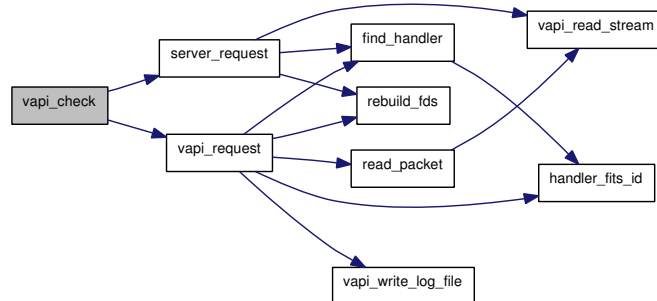
5.170.1.9 `static void server_request () [static]`

Here is the call graph for this function:



5.170.1.10 void vapi_check ()

Here is the call graph for this function:



5.170.1.11 void vapi_done ()

Here is the call graph for this function:

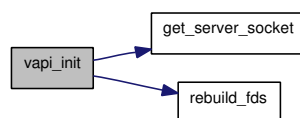


5.170.1.12 static void vapi_enabled (union param_val val, void * dat) [static]

5.170.1.13 static void vapi_hide_device_id (union param_val val, void * dat) [static]

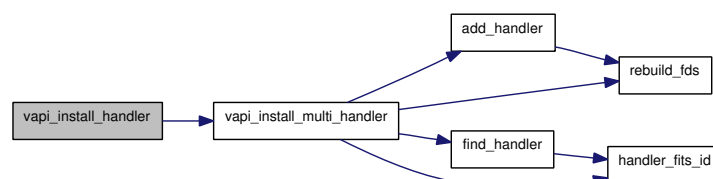
5.170.1.14 int vapi_init ()

Here is the call graph for this function:



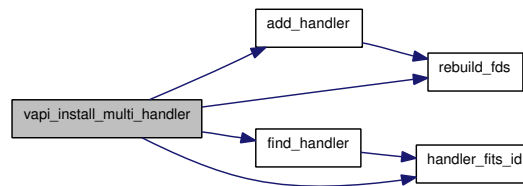
5.170.1.15 void vapi_install_handler (unsigned long id, void(*) (unsigned long, unsigned long, void *) read_func, void * dat)

Here is the call graph for this function:



5.170.1.16 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:



5.170.1.17 static void vapi_log_enabled (union param_val *val*, void * *dat*) [static]

5.170.1.18 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

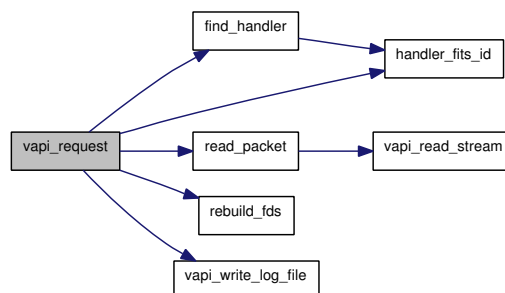
← *dat* The [config](#) data structure (not used here)

5.170.1.19 int vapi_num_unconnected (int *printout*)

5.170.1.20 static int vapi_read_stream (int *fd*, void * *buf*, int *len*) [static]

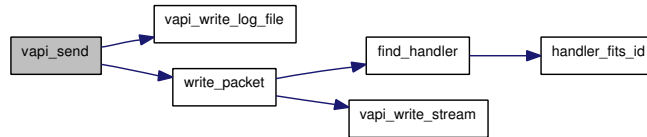
5.170.1.21 static void vapi_request (struct vapi_handler * *t*) [static]

Here is the call graph for this function:



5.170.1.22 void vapi_send (unsigned long *id*, unsigned long *data*)

Here is the call graph for this function:



5.170.1.23 static void vapi_server_port (union param_val *val*, void * *dat*) [static]

Set the VAPI server port

Ensure the value chosen is valid

Parameters:

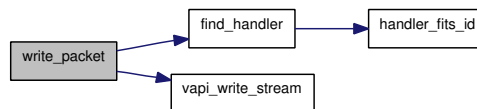
- ← *val* The value to use
- ← *dat* The [config](#) data structure (not used here)

5.170.1.24 void vapi_write_log_file (VAPI_COMMAND *command*, unsigned long *devid*, unsigned long *data*)

5.170.1.25 static int vapi_write_stream (int *fd*, void * *buf*, int *len*) [static]

5.170.1.26 static int write_packet (unsigned long *id*, unsigned long *data*) [static]

Here is the call graph for this function:



5.170.2 Variable Documentation

5.170.2.1 struct pollfd* fds = NULL [static]

5.170.2.2 int nfds = 0 [static]

5.170.2.3 unsigned int nhandlers = 0 [static]

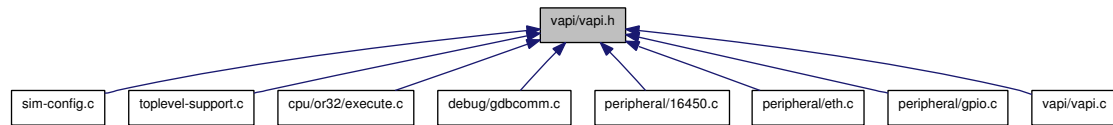
5.170.2.4 unsigned int server_fd = 0 [static]

5.170.2.5 unsigned int serverIP = 0 [static]

5.170.2.6 int tcp_level = 0 [static]

5.171 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` ()

5.171.1 Define Documentation

5.171.1.1 #define VAPI_MAX_DEVID 0xFFFF

Maximum value for VAPI device id

5.171.2 Enumeration Type Documentation

5.171.2.1 enum VAPI_COMMAND

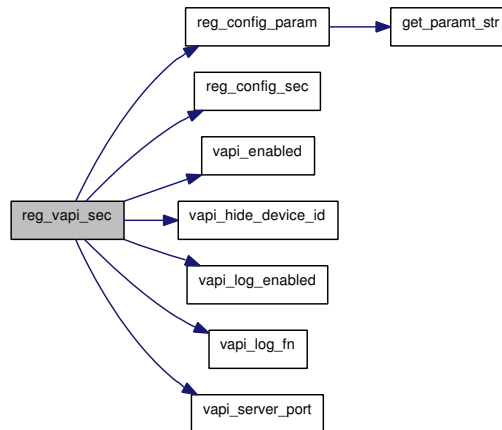
Enumerator:

`VAPI_COMMAND_REQUEST`
`VAPI_COMMAND_SEND`
`VAPI_COMMAND_END`

5.171.3 Function Documentation

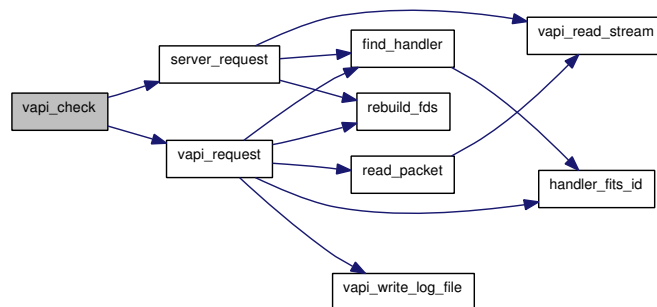
5.171.3.1 void reg_vapi_sec ()

Here is the call graph for this function:



5.171.3.2 void vapi_check ()

Here is the call graph for this function:



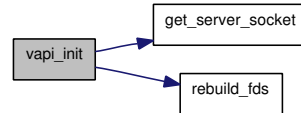
5.171.3.3 void vapi_done ()

Here is the call graph for this function:



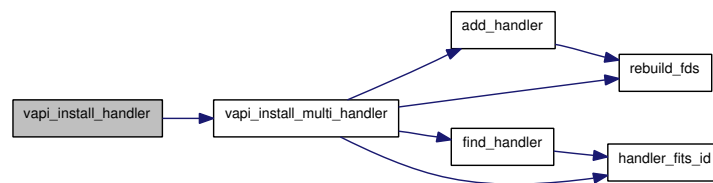
5.171.3.4 int vapi_init ()

Here is the call graph for this function:



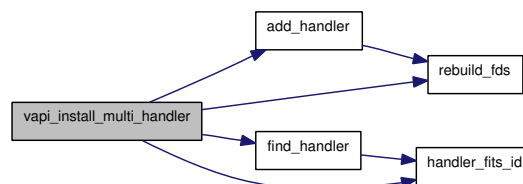
5.171.3.5 void vapi_install_handler (unsigned long id, void(*) (unsigned long, unsigned long, void *) read_func, void * dat)

Here is the call graph for this function:



5.171.3.6 void vapi_install_multi_handler (unsigned long base_id, unsigned long num_ids, void(*) (unsigned long, unsigned long, void *) read_func, void * dat)

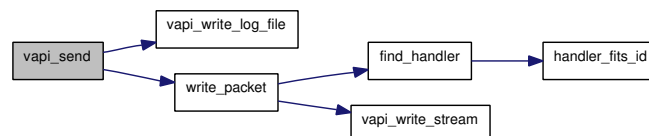
Here is the call graph for this function:



5.171.3.7 int vapi_num_unconnected (int printout)

5.171.3.8 void vapi_send (unsigned long id, unsigned long data)

Here is the call graph for this function:



5.171.3.9 void `vapi_write_log_file` (`VAPI_COMMAND` *command*, unsigned long *device_id*, unsigned long *data*)

Index

- [__ORSIM_DBCL_ERR](#)
debug.h, [793](#)
- [__ORSIM_DBCL_FIXME](#)
debug.h, [793](#)
- [__ORSIM_DBCL_TRACE](#)
debug.h, [793](#)
- [__ORSIM_DBCL_WARN](#)
debug.h, [793](#)
- [_DYNAMIC](#)
elf.h, [253](#)
- [__ORSIM_DBG_USE_FUNC](#)
debug.h, [792](#)
- [__ORSIM_DEBUG_CLASS](#)
debug.h, [793](#)
- [__ORSIM_DEBUG_LOG](#)
debug.h, [792](#)
- [__ORSIM_DPRINTF](#)
debug.h, [792](#)
- [__ORSIM_GET_DEBUGGING](#)
debug.h, [792](#)
- [__ORSIM_GET_DEBUGGING_ERR](#)
debug.h, [792](#)
- [__ORSIM_GET_DEBUGGING_FIXME](#)
debug.h, [792](#)
- [__ORSIM_GET_DEBUGGING_TRACE](#)
debug.h, [792](#)
- [__ORSIM_GET_DEBUGGING_WARN](#)
debug.h, [792](#)
- [__ORSIM_NO_DEC_DBCH](#)
debug.c, [790](#)
- [__op_param1](#)
dyn_rec.c, [346](#)
op.c, [401](#)
- [__op_param2](#)
dyn_rec.c, [346](#)
op.c, [401](#)
- [__op_param3](#)
dyn_rec.c, [346](#)
op.c, [401](#)
- [__or_dynop](#)
op.c, [394](#)
- [__orsim_dbchs](#)
debug.c, [790](#)
- [_csm_list](#), [11](#)
cmatch, [12](#)
- [cmovs](#), [12](#)
- [cnt](#), [12](#)
- [dead](#), [12](#)
- [from](#), [12](#)
- [next](#), [12](#)
- [ninsn](#), [12](#)
- [osize](#), [12](#)
- [ref](#), [12](#)
- [size](#), [12](#)
- [_cuc_func](#), [13](#)
bb, [14](#)
end_addr, [14](#)
fdeps, [14](#)
init_bb_reloc, [14](#)
lur, [14](#)
memory_order, [14](#)
msched, [14](#)
mtype, [14](#)
nfdeps, [14](#)
nmsched, [14](#)
num_bb, [14](#)
num_init_bb, [14](#)
num_runs, [14](#)
orig_time, [14](#)
saved_regs, [14](#)
start_addr, [14](#)
timings, [14](#)
tmp, [14](#)
used_regs, [14](#)
- [_dep_list_t](#), [15](#)
next, [15](#)
ref, [15](#)
- 16450.c
char_clks, [573](#)
DEFAULT_DEBUG_CHANNEL, [573](#)
MAX_SKEW, [567](#)
MIN, [567](#)
reg_uart_sec, [573](#)
send_char, [574](#)
uart_16550, [574](#)
uart_add_char, [574](#)
UART_ADDR_SPACE, [567](#)
uart_baseaddr, [574](#)
UART_BREAK_COUNT, [567](#)
uart_channel, [574](#)

- uart_char_clock, 574
- UART_CHAR_TIMEOUT, 567
- uart_check_char, 575
- uart_check_rdi, 575
- uart_check_rlsi, 576
- uart_clear_int, 576
- UART_CLOCK_DIVIDER, 567
- UART_DLH, 567
- UART_DLL, 567
- uart_enabled, 576
- UART_FCR, 567
- UART_FCR_FIE, 567
- UART_FCR_RRXFI, 567
- UART_FCR_RTXFI, 567
- UART_FGETC_SLOWDOWN, 567
- UART_FIFO_TRIGGER, 567
- UART_IER, 568
- UART_IER_MSI, 570
- UART_IER_RDI, 570
- UART_IER_RLSI, 570
- UART_IER_THRI, 570
- UART_IIR, 570
- UART_IIR_CTI, 570
- UART_IIR_ID, 570
- UART_IIR_MSI, 570
- UART_IIR_NO_INT, 570
- UART_IIR_RDI, 570
- UART_IIR_RLSI, 570
- UART_IIR_THRI, 570
- uart_int_cti, 577
- uart_int_msi, 577
- uart_int_rdi, 577
- uart_int_rlsi, 578
- uart_int_thri, 578
- uart_irq, 579
- uart_jitter, 579
- UART_LCR, 570
- UART_LCR_DLAB, 570
- UART_LCR_EPAR, 570
- UART_LCR_PARITY, 570
- UART_LCR_RESET, 570
- UART_LCR_SBC, 570
- UART_LCR_SPAR, 570
- UART_LCR_STOP, 570
- UART_LCR_WLEN5, 570
- UART_LCR_WLEN6, 570
- UART_LCR_WLEN7, 570
- UART_LCR_WLEN8, 570
- uart_loopback, 579
- UART_LSR, 570
- UART_LSR_BREAK, 570
- UART_LSR_FRAME, 570
- UART_LSR_OVRRUN, 570
- UART_LSR_PARITY, 570
- UART_LSR_RDRDY, 570
- UART_LSR_RXERR, 570
- UART_LSR_TXBUFE, 570
- UART_LSR_TXSERE, 570
- UART_MAX_FIFO_LEN, 570
- UART_MCR, 570
- UART_MCR_AUX1, 571
- UART_MCR_AUX2, 571
- UART_MCR_DTR, 571
- UART_MCR_LOOP, 571
- UART_MCR_RTS, 571
- UART_MSR, 571
- UART_MSR_CTS, 571
- UART_MSR_DCD, 571
- UART_MSR_DCTS, 571
- UART_MSR_DDCD, 571
- UART_MSR_DDSR, 571
- UART_MSR_DSR, 571
- UART_MSR_RI, 571
- UART_MSR_TERI, 571
- uart_newway, 579
- uart_next_int, 580
- uart_read_byte, 580
- uart_recv_break, 580
- uart_recv_break_start, 581
- uart_recv_break_stop, 581
- uart_recv_char, 581
- uart_reset, 582
- UART_RXBUF, 571
- uart_sched_recv_check, 582
- UART_SCR, 571
- uart_sec_end, 582
- uart_sec_start, 583
- uart_send_break, 583
- uart_status, 584
- uart_tx_send, 584
- UART_TXBUF, 571
- UART_VALID_FCR, 571
- UART_VALID_IER, 571
- UART_VALID_IIR, 571
- UART_VALID_LCR, 571
- UART_VALID_LSR, 571
- UART_VALID_MCR, 571
- UART_VALID_MSR, 571
- UART_VAPI_BUF_LEN, 571
- uart_vapi_cmd, 584
- uart_vapi_id, 585
- uart_vapi_read, 585
- uart_write_byte, 585
- 16450.h
 - reg_uart_sec, 588
 - uart_reset, 589
 - uart_status, 589

- a0
 - dma_channel, 78
- a1
 - dma_channel, 78
- abstract.c
 - adjust_rw_delay, 207
 - bit_mask, 207
 - cur_area, 221
 - cur_vadd, 221
 - data_ci, 221
 - dev_list, 221
 - disassemble_memory, 207
 - done_memory_table, 207
 - dump_memory, 207
 - eval_direct16, 208
 - eval_direct32, 208
 - eval_direct8, 208
 - eval_insn, 209
 - eval_mem16, 209
 - eval_mem32, 209
 - eval_mem8, 210
 - eval_mem_16_inv, 210
 - eval_mem_16_inv_direct, 211
 - eval_mem_32_inv, 211
 - eval_mem_32_inv_direct, 211
 - eval_mem_8_inv, 212
 - eval_mem_8_inv_direct, 212
 - evalsim_mem16, 212
 - evalsim_mem32, 212
 - evalsim_mem8, 213
 - generate_time_pretty, 213
 - insn_ci, 221
 - mc_area, 221
 - memory_table_status, 214
 - reg_mem_area, 214
 - register_memoryarea_mask, 214
 - set_direct16, 214
 - set_direct32, 215
 - set_direct8, 215
 - set_mem16, 216
 - set_mem32, 216
 - set_mem8, 217
 - set_mem_16_inv, 217
 - set_mem_16_inv_direct, 218
 - set_mem_32_inv, 218
 - set_mem_32_inv_direct, 218
 - set_mem_8_inv, 219
 - set_mem_8_inv_direct, 219
 - set_mem_valid, 219
 - set_program32, 219
 - set_program8, 219
 - setsim_mem16, 219
 - setsim_mem32, 220
 - setsim_mem8, 220
 - verify_memoryarea, 221
- abstract.h
 - adjust_rw_delay, 224
 - CT_NONE, 224
 - CT_PHYSICAL, 224
 - CT_VIRTUAL, 224
 - cur_area, 236
 - data_ci, 236
 - DEFAULT_MEMORY_LEN, 224
 - DEFAULT_MEMORY_START, 224
 - disassemble_memory, 224
 - done_memory_table, 225
 - dump_memory, 225
 - eval_direct16, 225
 - eval_direct32, 226
 - eval_direct8, 226
 - eval_insn, 227
 - eval_mem16, 227
 - eval_mem32, 228
 - eval_mem8, 228
 - evalsim_mem16, 228
 - evalsim_mem32, 229
 - evalsim_mem8, 229
 - generate_time_pretty, 230
 - hist_exec_tail, 236
 - HISTEXEC_LEN, 224
 - insn_ci, 236
 - INSNAME_LEN, 224
 - LABELNAME_LEN, 224
 - LE16, 224
 - LONGEST, 224
 - MAX_OPERANDS, 224
 - memory_table_status, 230
 - OP_MEM_ACCESS, 224
 - OPERANDNAME_LEN, 224
 - reg_mem_area, 230
 - set_direct16, 231
 - set_direct32, 231
 - set_direct8, 232
 - set_mem16, 232
 - set_mem32, 233
 - set_mem8, 233
 - set_mem_valid, 234
 - set_program32, 234
 - set_program8, 234
 - setsim_mem16, 234
 - setsim_mem32, 235
 - setsim_mem8, 235
 - STACK_SIZE, 224
 - ULONGEST, 224
 - verify_memoryarea, 236
- add_breakpoint
 - labels.c, 262
 - labels.h, 264

- add_crc
 - eth_device, 110
- add_data_dep
 - cuc.h, 465
 - insn.c, 479
- add_dep
 - cuc.h, 466
 - insn.c, 479
- add_handler
 - vapi.c, 840
- add_label
 - labels.c, 262
 - labels.h, 264
- add_latches
 - cuc.h, 466
 - insn.c, 479
- add_memory_dep
 - cuc.h, 466
 - cuc/memory.c, 496
- add_to_op_params
 - dyn_rec.c, 330
 - dyn_rec.h, 348
- add_to_opq
 - dyn_rec.c, 330
 - dyn_rec.h, 348
- addstats
 - stats.c, 275
 - stats.h, 278
- addend
 - reloc, 157
- addfstats
 - stats.c, 275
 - stats.h, 278
- addprogram
 - parse.c, 266
- addr
 - bpb_entry, 28
 - breakpoint_entry, 31
 - btic_entry, 32
 - fb_state, 115
 - func_struct, 119
 - hist_exec, 122
 - label_entry, 142
 - memory_hash, 150
 - mprofentry_struct, 151
 - stack_struct, 170
- ADDR_C
 - arch.h, 283
- addr_compare
 - dev_memarea, 76
- addr_mask
 - dev_memarea, 76
- addr_val
 - param_val, 155
- address
 - jtr_read_block_message, 133
 - jtr_read_message, 135
 - jtr_write_block_message, 137
 - jtr_write_message, 139
- addsstats
 - stats.c, 275
 - stats.h, 278
- adjust_rw_delay
 - abstract.c, 207
 - abstract.h, 224
- adv.c
 - calc_max, 437
 - detect_max_values, 437
 - insert_conditional_facts, 437
 - mark_successors, 437
 - mask, 437
 - max_op, 437
- am0
 - dma_channel, 78
- am1
 - dma_channel, 78
- analyse_function
 - cuc.c, 447
- analyse_timings
 - cuc.h, 466
 - timings.c, 504
- analysis
 - execute.c, 361
 - execute.h, 255
- apply_edge_condition
 - insn.c, 479
- arch.h
 - ADDR_C, 283
 - LINK_REGNO, 283
 - oraddr_t, 284
 - orreg_t, 284
 - PRIdREG, 283
 - PRIxADDR, 283
 - PRIxREG, 284
 - REG_C, 284
 - uorreg_t, 284
- archf, 16
 - gen_func_reloc, 16
 - gen_reloc, 16
 - get_real_func_len, 16
- archfs
 - dyngen.h, 355
 - dyngen_i386.c, 358
- argv
 - xterm_channel, 176
- asm
 - op.c, 394
 - op_i386.h, 412

- ASSIGN_FLAG
 - fields.h, 691
- AT_BASE
 - elf.h, 250
- AT_EGID
 - elf.h, 250
- AT_ENTRY
 - elf.h, 250
- AT_EUID
 - elf.h, 250
- AT_EXECPD
 - elf.h, 250
- AT_FLAGS
 - elf.h, 250
- AT_GID
 - elf.h, 250
- AT_IGNORE
 - elf.h, 250
- AT_NOTELF
 - elf.h, 250
- AT_NULL
 - elf.h, 250
- AT_PAGESZ
 - elf.h, 250
- AT_PHDR
 - elf.h, 250
- AT_PHEMT
 - elf.h, 250
- AT_PHNUM
 - elf.h, 250
- AT_UID
 - elf.h, 250
- ATA_ADDR_SPACE
 - atahost.h, 626
- ATA_ASR
 - atadevice.h, 601
- ata_baseaddr
 - atahost.c, 615
- ATA_BELECO
 - atahost.h, 626
- ATA_BELEC1
 - atahost.h, 626
- ata_calc_lba
 - atadevice_cmdi.c, 604
- ATA_CHR
 - atadevice.h, 601
- ATA_CLR
 - atadevice.h, 601
- ata_cmd_complete
 - atadevice_cmdi.c, 604
- ATA_CR
 - atadevice.h, 601
- ATA_CTRL
 - atahost.h, 626
- ATA_DA
 - atadevice.h, 601
- ATA_DAR_DS0
 - atadevice.h, 601
- ATA_DAR_DS1
 - atadevice.h, 601
- ATA_DAR_H
 - atadevice.h, 601
- ATA_DAR_WTG
 - atadevice.h, 601
- ATA_DCR
 - atadevice.h, 601
- ATA_DCR_IEN
 - atadevice.h, 601
- ATA_DCR_RST
 - atadevice.h, 601
- ata_dev_id
 - atahost.c, 615
- ata_device, 17
 - command, 20
 - conf, 20
 - cylinder_high, 20
 - cylinder_low, 20
 - daspi, 20
 - daspo, 20
 - dataport_i, 20
 - dbuf, 20
 - dbuf_cnt, 20
 - dbuf_ptr, 20
 - dev, 20
 - device_control, 20
 - device_head, 20
 - dma_mode, 20
 - dmarq, 20
 - end_t_func, 20
 - error, 20
 - features, 20
 - file, 20
 - firmware, 20
 - heads, 20
 - heads_per_cylinder, 20
 - host, 20
 - internals, 20
 - intrq, 20
 - iordy, 20
 - lba, 20
 - mwdma, 20
 - nr_sect, 20
 - packet, 20
 - pdiagi, 20
 - pdiago, 20
 - pio, 20
 - pio_mode, 20
 - regs, 20

- sector_count, 20
- sector_number, 20
- sectors, 20
- sectors_per_track, 20
- sigs, 20
- size, 20
- size_sect, 20
- state, 20
- status, 20
- stream, 20
- type, 20
- ata_device_do_command_register
 - atadevice.c, 595
- ata_device_do_control_register
 - atadevice.c, 595
- ata_device_execute_cmd
 - atadevice_cmdi.c, 604
 - atadevice_cmdi.h, 611
- ata_device_hw_reset
 - atadevice.c, 595
- ata_device_init
 - atadevice.c, 595
- ata_device_reset_cmd
 - atadevice_cmdi.c, 604
- ata_device_write
 - atadevice.c, 595
- ata_devices, 22
 - device, 22
- ata_devices_hw_reset
 - atadevice.c, 596
 - atadevice.h, 601
- ata_devices_init
 - atadevice.c, 596
 - atadevice.h, 602
- ata_devices_read
 - atadevice.c, 596
 - atadevice.h, 602
- ata_devices_write
 - atadevice.c, 596
 - atadevice.h, 602
- ATA_DEVID
 - atahost.h, 626
- ATA_DHR
 - atadevice.h, 601
- ATA_DHR_DEV
 - atadevice.h, 601
- ATA_DHR_H
 - atadevice.h, 601
- ATA_DHR_LBA
 - atadevice.h, 601
- ata_dma_delay
 - atahost.h, 626
- ATA_DMA_EN
 - atahost.h, 626
- ata_dma_mode0_td
 - atahost.c, 615
- ata_dma_mode0_teoc
 - atahost.c, 615
- ata_dma_mode0_tm
 - atahost.c, 615
- ATA_DMA_RD
 - atahost.h, 626
- ATA_DMA_TIP
 - atahost.h, 626
- ATA_DMA_WR
 - atahost.h, 626
- ATA_DMARQ
 - atahost.h, 626
- ATA_DR
 - atadevice.h, 601
- ATA_DRBE
 - atahost.h, 626
- ATA_DTBF
 - atahost.h, 626
- ATA_DTR0
 - atahost.h, 626
- ATA_DTR1
 - atahost.h, 626
- ata_enabled
 - atahost.c, 615
- ata_enddevice
 - atahost.c, 615
- ATA_ERR
 - atadevice.h, 601
- ATA_ERR_ABT
 - atadevice.h, 601
- ATA_ERR_AMN
 - atadevice.h, 601
- ATA_ERR_BBK
 - atadevice.h, 601
- ATA_ERR_IDNF
 - atadevice.h, 601
- ATA_ERR_TON
 - atadevice.h, 601
- ATA_ERR_UNC
 - atadevice.h, 601
- ata_execute_device_diagnostics_cmd
 - atadevice_cmdi.c, 604
 - atadevice_cmdi.h, 612
- ata_file
 - atahost.c, 615
- ata_firmware
 - atahost.c, 615
- ATA_FR
 - atadevice.h, 601
- ATA_FTE0
 - atahost.h, 626
- ATA_FTE1

- atahost.h, 626
- ata_heads
 - atahost.c, 616
- ata_host, 23
 - baseaddr, 24
 - ctrl, 24
 - dev_id, 24
 - dev_sel, 24
 - devices, 24
 - dma_mode0_td, 24
 - dma_mode0_teoc, 24
 - dma_mode0_tm, 24
 - dtr0, 24
 - dtr1, 24
 - enabled, 24
 - irq, 24
 - mem, 24
 - pctr, 24
 - pftr0, 24
 - pftr1, 24
 - pio_mode0_t1, 24
 - pio_mode0_t2, 24
 - pio_mode0_t4, 24
 - pio_mode0_teoc, 24
 - regs, 24
 - rev, 24
 - rxb, 24
 - stat, 24
 - txb, 24
- ATA_IDE_EN
 - atahost.h, 626
- ATA_IDEIS
 - atahost.h, 626
- ata_identify_device_cmd
 - atadevice_cmdi.c, 605
- ata_initialize_device_parameters_cmd
 - atadevice_cmdi.c, 605
- ata_int
 - atahost.c, 616
 - atahost.h, 626
- ATA_IORDY
 - atahost.h, 626
- ATA_IORDY_FTE0
 - atahost.h, 626
- ATA_IORDY_FTE1
 - atahost.h, 626
- ata_irq
 - atahost.c, 616
- ata_mwdma
 - atahost.c, 616
- ata_packet
 - atahost.c, 616
- ATA_PCTR
 - atahost.h, 626
- ATA_PFFTR0
 - atahost.h, 626
- ATA_PFFTR1
 - atahost.h, 626
- ata_pio
 - atahost.c, 616
- ata_pio_delay
 - atahost.h, 626
- ata_pio_mode0_t1
 - atahost.c, 616
- ata_pio_mode0_t2
 - atahost.c, 617
- ata_pio_mode0_t4
 - atahost.c, 617
- ata_pio_mode0_teoc
 - atahost.c, 617
- ATA_PIO_TIP
 - atahost.h, 626
- ata_pretty_status
 - atadevice.c, 597
- ATA_PWPP
 - atahost.h, 626
- ATA_PWPPF
 - atahost.h, 626
- ata_read32
 - atahost.c, 617
- ata_read_native_max_addr
 - atadevice_cmdi.c, 605
- ata_read_sect
 - atadevice_cmdi.c, 605
- ata_read_sectors_cmd
 - atadevice_cmdi.c, 605
- ata_reset
 - atahost.c, 617
- ata_rev
 - atahost.c, 617
- ATA_REVNO
 - atahost.h, 626
- ATA_RST
 - atahost.h, 626
- ATA_RXB
 - atahost.h, 626
- ATA_SCR
 - atadevice.h, 601
- ata_sec_end
 - atahost.c, 617
- ata_sec_start
 - atahost.c, 618
- ata_sectors
 - atahost.c, 618
- ata_set_device_signature
 - atadevice_cmdi.c, 606
- ata_set_features
 - atadevice_cmdi.c, 606

- ata_set_sect
 - atadevice_cmdi.c, 606
- ata_size
 - atahost.c, 619
- ATA_SNR
 - atadevice.h, 601
- ATA_SR
 - atadevice.h, 601
- ATA_SR_BSY
 - atadevice.h, 601
- ATA_SR_COR
 - atadevice.h, 601
- ATA_SR_DF
 - atadevice.h, 601
- ATA_SR_DRDY
 - atadevice.h, 601
- ATA_SR_DRQ
 - atadevice.h, 601
- ATA_SR_DSC
 - atadevice.h, 601
- ATA_SR_ERR
 - atadevice.h, 601
- ATA_SR_IDX
 - atadevice.h, 601
- ata_start_device
 - atahost.c, 619
- ATA_STAT
 - atahost.h, 626
- ATA_STATE_HW_RST
 - atadevice.h, 601
- ATA_STATE_IDLE
 - atadevice.h, 601
- ATA_STATE_SW_RST
 - atadevice.h, 601
- ata_status
 - atahost.c, 619
- ATA_T1
 - atahost.h, 626
- ATA_T2
 - atahost.h, 626
- ATA_T4
 - atahost.h, 626
- ATA_TD
 - atahost.h, 626
- ATA_TEOC
 - atahost.h, 626
- ATA_TM
 - atahost.h, 626
- ATA_TXB
 - atahost.h, 626
- ata_type
 - atahost.c, 619
- ata_write32
 - atahost.c, 619
- ata_write_sect
 - atadevice_cmdi.c, 606
- ata_write_sectors
 - atadevice_cmdi.c, 606
- atacmd.h
 - CFA_DISABLE_8BIT_PIO_TRANSFER_ -
MODE, 593
 - CFA_DISABLE_POWER_MODE1, 593
 - CFA_ENABLE_8BIT_PIO_TRANSFER_ -
MODE, 593
 - CFA_ENABLE_POWER_MODE1, 593
 - CFA_ERASE_SECTORS, 593
 - CFA_REQUEST_EXTENDED_ERROR_ -
CODE, 593
 - CFA_TRANSLATE_SECTOR, 593
 - CFA_WRITE_MULTIPLE_WITHOUT_ -
ERASE, 593
 - CFA_WRITE_SECTORS_WITHOUT_ -
ERASE, 593
 - CHECK_POWER_MODE, 593
 - DEVICE_RESET, 593
 - DISABLE_ADVANCED_POWER_ -
MANAGEMENT, 593
 - DISABLE_MEDIA_STATUS_ -
NOTIFICATION, 593
 - DISABLE_POWERUP_IN_STANDBY_ -
FEATURE_SET, 593
 - DISABLE_READ_LOOKAHEAD, 593
 - DISABLE_RELEASE_INTERRUPT, 593
 - DISABLE_REVERTING_TO_POWERON_ -
DEFAULTS, 593
 - DISABLE_SERVICE_INTERRUPT, 593
 - DISABLE_WRITE_CACHE, 593
 - DOWNLOAD_MICROCODE, 593
 - ENABLE_ADVANCED_POWER_ -
MANAGEMENT, 593
 - ENABLE_MEDIA_STATUS_ -
NOTIFICATION, 593
 - ENABLE_POWERUP_IN_STANDBY_ -
FEATURE_SET, 593
 - ENABLE_READ_LOOKAHEAD_ -
FEATURE, 593
 - ENABLE_RELEASE_INTERRUPT, 593
 - ENABLE_REVERTING_TO_POWERON_ -
DEFAULTS, 593
 - ENABLE_SERVICE_INTERRUPT, 593
 - ENABLE_WRITE_CACHE, 593
 - EXECUTE_DEVICE_DIAGNOSTICS, 593
 - FLUSH_CACHE, 593
 - GET_MEDIA_STATUS, 593
 - IDENTIFY_DEVICE, 593
 - IDENTIFY_PACKET_DEVICE, 593
 - IDLE, 593
 - IDLE_IMMEDIATE, 593

- INITIALIZE_DEVICE_PARAMETERS, 593
 - MEDIA_EJECT, 593
 - MEDIA_LOCK, 593
 - MEDIA_UNLOCK, 593
 - NOP, 593
 - PACKET, 593
 - POWERUP_IN_STANDBY_FEATURE_-
SET_SPINUP, 593
 - READ_BUFFER, 593
 - READ_DMA, 593
 - READ_DMA_QUEUED, 593
 - READ_MULTIPLE, 593
 - READ_NATIVE_MAX_ADDRESS, 593
 - READ_SECTOR, 593
 - READ_SECTORS, 593
 - READ_VERIFY_SECTOR, 593
 - READ_VERIFY_SECTORS, 593
 - SECURITY_DISABLE_PASSWORD, 593
 - SECURITY_ERASE_PREPARE, 593
 - SECURITY_ERASE_UNIT, 593
 - SECURITY_FREEZE_LOCK, 593
 - SECURITY_SET_PASSWORD, 593
 - SECURITY_UNLOCK, 593
 - SEEK, 593
 - SERVICE, 593
 - SET_FEATURES, 593
 - SET_MAX, 593
 - SET_MAX_ADDRESS, 593
 - SET_MAX_FREEZE_LOCK, 593
 - SET_MAX_LOCK, 593
 - SET_MAX_SET_PASSWORD, 593
 - SET_MAX_UNLOCK, 593
 - SET_MULTIPLE_MODE, 593
 - SET_TRANSFER_MODE_SECTOR_-
COUNT_REG, 593
 - SLEEP, 593
 - SMART, 593
 - SMART_ATTRIBUTE_AUTOSAVE, 593
 - SMART_DISABLE_OPERATIONS, 593
 - SMART_ENABLE_OPERATIONS, 593
 - SMART_EXECUTE_OFFLINE_-
IMMEDIATE, 593
 - SMART_READ_DATA, 593
 - SMART_READ_LOG, 593
 - SMART_RETURN_STATUS, 593
 - SMART_SAVE_ATTRIBUTE_VALUES, 593
 - SMART_WRITE_LOG, 593
 - STANDBY, 593
 - STANDBY_IMMEDIATE, 593
 - WRITE_BUFFER, 593
 - WRITE_DMA, 593
 - WRITE_DMA_QUEUED, 593
 - WRITE_MULTIPLE, 593
 - WRITE_SECTOR, 593
 - WRITE_SECTORS, 593
- atadevice.c
 - ata_device_do_command_register, 595
 - ata_device_do_control_register, 595
 - ata_device_hw_reset, 595
 - ata_device_init, 595
 - ata_device_write, 595
 - ata_devices_hw_reset, 596
 - ata_devices_init, 596
 - ata_devices_read, 596
 - ata_devices_write, 596
 - ata_pretty_status, 597
 - DEFAULT_DEBUG_CHANNEL, 597
 - open_file, 597
 - open_local, 597
 - atadevice.h
 - ATA_ASR, 601
 - ATA_CHR, 601
 - ATA_CLR, 601
 - ATA_CR, 601
 - ATA_DA, 601
 - ATA_DAR_DS0, 601
 - ATA_DAR_DS1, 601
 - ATA_DAR_H, 601
 - ATA_DAR_WTG, 601
 - ATA_DCR, 601
 - ATA_DCR_IEN, 601
 - ATA_DCR_RST, 601
 - ata_devices_hw_reset, 601
 - ata_devices_init, 602
 - ata_devices_read, 602
 - ata_devices_write, 602
 - ATA_DHR, 601
 - ATA_DHR_DEV, 601
 - ATA_DHR_H, 601
 - ATA_DHR_LBA, 601
 - ATA_DR, 601
 - ATA_ERR, 601
 - ATA_ERR_ABT, 601
 - ATA_ERR_AMN, 601
 - ATA_ERR_BBK, 601
 - ATA_ERR_IDNF, 601
 - ATA_ERR_TON, 601
 - ATA_ERR_UNC, 601
 - ATA_FR, 601
 - ATA_SCR, 601
 - ATA_SNR, 601
 - ATA_SR, 601
 - ATA_SR_BSY, 601
 - ATA_SR_COR, 601
 - ATA_SR_DF, 601
 - ATA_SR_DRDY, 601
 - ATA_SR_DRQ, 601
 - ATA_SR_DSC, 601

- ATA_SR_ERR, 601
- ATA_SR_IDX, 601
- ATA_STATE_HW_RST, 601
- ATA_STATE_IDLE, 601
- ATA_STATE_SW_RST, 601
- TYPE_FILE, 601
- TYPE_LOCAL, 601
- TYPE_NO_CONNECT, 601
- atadevice_cmdi.c
 - ata_calc_lba, 604
 - ata_cmd_complete, 604
 - ata_device_execute_cmd, 604
 - ata_device_reset_cmd, 604
 - ata_execute_device_diagnostics_cmd, 604
 - ata_identify_device_cmd, 605
 - ata_initialize_device_parameters_cmd, 605
 - ata_read_native_max_addr, 605
 - ata_read_sect, 605
 - ata_read_sectors_cmd, 605
 - ata_set_device_signature, 606
 - ata_set_features, 606
 - ata_set_sect, 606
 - ata_write_sect, 606
 - ata_write_sectors, 606
 - DEFAULT_DEBUG_CHANNEL, 607
- atadevice_cmdi.h
 - ata_device_execute_cmd, 611
 - ata_execute_device_diagnostics_cmd, 612
 - BYTES_PER_SECTOR, 611
 - MIN_MWDMA_CYCLE_TIME, 611
 - MIN_PIO_CYCLE_TIME_IORDY, 611
 - MIN_PIO_CYCLE_TIME_NO_IORDY, 611
 - QUEUE_DEPTH, 611
 - RECOMMENDED_MWDMA_CYCLE_-
TIME, 611
 - SET_FEATURES_REQUIRED_AFTER_-
POWER_UP, 611
 - SUPPORT_APM, 611
 - SUPPORT_CFA, 611
 - SUPPORT_DEVICE_RESET_CMD, 611
 - SUPPORT_DOWNLOAD_MICROCODE,
611
 - SUPPORT_HOST_PROTECTED_AREA,
611
 - SUPPORT_LOOKAHEAD, 611
 - SUPPORT_NOP_CMD, 611
 - SUPPORT_POWER_MANAGEMENT, 611
 - SUPPORT_POWER_UP_IN_STANDBY_-
MODE, 611
 - SUPPORT_READ_BUFFER_CMD, 611
 - SUPPORT_READ_WRITE_DMA_-
QUEUED, 611
 - SUPPORT_RELEASE_INTERRUPT, 611
 - SUPPORT_REMOVABLE_MEDIA, 611
 - SUPPORT_REMOVABLE_MEDIA_-
NOTIFICATION, 611
 - SUPPORT_SECURITY_MODE, 611
 - SUPPORT_SERVICE_INTERRUPT, 611
 - SUPPORT_SET_MAX, 611
 - SUPPORT_SMART, 611
 - SUPPORT_WRITE_BUFFER_CMD, 611
 - SUPPORT_WRITE_CACHE, 611
- atahost.c
 - ata_baseaddr, 615
 - ata_dev_id, 615
 - ata_dma_mode0_td, 615
 - ata_dma_mode0_teoc, 615
 - ata_dma_mode0_tm, 615
 - ata_enabled, 615
 - ata_enddevice, 615
 - ata_file, 615
 - ata_firmware, 615
 - ata_heads, 616
 - ata_int, 616
 - ata_irq, 616
 - ata_mwdma, 616
 - ata_packet, 616
 - ata_pio, 616
 - ata_pio_mode0_t1, 616
 - ata_pio_mode0_t2, 617
 - ata_pio_mode0_t4, 617
 - ata_pio_mode0_teoc, 617
 - ata_read32, 617
 - ata_reset, 617
 - ata_rev, 617
 - ata_sec_end, 617
 - ata_sec_start, 618
 - ata_sectors, 618
 - ata_size, 619
 - ata_start_device, 619
 - ata_status, 619
 - ata_type, 619
 - ata_write32, 619
 - conf_dev, 622
 - DEFAULT_DEBUG_CHANNEL, 619
 - DMA_MODE0_TD, 615
 - DMA_MODE0_TEOC, 615
 - DMA_MODE0_TM, 615
 - PIO_MODE0_T1, 615
 - PIO_MODE0_T2, 615
 - PIO_MODE0_T4, 615
 - PIO_MODE0_TEOC, 615
 - reg_ata_sec, 621
- atahost.h
 - ATA_ADDR_SPACE, 626
 - ATA_BELEC0, 626
 - ATA_BELEC1, 626
 - ATA_CTRL, 626

- ATA_DEVID, 626
- ata_dma_delay, 626
- ATA_DMA_EN, 626
- ATA_DMA_RD, 626
- ATA_DMA_TIP, 626
- ATA_DMA_WR, 626
- ATA_DMARQ, 626
- ATA_DRBE, 626
- ATA_DTBF, 626
- ATA_DTR0, 626
- ATA_DTR1, 626
- ATA_FTE0, 626
- ATA_FTE1, 626
- ATA_IDE_EN, 626
- ATA_IDEIS, 626
- ata_int, 626
- ATA_IORDY, 626
- ATA_IORDY_FTE0, 626
- ATA_IORDY_FTE1, 626
- ATA_PCTR, 626
- ATA_PFTR0, 626
- ATA_PFTR1, 626
- ata_pio_delay, 626
- ATA_PIO_TIP, 626
- ATA_PWPP, 626
- ATA_PWPPF, 626
- ATA_REVNO, 626
- ATA_RST, 626
- ATA_RXB, 626
- ATA_STAT, 626
- ATA_T1, 626
- ATA_T2, 626
- ATA_T4, 626
- ATA_TD, 626
- ATA_TEOC, 626
- ATA_TM, 626
- ATA_TXB, 626
- is_ata_hostadr, 626
- reg_ata_sec, 627
- atahost_define.h
 - DMA_MODE0_TD, 629
 - DMA_MODE0_TEOC, 629
 - DMA_MODE0_TM, 629
 - PIO_MODE0_T1, 629
 - PIO_MODE0_T2, 629
 - PIO_MODE0_T4, 629
 - PIO_MODE0_TEOC, 629
- audio_cnt
 - sprs.c, 316
- automata
 - or32.c, 431
- aux
 - gpio_device, 121
- auxiliary_inputs
 - gpio_device, 121
- b
 - cuc_timings, 66
- ba_mask
 - mc, 144
- backward
 - branchstat, 30
- base_id
 - vapi_handler, 173
- base_include
 - sim-config.c, 774
- base_vapi_id
 - eth_device, 110
 - gpio_device, 121
- baseaddr
 - ata_host, 24
 - dev_16450, 71
 - dev_generic, 74
 - dma_controller, 81
 - eth_device, 110
 - fb_state, 115
 - gpio_device, 121
 - kbd_state, 141
 - mc, 144
 - mem_config, 147
 - vga_state, 175
- basename
 - xterm.c, 646
- baud_table
 - tty.c, 643
- bb
 - _cuc_func, 14
- bb.c
 - build_bb, 439
 - count_bb_seq, 439
 - cpy_bb, 439
 - cuc_check, 439
 - detect_bb, 439
 - dup_func, 440
 - expand_bb, 440
 - free_func, 440
 - generate_bb_seq, 440
 - join_bb, 440
 - optimize_bb, 441
 - preunroll_loop, 441
 - print_bb_num, 441
 - print_cuc_bb, 442
 - recalc_last_used_reg, 442
 - reg_dep, 442
 - reg_dep_rec, 442
 - relocate_bb, 442
 - remove_dead_bb, 442
 - roll_loop, 442

- simplify_bb, 442
- BB_DEAD
 - cuc.h, 465
- BB_INLOOP
 - cuc.h, 465
- BB_OPTIONAL
 - cuc.h, 465
- bb_size
 - timings.c, 504
- BBID_END
 - cuc.h, 465
- BBID_START
 - cuc.h, 465
- bd
 - eth_device, 110
- bd_addr
 - eth_device, 110
- bd_index
 - eth_device, 110
- bd_ram
 - eth_device, 110
- bf
 - mstats_entry, 152
- bff, 26
 - close_obj, 26
 - get_func_len, 26
 - get_func_name, 26
 - get_func_reloc, 26
 - get_func_start, 26
 - open_obj, 26
- bffs
 - dyngen.h, 355
 - dyngen_elf.c, 357
- bit_mask
 - abstract.c, 207
- bits
 - INFOHEADER, 128
 - spr_def, 168
- block_jtag
 - gdbcomm.c, 527
 - gdbcomm.h, 530
- block_mask
 - ic, 124
- block_offset_mask
 - ic, 124
- blocksize
 - config, 52
 - ic, 124
- blocksize_log2
 - ic, 124
- BMP_HEADER, 27
 - offset, 27
 - reserved1, 27
 - reserved2, 27
- size, 27
- type, 27
- bnf
 - mstats_entry, 152
- boolean
 - gdbcomm.h, 530
- bpb
 - branch-predict.c, 181
 - config, 52
 - mstats_entry, 152
- bpb/branch-predict.c, 177
- bpb/branch-predict.h, 182
- bpb_btic
 - branch-predict.c, 180
- bpb_enabled
 - branch-predict.c, 180
- bpb_entry, 28
 - addr, 28
 - lru, 28
 - taken, 28
 - way, 28
- bpb_hitdelay
 - branch-predict.c, 180
- bpb_info
 - branch-predict.c, 180
 - branch-predict.h, 183
- BPB_LEN
 - branch-predict.c, 180
- bpb_missdelay
 - branch-predict.c, 180
- BPB_PSTATES
 - branch-predict.c, 180
- bpb_sbp_bf_fwd
 - branch-predict.c, 180
- bpb_sbp_bnf_fwd
 - branch-predict.c, 180
- bpb_update
 - branch-predict.c, 180
 - branch-predict.h, 183
- BPB_USTATES
 - branch-predict.c, 180
- BPB_WAYS
 - branch-predict.c, 180
- bpbstat, 29
 - correct, 29
 - hit, 29
 - incorrect, 29
 - miss, 29
- branch-predict.c
 - bpb, 181
 - bpb_btic, 180
 - bpb_enabled, 180
 - bpb_hitdelay, 180
 - bpb_info, 180

- BPB_LEN, 180
- bpb_missdelay, 180
- BPB_PSTATES, 180
- bpb_sbp_bf_fwd, 180
- bpb_sbp_bnf_fwd, 180
- bpb_update, 180
- BPB_USTATES, 180
- BPB_WAYS, 180
- btic, 181
- BTIC_BLOCKSIZE, 180
- btic_info, 180
- BTIC_LEN, 180
- btic_update, 180
- BTIC_USTATES, 180
- BTIC_WAYS, 180
- reg_bpb_sec, 180
- branch-predict.h
 - bpb_info, 183
 - bpb_update, 183
 - btic_info, 183
 - btic_update, 183
 - reg_bpb_sec, 183
- branch_index
 - verilog.c, 507
- branchstat, 30
 - backward, 30
 - forward, 30
 - nottaken, 30
 - taken, 30
- breakpoint
 - execute.c, 368
- breakpoint_entry, 31
 - addr, 31
 - next, 31
- breakpoints
 - labels.c, 262
 - labels.h, 264
- bsize
 - COFF_AOUTHDR, 38
- btic
 - branch-predict.c, 181
 - config, 52
 - mstats_entry, 152
- BTIC_BLOCKSIZE
 - branch-predict.c, 180
- btic_entry, 32
 - addr, 32
 - insn, 32
 - lru, 32
 - way, 32
- btic_info
 - branch-predict.c, 180
 - branch-predict.h, 183
- BTIC_LEN
 - branch-predict.c, 180
- btic_update
 - branch-predict.c, 180
 - branch-predict.h, 183
- BTIC_USTATES
 - branch-predict.c, 180
- BTIC_WAYS
 - branch-predict.c, 180
- bticstat, 33
 - hit, 33
 - miss, 33
- buf
 - kbd_state, 141
- buf_count
 - kbd_state, 141
- buf_head
 - kbd_state, 141
- BUF_SIZE
 - mprofiler.c, 557
- buf_tail
 - kbd_state, 141
- build_automata
 - or32.c, 428
- build_bb
 - bb.c, 439
 - cuc.h, 466
- build_insn
 - load.c, 491
- byte_enabled
 - dev_generic, 74
- byteadd
 - mstats_entry, 152
- bytes_left
 - eth_device, 110
- BYTES_PER_SECTOR
 - atadevice_cmdi.h, 611
- bytes_read
 - eth_device, 110
- bytes_sent
 - eth_device, 110
- C_ALIAS
 - coff.h, 242
- C_ARG
 - coff.h, 242
- C_AUTO
 - coff.h, 242
- C_AUTOARG
 - coff.h, 242
- C_BCOMM
 - coff.h, 242
- C_BINCL
 - coff.h, 242
- C_BLOCK

- coff.h, 242
- C_BSTAT
 - coff.h, 242
- C_DECL
 - coff.h, 242
- C_DEFINE
 - coff.h, 242
- C_ECOML
 - coff.h, 242
- C_ECOMM
 - coff.h, 242
- C_EFCN
 - coff.h, 242
- C_EINCL
 - coff.h, 242
- C_ENTAG
 - coff.h, 242
- C_ENTRY
 - coff.h, 242
- C_EOS
 - coff.h, 242
- C_ESTAT
 - coff.h, 242
- C_EXT
 - coff.h, 242
- C_EXTDEF
 - coff.h, 242
- C_EXTLAB
 - coff.h, 242
- C_FCNC
 - coff.h, 242
- C_FIELD
 - coff.h, 242
- C_FILE
 - coff.h, 242
- c_file_head
 - dyngen.c, 353
- C_FUN
 - coff.h, 242
- C_GSYM
 - coff.h, 242
- C_HIDDEN
 - coff.h, 242
- C_HIDEXT
 - coff.h, 242
- C_LABEL
 - coff.h, 242
- C_LASTENT
 - coff.h, 242
- C_LEAFEXT
 - coff.h, 242
- C_LEAFPROC
 - coff.h, 242
- C_LEAFSTAT
 - coff.h, 242
- C_LINE
 - coff.h, 242
- C_LSYM
 - coff.h, 242
- C_MOE
 - coff.h, 242
- C_MOS
 - coff.h, 242
- C_MOU
 - coff.h, 242
- C_NT_WEAK
 - coff.h, 242
- C_NULL
 - coff.h, 242
- C_OPTVAR
 - coff.h, 242
- C_PRAGMA
 - coff.h, 242
- C_PSYM
 - coff.h, 242
- C_REG
 - coff.h, 242
- C_REGPARAM
 - coff.h, 242
- c_rel_file_head
 - dyngen.c, 353
- c_rel_file_tail
 - dyngen.c, 353
- C_RPSYM
 - coff.h, 242
- C_RSYM
 - coff.h, 242
- C_SCALL
 - coff.h, 242
- C_SECTION
 - coff.h, 242
- C_SEGMENT
 - coff.h, 242
- C_SHADOW
 - coff.h, 242
- C_STAT
 - coff.h, 242
- C_STATLAB
 - coff.h, 242
- C_STRTAG
 - coff.h, 242
- C_STSYM
 - coff.h, 242
- c_sw_file_head
 - dyngen.c, 353
- c_sw_file_tail
 - dyngen.c, 354
- C_SYSTEM

- coeff.h**, 242
- C_TCSYM**
 - coeff.h**, 242
- C_THUMBEXT**
 - coeff.h**, 242
- C_THUMBEXTFUNC**
 - coeff.h**, 242
- C_THUMBLABEL**
 - coeff.h**, 242
- C_THUMBSTAT**
 - coeff.h**, 242
- C_THUMBSTATFUNC**
 - coeff.h**, 242
- C_TPDEF**
 - coeff.h**, 242
- C_UEXT**
 - coeff.h**, 242
- C_ULABEL**
 - coeff.h**, 242
- C_UNTAG**
 - coeff.h**, 242
- C_USTATIC**
 - coeff.h**, 242
- C_VERSION**
 - coeff.h**, 242
- C_WEAKEXT**
 - coeff.h**, 242
- cache/dcache-model.c**, 184
- cache/dcache-model.h**, 189
- cache/icache-model.c**, 192
- cache/icache-model.h**, 197
- cachestats_entry**, 34
 - readhit**, 34
 - readmiss**, 34
 - writehit**, 34
 - writemiss**, 34
- calc_cycles**
 - cuc.c**, 448
- calc_max**
 - adv.c**, 437
- calc_size**
 - cuc.c**, 448
- calculate_watchpoints**
 - debug-unit.c**, 512
- caller_saved**
 - cuc.c**, 458
 - cuc.h**, 477
- calling_convention**
 - config**, 52
- calls**
 - func_struct**, 119
- cam_addr**
 - fb_state**, 115
- CAM_SIZEY**
 - fb.c**, 685
- CAM_SIZEX**
 - fb.c**, 685
- camera_pos**
 - fb_state**, 115
- camerax**
 - fb_state**, 115
- cameray**
 - fb_state**, 115
- ccmd**
 - kbd_state**, 141
- ccmdbyte**
 - kbd_state**, 141
- ce**
 - mem_config**, 147
- CFA_DISABLE_8BIT_PIO_TRANSFER_MODE**
 - atacmd.h**, 593
- CFA_DISABLE_POWER_MODE1**
 - atacmd.h**, 593
- CFA_ENABLE_8BIT_PIO_TRANSFER_MODE**
 - atacmd.h**, 593
- CFA_ENABLE_POWER_MODE1**
 - atacmd.h**, 593
- CFA_ERASE_SECTORS**
 - atacmd.h**, 593
- CFA_REQUEST_EXTENDED_ERROR_CODE**
 - atacmd.h**, 593
- CFA_TRANSLATE_SECTOR**
 - atacmd.h**, 593
- CFA_WRITE_MULTIPLE_WITHOUT_ERASE**
 - atacmd.h**, 593
- CFA_WRITE_SECTORS_WITHOUT_ERASE**
 - atacmd.h**, 593
- ch**
 - dma_controller**, 81
- chain**
 - jtr_chain_message**, 130
- change_buf_addr**
 - fb.c**, 686
- change_insn_type**
 - insn.c**, 480
 - insn.h**, 488
- channel**, 35
 - data**, 35
 - dev_16450**, 71
 - ops**, 35
- channel.c**
 - channel_close**, 631
 - channel_init**, 631
 - channel_open**, 631
 - channel_read**, 631
 - channel_write**, 631
 - find_channel_factory**, 631
 - head**, 631

- preloaded, 631
- channel.h
 - channel_close, 632
 - channel_init, 632
 - channel_open, 632
 - channel_read, 632
 - channel_write, 632
- channel_close
 - channel.c, 631
 - channel.h, 632
- channel_factory, 36
 - name, 36
 - next, 36
 - ops, 36
- channel_init
 - channel.c, 631
 - channel.h, 632
- channel_mask
 - dma_channel, 78
- CHANNEL_ND_I
 - dma.c, 659
- channel_number
 - dma_channel, 78
- channel_open
 - channel.c, 631
 - channel.h, 632
- channel_ops, 37
 - close, 37
 - free, 37
 - init, 37
 - isok, 37
 - open, 37
 - read, 37
 - status, 37
 - write, 37
- channel_read
 - channel.c, 631
 - channel.h, 632
- channel_str
 - dev_16450, 71
- channel_write
 - channel.c, 631
 - channel.h, 632
- channels/generic.c
 - generic_close, 698
 - generic_free, 698
 - generic_open, 698
- channels/generic.h
 - generic_close, 700
 - generic_free, 700
 - generic_open, 700
- char_clks
 - 16450.c, 573
 - dev_16450, 71
- check_debug_unit
 - debug-unit.c, 512
 - debug-unit.h, 520
- check_depend
 - execute.c, 362
- check_dma_ack_o
 - dma.c, 659
 - dma.h, 665
- check_insn_exec
 - sim-cmd.c, 757
- check_int
 - toplevel-support.c, 826
 - toplevel-support.h, 831
- CHECK_INT_TIME
 - sim-config.h, 782
- check_memory_conflict
 - cuc/memory.c, 496
- CHECK_POWER_MODE
 - atacmd.h, 593
- chunk_size
 - dma_channel, 78
- class_ptr
 - config, 52
- clean_deps
 - cuc.h, 467
 - cuc/memory.c, 496
- clear_dma_nd_i
 - dma.c, 659
 - dma.h, 665
- clear_dma_req_i
 - dma.c, 659
 - dma.h, 665
- CLEAR_FLAG
 - fields.h, 691
- clear_interrupt
 - pic.c, 738
 - pic.h, 741
- clkcycle_ps
 - config, 52
- close
 - channel_ops, 37
- close_obj
 - bff, 26
- cmatch
 - _csm_list, 12
 - cuc_shared_item, 64
- cmd_handle
 - sim_command, 164
- cmov_needed
 - insn.c, 480
- cmovs
 - _csm_list, 12
- cnt
 - _csm_list, 12

- cuc_bb, 60
- memory_hash, 150
- cnt_dynamic
 - dstats_entry, 85
 - fstats_entry, 118
 - sstats_entry, 169
- CNV16
 - fb.c, 685
- CNV32
 - fb.c, 685
- code
 - ps2kbd.c, 728
- coff.h
 - C_ALIAS, 242
 - C_ARG, 242
 - C_AUTO, 242
 - C_AUTOARG, 242
 - C_BCOMM, 242
 - C_BINCL, 242
 - C_BLOCK, 242
 - C_BSTAT, 242
 - C_DECL, 242
 - C_DEFINE, 242
 - C_ECOML, 242
 - C_ECOMM, 242
 - C_EFCN, 242
 - C_EINCL, 242
 - C_ENTAG, 242
 - C_ENTRY, 242
 - C_EOS, 242
 - C_ESTAT, 242
 - C_EXT, 242
 - C_EXTDEF, 242
 - C_EXTLAB, 242
 - C_FCNC, 242
 - C_FIELD, 242
 - C_FILE, 242
 - C_FUN, 242
 - C_GSYM, 242
 - C_HIDDEN, 242
 - C_HIDEEXT, 242
 - C_LABEL, 242
 - C_LASTENT, 242
 - C_LEAFEXT, 242
 - C_LEAFPROC, 242
 - C_LEAFSTAT, 242
 - C_LINE, 242
 - C_LSYM, 242
 - C_MOE, 242
 - C_MOS, 242
 - C_MOU, 242
 - C_NT_WEAK, 242
 - C_NULL, 242
 - C_OPTVAR, 242
 - C_PRAGMA, 242
 - C_PSYM, 242
 - C_REG, 242
 - C_REGPARM, 242
 - C_RPSYM, 242
 - C_RSYM, 242
 - C_SCALL, 242
 - C_SECTION, 242
 - C_SEGMENT, 242
 - C_SHADOW, 242
 - C_STAT, 242
 - C_STATLAB, 242
 - C_STRTAG, 242
 - C_STSYM, 242
 - C_SYSTEM, 242
 - C_TCSYM, 242
 - C_THUMBEXT, 242
 - C_THUMBEXTFUNC, 242
 - C_THUMBLABEL, 242
 - C_THUMBSTAT, 242
 - C_THUMBSTATFUNC, 242
 - C_TPDEF, 242
 - C_UEXT, 242
 - C_ULABEL, 242
 - C_UNTAG, 242
 - C_USTATIC, 242
 - C_VERSION, 242
 - C_WEAKEXT, 242
 - COFF_AOUTSZ, 242
 - COFF_AUXENT, 242
 - COFF_AUXESZ, 242
 - COFF_BSS, 242
 - COFF_COMMENT, 242
 - COFF_DATA, 242
 - COFF_DEF_BSS_SECTION_ALIGNMENT, 242
 - COFF_DEF_DATA_SECTION_ALIGNMENT, 242
 - COFF_DEF_SECTION_ALIGNMENT, 242
 - COFF_DEF_TEXT_SECTION_ALIGNMENT, 242
 - COFF_DMAGIC, 242
 - COFF_E_DIMNUM, 242
 - COFF_E_FILNMLEN, 242
 - COFF_E_SYMNMLEN, 242
 - COFF_ETEXT, 242
 - COFF_F_AR16WR, 242
 - COFF_F_AR32W, 242
 - COFF_F_AR32WR, 242
 - COFF_F_EXEC, 242
 - COFF_F_LNNO, 242
 - COFF_F_LSYMS, 242
 - COFF_F_MINMAL, 242
 - COFF_F_NODF, 242

- COFF_F_PATCH, 242
- COFF_F_RELFLG, 242
- COFF_F_SWABD, 242
- COFF_F_UPDATE, 242
- COFF_FILHDR, 242
- COFF_FILHSZ, 242
- COFF_I386BADMAG, 242
- COFF_I386MAGIC, 242
- COFF_JMAGIC, 242
- COFF_LIB, 242
- COFF_LINENO, 242
- COFF_LINESZ, 242
- COFF_LONG, 242
- COFF_LONG_H, 242
- COFF_LONG_L, 242
- COFF_N_BTMASK, 242
- COFF_N_BTSHFT, 242
- COFF_N_TMASK, 242
- COFF_N_TSHIFT, 242
- COFF_OMAGIC, 242
- COFF_RELOC, 242
- COFF_RELSZ, 242
- COFF_SCNHDR, 242
- COFF_SCNHSZ, 242
- COFF_SECT_BSS, 242
- COFF_SECT_DATA, 242
- COFF_SECT_REQD, 242
- COFF_SECT_TEXT, 242
- COFF_SHMAGIC, 242
- COFF_SHORT, 242
- COFF_SHORT_H, 242
- COFF_SHORT_L, 242
- COFF_SLIBHD, 242
- COFF_SLIBSZ, 242
- COFF_STMAGIC, 242
- COFF_STYP_BSS, 242
- COFF_STYP_COPY, 242
- COFF_STYP_DATA, 242
- COFF_STYP_DSECT, 242
- COFF_STYP_GROUP, 242
- COFF_STYP_INFO, 242
- COFF_STYP_LIB, 242
- COFF_STYP_NOLOAD, 242
- COFF_STYP_OVER, 242
- COFF_STYP_PAD, 242
- COFF_STYP_REG, 242
- COFF_STYP_TEXT, 242
- COFF_SYMENT, 242
- COFF_SYMESZ, 242
- COFF_TEXT, 242
- COFF_ZMAGIC, 242
- E_DIMNUM, 242
- E_FILNMLEN, 242
- E_SYMNMLEN, 242
- KEEP_ENDIAN_LONG, 242
- KEEP_ENDIAN_SHORT, 243
- SWAP_ENDIAN_LONG, 243
- SWAP_ENDIAN_SHORT, 243
- COFF_AOUTHDR, 38
 - bsize, 38
 - data_start, 38
 - dsize, 38
 - entry, 38
 - magic, 38
 - text_start, 38
 - tsize, 38
 - vstamp, 38
- COFF_AOUTSZ
 - coff.h, 242
- COFF_AUXENT
 - coff.h, 242
- COFF_auxent, 39
 - x_ary, 41
 - x_dimen, 41
 - x_endndx, 41
 - x_fcn, 41
 - x_fcny, 41
 - x_file, 41
 - x_fname, 41
 - x_fsize, 41
 - x_inno, 41
 - x_innopt, 41
 - x_lnsz, 41
 - x_misc, 41
 - x_n, 41
 - x_nlinno, 41
 - x_nreloc, 41
 - x_offset, 41
 - x_scn, 41
 - x_scnlen, 41
 - x_size, 41
 - x_sym, 41
 - x_tagndx, 41
 - x_tv, 41
 - x_tvfill, 41
 - x_tvlen, 41
 - x_tvndx, 41
 - x_tvrn, 41
 - x_zeroes, 41
- COFF_AUXESZ
 - coff.h, 242
- COFF_BSS
 - coff.h, 242
- COFF_COMMENT
 - coff.h, 242
- COFF_DATA
 - coff.h, 242
- COFF_DEF_BSS_SECTION_ALIGNMENT

- [coff.h](#), [242](#)
- COFF_DEF_DATA_SECTION_ALIGNMENT
 - [coff.h](#), [242](#)
- COFF_DEF_SECTION_ALIGNMENT
 - [coff.h](#), [242](#)
- COFF_DEF_TEXT_SECTION_ALIGNMENT
 - [coff.h](#), [242](#)
- COFF_DMAGIC
 - [coff.h](#), [242](#)
- COFF_E_DIMNUM
 - [coff.h](#), [242](#)
- COFF_E_FILNMLEN
 - [coff.h](#), [242](#)
- COFF_E_SYMNMLEN
 - [coff.h](#), [242](#)
- COFF_ETEXT
 - [coff.h](#), [242](#)
- COFF_F_AR16WR
 - [coff.h](#), [242](#)
- COFF_F_AR32W
 - [coff.h](#), [242](#)
- COFF_F_AR32WR
 - [coff.h](#), [242](#)
- COFF_F_EXEC
 - [coff.h](#), [242](#)
- COFF_F_LNNO
 - [coff.h](#), [242](#)
- COFF_F_LSYMS
 - [coff.h](#), [242](#)
- COFF_F_MINMAL
 - [coff.h](#), [242](#)
- COFF_F_NODF
 - [coff.h](#), [242](#)
- COFF_F_PATCH
 - [coff.h](#), [242](#)
- COFF_F_RELFLG
 - [coff.h](#), [242](#)
- COFF_F_SWABD
 - [coff.h](#), [242](#)
- COFF_F_UPDATE
 - [coff.h](#), [242](#)
- COFF_filehdr, [43](#)
 - [f_flags](#), [43](#)
 - [f_magic](#), [43](#)
 - [f_nscns](#), [43](#)
 - [f_nsyms](#), [43](#)
 - [f_opthdr](#), [43](#)
 - [f_symptr](#), [43](#)
 - [f_timdat](#), [43](#)
- COFF_FILHDR
 - [coff.h](#), [242](#)
- COFF_FILHSZ
 - [coff.h](#), [242](#)
- COFF_I386BADMAG
 - [coff.h](#), [242](#)
- COFF_I386MAGIC
 - [coff.h](#), [242](#)
- COFF_JMAGIC
 - [coff.h](#), [242](#)
- COFF_LIB
 - [coff.h](#), [242](#)
- COFF_LINENO
 - [coff.h](#), [242](#)
- COFF_lineno, [44](#)
 - [l_addr](#), [44](#)
 - [l_inno](#), [44](#)
 - [l_paddr](#), [44](#)
 - [l_symndx](#), [44](#)
- COFF_LINESZ
 - [coff.h](#), [242](#)
- COFF_LONG
 - [coff.h](#), [242](#)
- COFF_LONG_H
 - [coff.h](#), [242](#)
- COFF_LONG_L
 - [coff.h](#), [242](#)
- COFF_N_BTMASK
 - [coff.h](#), [242](#)
- COFF_N_BTSHFT
 - [coff.h](#), [242](#)
- COFF_N_TMASK
 - [coff.h](#), [242](#)
- COFF_N_TSHIFT
 - [coff.h](#), [242](#)
- COFF_OMAGIC
 - [coff.h](#), [242](#)
- COFF_RELOC
 - [coff.h](#), [242](#)
- COFF_reloc, [45](#)
 - [r_symndx](#), [45](#)
 - [r_type](#), [45](#)
 - [r_vaddr](#), [45](#)
- COFF_RELSZ
 - [coff.h](#), [242](#)
- COFF_SCNHDR
 - [coff.h](#), [242](#)
- COFF_scnhdr, [46](#)
 - [s_flags](#), [46](#)
 - [s_innoptr](#), [46](#)
 - [s_name](#), [46](#)
 - [s_nlnno](#), [46](#)
 - [s_nreloc](#), [46](#)
 - [s_paddr](#), [46](#)
 - [s_relptr](#), [46](#)
 - [s_scnptr](#), [46](#)
 - [s_size](#), [46](#)
 - [s_vaddr](#), [46](#)
- COFF_SCNHSZ

- coff.h, 242
- COFF_SECT_BSS
 - coff.h, 242
- COFF_SECT_DATA
 - coff.h, 242
- COFF_SECT_REQD
 - coff.h, 242
- COFF_SECT_TEXT
 - coff.h, 242
- COFF_SHMAGIC
 - coff.h, 242
- COFF_SHORT
 - coff.h, 242
- COFF_SHORT_H
 - coff.h, 242
- COFF_SHORT_L
 - coff.h, 242
- COFF_slib, 47
 - sl_entsz, 47
 - sl_pathndx, 47
- COFF_SLIBHD
 - coff.h, 242
- COFF_SLIBSZ
 - coff.h, 242
- COFF_STMAGIC
 - coff.h, 242
- COFF_STYP_BSS
 - coff.h, 242
- COFF_STYP_COPY
 - coff.h, 242
- COFF_STYP_DATA
 - coff.h, 242
- COFF_STYP_DSECT
 - coff.h, 242
- COFF_STYP_GROUP
 - coff.h, 242
- COFF_STYP_INFO
 - coff.h, 242
- COFF_STYP_LIB
 - coff.h, 242
- COFF_STYP_NOLOAD
 - coff.h, 242
- COFF_STYP_OVER
 - coff.h, 242
- COFF_STYP_PAD
 - coff.h, 242
- COFF_STYP_REG
 - coff.h, 242
- COFF_STYP_TEXT
 - coff.h, 242
- COFF_SYMENT
 - coff.h, 242
- COFF_syment, 48
 - e, 48
 - e_name, 48
 - e_numaux, 48
 - e_offset, 48
 - e_sclass, 48
 - e_scnnum, 48
 - e_type, 48
 - e_value, 48
 - e_zeroes, 48
- COFF_SYMESZ
 - coff.h, 242
- COFF_TEXT
 - coff.h, 242
- COFF_ZMAGIC
 - coff.h, 242
- collconf
 - eth_device, 110
- command
 - ata_device, 20
 - jtr_chain_message, 130
 - jtr_read_block_message, 133
 - jtr_read_message, 135
 - jtr_write_block_message, 137
 - jtr_write_message, 139
- common_i386.h
 - get_pc, 319
 - high32, 319
 - low32, 319
 - set_pc, 319
 - upd_sim_cycles, 319
 - useless_x86, 319
 - val3232, 319
 - val64, 319
- COMP
 - op.c, 394
- COMP_CAST
 - op.c, 394
- COMP_NAME
 - op.c, 394
- compression
 - INFOHEADER, 128
- comutative
 - cuc_known_insn, 63
- conf
 - ata_device, 20
- conf_dev
 - atahost.c, 622
- config, 49
 - blocksize, 52
 - bpb, 52
 - btic, 52
 - calling_convention, 52
 - class_ptr, 52
 - clkcycle_ps, 52
 - cpu, 52

- cuc, 52
- dc, 52
- debug, 52
- dependstats, 52
- enable_bursts, 52
- enabled, 52
- exe_log, 52
- exe_log_end, 52
- exe_log_fn, 52
- exe_log_marker, 52
- exe_log_start, 52
- exe_log_type, 52
- ext, 52
- gdb_enabled, 52
- hazards, 52
- hide_device_id, 52
- history, 52
- hitdelay, 52
- load_hitdelay, 52
- load_missdelay, 52
- log_enabled, 52
- memory_order, 52
- missdelay, 52
- mprof_fn, 52
- mprofile, 52
- no_multicycle, 52
- nsets, 52
- nways, 52
- pic, 52
- pm, 52
- prof_fn, 52
- profile, 52
- read_up, 52
- sbp_bf_fwd, 52
- sbp_bnf_fwd, 52
- sbuf_len, 52
- server_port, 52
- sim, 52
- sim-config.c, 780
- sim-config.h, 787
- store_hitdelay, 52
- store_missdelay, 52
- superscalar, 52
- timings_fn, 52
- ustates, 52
- vapi, 52
- vapi_fn, 52
- vapi_id, 52
- verbose, 52
- write_up, 52
- config::pic, 54
 - edge_trigger, 54
 - enabled, 54
- CONFIG_ERROR
 - sim-config.h, 782
- config_param, 55
 - func, 55
 - name, 55
 - next, 55
 - type, 55
- config_section, 56
 - dat, 56
 - name, 56
 - next, 56
 - params, 56
 - sec_end, 56
 - sec_start, 56
- connected
 - tcp_channel, 171
- controller
 - dma_channel, 78
- controlmoder
 - eth_device, 110
- conv
 - load.c, 494
- correct
 - bpstat, 29
- count_bb_seq
 - bb.c, 439
- count_cmovs
 - insn.c, 480
- cover_insn
 - or32.c, 428
- cpu
 - config, 52
 - runtime, 160
- cpu-config.c, 200
 - cpu_cfg, 201
 - cpu_cfg, 201
 - cpu_dependstats, 201
 - cpu_hazards, 201
 - cpu_rev, 201
 - cpu_sbuf_len, 201
 - cpu_sr, 201
 - cpu_superscalar, 202
 - cpu_upr, 202
 - cpu_ver, 202
 - reg_cpu_sec, 202
 - WARNING, 201
- cpu-config.h, 204
 - reg_cpu_sec, 204
- cpu/common/abstract.c, 205
- cpu/common/abstract.h, 222
- cpu/common/coff.h, 237
- cpu/common/elf.h, 244
- cpu/common/execute.h, 254
- cpu/common/labels.c, 261
- cpu/common/labels.h, 263

- cpu/common/parse.c, 265
- cpu/common/parse.h, 271
- cpu/common/stats.c, 273
- cpu/common/stats.h, 277
- cpu/common/trace.c, 279
- cpu/common/trace.h, 281
- cpu/or1k/arch.h, 283
- cpu/or1k/except.c, 285
- cpu/or1k/except.h, 287
- cpu/or1k/spr-defs.h, 291
- cpu/or1k/spr-dump.c, 301
- cpu/or1k/spr-dump.h, 313
- cpu/or1k/sprs.c, 314
- cpu/or1k/sprs.h, 317
- cpu/or32/common_i386.h, 319
- cpu/or32/def_op_t.h, 320
- cpu/or32/dyn32_defs.h, 322
- cpu/or32/dyn_rec.c, 323
- cpu/or32/dyn_rec.h, 347
- cpu/or32/dyngen.c, 352
- cpu/or32/dyngen.h, 355
- cpu/or32/dyngen_elf.c, 356
- cpu/or32/dyngen_i386.c, 358
- cpu/or32/execute.c, 360
- cpu/or32/generate.c, 370
- cpu/or32/i386_regs.h, 373
- cpu/or32/insnset.c, 374
- cpu/or32/op.c, 387
- cpu/or32/op_1t.h, 402
- cpu/or32/op_1t_op.h, 403
- cpu/or32/op_2t.h, 404
- cpu/or32/op_2t_op.h, 405
- cpu/or32/op_3t.h, 406
- cpu/or32/op_3t_op.h, 407
- cpu/or32/op_arith_op.h, 408
- cpu/or32/op_comp_op.h, 409
- cpu/or32/op_extend_op.h, 410
- cpu/or32/op_ff1_op.h, 411
- cpu/or32/op_i386.h, 412
- cpu/or32/op_lwhb_op.h, 413
- cpu/or32/op_mac_op.h, 414
- cpu/or32/op_mftspr_op.h, 415
- cpu/or32/op_support.c, 416
- cpu/or32/op_support.h, 419
- cpu/or32/op_swhb_op.h, 421
- cpu/or32/op_t_reg_mov_op.h, 422
- cpu/or32/or32.c, 426
- cpu/or32/rec_i386.h, 433
- cpu/or32/sched_i386.h, 434
- cpu/or32/simpl32_defs.h, 435
- cpu_cfg
 - cpu-config.c, 201
- cpu_cfgr
 - cpu-config.c, 201
- cpu_clock
 - execute.c, 362
 - execute.h, 255
- cpu_dependstats
 - cpu-config.c, 201
- cpu_hazards
 - cpu-config.c, 201
- cpu_reset
 - execute.c, 363
 - execute.h, 256
- cpu_rev
 - cpu-config.c, 201
- cpu_sbuf_len
 - cpu-config.c, 201
- cpu_sr
 - cpu-config.c, 201
- cpu_state, 57
 - delay_insn, 58
 - execute.c, 368
 - execute.h, 260
 - icomplet, 58
 - insn_ea, 57
 - iqueue, 58
 - pc, 58
 - pc_delay, 58
 - pic_lines, 58
 - reg, 57
 - sprs, 57
- CPU_STATE_REG
 - i386_regs.h, 373
- cpu_superscalar
 - cpu-config.c, 202
- cpu_upr
 - cpu-config.c, 202
- cpu_ver
 - cpu-config.c, 202
- cpy_bb
 - bb.c, 439
- crc32
 - crc32.c, 648
 - crc32.h, 649
- crc32.c
 - crc32, 648
 - crc32_close, 648
 - crc32_feed_bytes, 648
 - crc32_init, 648
 - crc32_table, 648
- crc32.h
 - crc32, 649
 - crc32_close, 649
 - crc32_feed_bytes, 649
 - crc32_init, 649
- crc32_close
 - crc32.c, 648

- crc32.h, 649
- crc32_feed_bytes
 - crc32.c, 648
 - crc32.h, 649
- crc32_init
 - crc32.c, 648
 - crc32.h, 649
- crc32_table
 - crc32.c, 648
- crc_dly
 - eth_device, 110
- crc_value
 - eth_device, 110
- cs
 - mc_area, 145
- csc
 - mc, 144
- cse
 - cuc.h, 467
 - insn.c, 480
- csm
 - cuc.h, 467
 - insn.c, 480
- csm_gen
 - cuc.h, 467
 - insn.c, 480
- csr
 - dma_channel, 78
 - dma_controller, 81
 - mc, 144
- CT_NONE
 - abstract.h, 224
- CT_PHYSICAL
 - abstract.h, 224
- CT_VIRTUAL
 - abstract.h, 224
- ctrl
 - ata_host, 24
 - fb_state, 115
 - gpio_device, 121
 - vga_state, 175
- ctrl_c
 - toplevel-support.c, 827
 - toplevel-support.h, 832
- cuc
 - config, 52
 - runtime, 160
- cuc.c
 - analyse_function, 447
 - calc_cycles, 448
 - calc_size, 448
 - caller_saved, 458
 - cuc_calling_conv, 448
 - cuc_debug, 459
 - cuc_enable_bursts, 448
 - cuc_memory_order, 448
 - cuc_no_multicycle, 448
 - cuc_optimize, 449
 - cuc_timings_fn, 449
 - extract_function, 450
 - flog, 459
 - format_func_options, 450
 - func, 459
 - func_v, 459
 - gen_option, 450
 - generate_function, 452
 - main_cuc, 453
 - option_char, 459
 - options_cmd, 455
 - preunroll_bb, 455
 - print_option, 458
 - reg_cuc_sec, 458
 - set_func_deps, 458
 - tim_comp, 458
- cuc.h
 - add_data_dep, 465
 - add_dep, 466
 - add_latches, 466
 - add_memory_dep, 466
 - analyse_timings, 466
 - BB_DEAD, 465
 - BB_INLOOP, 465
 - BB_OPTIONAL, 465
 - BBID_END, 465
 - BBID_START, 465
 - build_bb, 466
 - caller_saved, 477
 - clean_deps, 467
 - cse, 467
 - csm, 467
 - csm_gen, 467
 - cuc_check, 467
 - cuc_debug, 477
 - cuc_func, 465
 - cuc_load, 467
 - CUC_MAX_STACK, 465
 - cuc_shared_list, 465
 - CUC_WIDTH_ITERATIONS, 465
 - cucdebug, 465
 - dep_list, 465
 - detect_bb, 468
 - detect_max_values, 468
 - dispose_list, 468
 - dup_func, 469
 - expand_bb, 469
 - FLAG_REG, 465
 - flog, 477
 - free_func, 469

- generate_bb_seq, 469
- insert_conditional_facts, 469
- insert_insns, 470
- INSN, 465
- insn, 477
- IT_BBEND, 465
- IT_BBSTART, 465
- IT_BRANCH, 465
- IT_COND, 465
- IT_CUT, 465
- IT_FLAG1, 465
- IT_FLAG2, 465
- IT_INDELAY, 465
- IT_LATCHED, 465
- IT_MEMADD, 465
- IT_MEMORY, 465
- IT_OUTPUT, 465
- IT_SIGNED, 465
- IT_UNUSED, 465
- IT_VOLATILE, 465
- log, 465
- LRBB_REG, 465
- main_cuc, 470
- mark_cut, 473
- MAX, 465
- MAX_BB, 465
- MAX_INSNS, 465
- MAX_PREROLL, 465
- MAX_REGS, 465
- MAX_UNROLL, 465
- MIN, 465
- MO_EXACT, 465
- MO_NONE, 465
- MO_STRONG, 465
- MO_WEAK, 465
- MT_BURST, 465
- MT_BURSTE, 465
- MT_CALL, 465
- MT_LOAD, 465
- MT_SIGNED, 465
- MT_STORE, 465
- MT_WIDTH, 465
- negate_conditional, 473
- num_insn, 477
- OPT_BB, 465
- OPT_CONST, 465
- OPT_DEST, 465
- OPT_JUMP, 465
- OPT_LRBB, 465
- OPT_NONE, 465
- OPT_REF, 465
- OPT_REGISTER, 465
- optimize_bb, 473
- optimize_cmovs, 473
- optimize_tree, 473
- preunroll_loop, 474
- print_bb_num, 474
- print_cuc_bb, 474
- print_cuc_insns, 474
- print_insns, 475
- recalc_cnts, 475
- REF, 465
- REF_BB, 465
- REF_I, 465
- reg_cuc_sec, 475
- reg_dep, 475
- reloc, 477
- remove_dead, 476
- remove_dead_bb, 476
- remove_nops, 476
- remove_trivial_regs, 476
- schedule_memory, 476
- set_io, 477
- cuc/adv.c, 436
- cuc/bb.c, 438
- cuc/cuc.c, 444
- cuc/cuc.h, 460
- cuc/insn.c, 478
- cuc/insn.h, 484
- cuc/load.c, 490
- cuc/memory.c, 495
 - add_memory_dep, 496
 - check_memory_conflict, 496
 - clean_deps, 496
 - join_transfers, 496
 - mem_ordering_cmp, 496
 - same_transfers, 496
 - schedule_memory, 496
- cuc/timings.c, 503
- cuc/verilog.c, 506
- cuc/verilog.h, 509
- cuc_bb, 59
 - cnt, 60
 - first, 60
 - insn, 60
 - last, 60
 - last_used_reg, 60
 - mdep, 60
 - next, 60
 - ninsn, 60
 - nmemory, 60
 - ntim, 60
 - prev, 60
 - selected_tim, 60
 - tim, 60
 - tmp, 60
 - type, 60
 - unrolled, 60

- cuc_calling_conv
 - cuc.c, 448
- cuc_check
 - bb.c, 439
 - cuc.h, 467
- cuc_conv, 61
 - from, 61
 - to, 61
- cuc_debug
 - cuc.c, 459
 - cuc.h, 477
- cuc_enable_bursts
 - cuc.c, 448
- cuc_func
 - cuc.h, 465
- cuc_insn, 62
 - dep, 62
 - disasm, 62
 - index, 62
 - insn, 62
 - max, 62
 - op, 62
 - opt, 62
 - tmp, 62
 - type, 62
- cuc_insn_name
 - insn.c, 480
 - insn.h, 488
- cuc_known_insn, 63
 - comutative, 63
 - name, 63
 - rtl, 63
- cuc_load
 - cuc.h, 467
 - load.c, 491
- CUC_MAX_STACK
 - cuc.h, 465
- cuc_memory_order
 - cuc.c, 448
- cuc_no_multicycle
 - cuc.c, 448
- cuc_optimize
 - cuc.c, 449
- cuc_shared_item, 64
 - cmatch, 64
 - ref, 64
- cuc_shared_list
 - cuc.h, 465
- cuc_timing_table, 65
 - delay, 65
 - delayi, 65
 - size, 65
 - sizei, 65
- cuc_timings, 66
 - b, 66
 - new_time, 66
 - nshared, 66
 - preroll, 66
 - shared, 66
 - size, 66
 - unroll, 66
- cuc_timings_fn
 - cuc.c, 449
- CUC_WIDTH_ITERATIONS
 - cuc.h, 465
- cucdebug
 - cuc.h, 465
- cum_cycles
 - func_struct, 119
- cumulative
 - profiler.c, 751
- cur_area
 - abstract.c, 221
 - abstract.h, 236
- cur_section
 - sim-config.c, 780
 - sim-config.h, 787
- cur_vadd
 - abstract.c, 221
- CURINSN
 - execute.h, 255
- curpass
 - or32.c, 431
- curr
 - gpio_device, 121
- current_descriptor
 - dma_channel, 78
- current_scan_chain
 - debug-unit.c, 518
- cut_tree
 - timings.c, 504
- cycle_duration
 - runtime, 160
- cycles
 - runtime, 160
 - stack_struct, 170
- cycles_start
 - tick.c, 816
- cylinder_high
 - ata_device, 20
- cylinder_low
 - ata_device, 20
- d_ptr
 - dynamic, 87
 - Elf64_Dyn, 96
- d_tag
 - dynamic, 87

- Elf64_Dyn, 96
- d_un
 - dynamic, 87
 - Elf64_Dyn, 96
- d_val
 - dynamic, 87
 - Elf64_Dyn, 96
- DADDR_PAGE
 - dmmu.h, 546
- daspi
 - ata_device, 20
- daspo
 - ata_device, 20
- dat
 - config_section, 56
 - sim_reset_hook, 165
 - sim_stat, 166
- data
 - channel, 35
 - jtr_read_block_response, 134
 - jtr_write_block_message, 137
- data_ci
 - abstract.c, 221
 - abstract.h, 236
- data_h
 - jtr_read_response, 136
 - jtr_write_message, 139
- data_l
 - jtr_read_response, 136
 - jtr_write_message, 139
- data_start
 - COFF_AOUTHDR, 38
- dataport_i
 - ata_device, 20
- dbuf
 - ata_device, 20
- dbuf_cnt
 - ata_device, 20
- dbuf_ptr
 - ata_device, 20
- dc
 - config, 52
 - dcache-model.c, 188
- dc_blocksize
 - dcache-model.c, 185
- dc_enabled
 - dcache-model.c, 185
- dc_info
 - dcache-model.c, 185
 - dcache-model.h, 190
- dc_inv
 - dcache-model.c, 185
 - dcache-model.h, 190
- dc_load_hitdelay
 - dcache-model.c, 185
- dc_load_missdelay
 - dcache-model.c, 185
- dc_nsets
 - dcache-model.c, 185
- dc_nways
 - dcache-model.c, 186
- dc_set, 67
 - line, 67
 - lru, 67
 - tagaddr, 67
 - way, 67
- dc_simulate_read
 - dcache-model.c, 186
 - dcache-model.h, 190
- dc_simulate_write
 - dcache-model.c, 186
 - dcache-model.h, 190
- dc_stats
 - stats.c, 275
 - stats.h, 278
- dc_store_hitdelay
 - dcache-model.c, 187
- dc_store_missdelay
 - dcache-model.c, 187
- dc_ustates
 - dcache-model.c, 187
- dcache-model.c
 - dc, 188
 - dc_blocksize, 185
 - dc_enabled, 185
 - dc_info, 185
 - dc_inv, 185
 - dc_load_hitdelay, 185
 - dc_load_missdelay, 185
 - dc_nsets, 185
 - dc_nways, 186
 - dc_simulate_read, 186
 - dc_simulate_write, 186
 - dc_store_hitdelay, 187
 - dc_store_missdelay, 187
 - dc_ustates, 187
 - reg_dc_sec, 187
- dcache-model.h
 - dc_info, 190
 - dc_inv, 190
 - dc_simulate_read, 190
 - dc_simulate_write, 190
 - MAX_DC_BLOCK_SIZE, 190
 - MAX_DC_SETS, 190
 - MAX_DC_WAYS, 190
 - MIN_DC_BLOCK_SIZE, 190
 - reg_dc_sec, 191
- dead

- `_csm_list`, 12
- `debug`
 - `config`, 52
 - `debug.c`, 790
 - `debug.h`, 793
- `debug-unit.c`
 - `calculate_watchpoints`, 512
 - `check_debug_unit`, 512
 - `current_scan_chain`, 518
 - `debug_enabled`, 512
 - `debug_gdb_enabled`, 513
 - `debug_get_mem`, 513
 - `debug_get_register`, 513
 - `debug_ignore_exception`, 514
 - `debug_server_port`, 514
 - `debug_set_chain`, 515
 - `debug_set_mem`, 515
 - `debug_set_register`, 515
 - `debug_vapi_id`, 516
 - `DECLARE_DEBUG_CHANNEL`, 516
 - `DEVELOPINT_MAX`, 512
 - `DEVELOPINT_RISCOP`, 512
 - `development`, 518
 - `development_interface_address_space`, 512
 - `du_reset`, 516
 - `get_devint_reg`, 516, 517
 - `in_reset`, 518
 - `reg_debug_sec`, 517
 - `RISCOP_RESET`, 511
 - `RISCOP_STALL`, 511
 - `set_devint_reg`, 517, 518
 - `set_stall_state`, 518
- `debug-unit.h`
 - `check_debug_unit`, 520
 - `debug_get_register`, 520
 - `debug_ignore_exception`, 521
 - `debug_scan_chain_ids`, 519
 - `debug_set_chain`, 521
 - `debug_set_register`, 521
 - `debug_unit_action`, 519
 - `DebugInstructionFetch`, 520
 - `DebugLoadAddress`, 520
 - `DebugLoadData`, 520
 - `DebugStoreAddress`, 520
 - `DebugStoreData`, 520
 - `du_reset`, 522
 - `JTAG_CHAIN_DEBUG_UNIT`, 519
 - `JTAG_CHAIN_DEVELOPMENT`, 519
 - `JTAG_CHAIN_GLOBAL`, 519
 - `JTAG_CHAIN_TRACE`, 519
 - `JTAG_CHAIN_WISHBONE`, 519
 - `reg_debug_sec`, 522
 - `set_stall_state`, 523
- `debug.c`
 - `__ORSIM_NO_DEC_DBCH`, 790
 - `__orsim_dbchs`, 790
 - `debug`, 790
 - `debug_classes`, 790
 - `DECLARE_DEBUG_CHANNEL`, 790
 - `orsim_dbcl_set`, 790
 - `orsim_dbcl_set_name`, 790
 - `orsim_dbg_log`, 790
 - `parse_dbchs`, 790
- `debug.h`
 - `__ORSIM_DBCL_ERR`, 793
 - `__ORSIM_DBCL_FIXME`, 793
 - `__ORSIM_DBCL_TRACE`, 793
 - `__ORSIM_DBCL_WARN`, 793
 - `__ORSIM_DBG_USE_FUNC`, 792
 - `__ORSIM_DEBUG_CLASS`, 793
 - `__ORSIM_DEBUG_LOG`, 792
 - `__ORSIM_DPRINTF`, 792
 - `__ORSIM_GET_DEBUGGING`, 792
 - `__ORSIM_GET_DEBUGGING_ERR`, 792
 - `__ORSIM_GET_DEBUGGING_FIXME`, 792
 - `__ORSIM_GET_DEBUGGING_TRACE`, 792
 - `__ORSIM_GET_DEBUGGING_WARN`, 792
 - `debug`, 793
 - `DECLARE_DEBUG_CHANNEL`, 792
 - `DEFAULT_DEBUG_CHANNEL`, 792
 - `ERR`, 792
 - `ERR_`, 793
 - `ERR_ON`, 793
 - `FIXME`, 793
 - `FIXME_`, 793
 - `FIXME_ON`, 793
 - `orsim_dbcl_set_name`, 793
 - `orsim_dbg_log`, 794
 - `parse_dbchs`, 794
 - `TRACE`, 793
 - `TRACE_`, 793
 - `TRACE_ON`, 793
 - `WARN`, 793
 - `WARN_`, 793
 - `WARN_ON`, 793
- `debug/debug-unit.c`, 510
- `debug/debug-unit.h`, 519
- `debug/gdb.h`, 524
- `debug/gdbcomm.c`, 526
- `debug/gdbcomm.h`, 530
- `debug_classes`
 - `debug.c`, 790
- `debug_enabled`
 - `debug-unit.c`, 512
- `debug_gdb_enabled`
 - `debug-unit.c`, 513
- `debug_get_mem`
 - `debug-unit.c`, 513

- debug_get_register
 - debug-unit.c, 513
 - debug-unit.h, 520
- debug_ignore_exception
 - debug-unit.c, 514
 - debug-unit.h, 521
- debug_scan_chain_ids
 - debug-unit.h, 519
- debug_server_port
 - debug-unit.c, 514
- debug_set_chain
 - debug-unit.c, 515
 - debug-unit.h, 521
- debug_set_mem
 - debug-unit.c, 515
- debug_set_register
 - debug-unit.c, 515
 - debug-unit.h, 521
- debug_unit_action
 - debug-unit.h, 519
- debug_vapi_id
 - debug-unit.c, 516
- DebugInstructionFetch
 - debug-unit.h, 520
- DebugLoadAddress
 - debug-unit.h, 520
- DebugLoadData
 - debug-unit.h, 520
- DebugStoreAddress
 - debug-unit.h, 520
- DebugStoreData
 - debug-unit.h, 520
- DECLARE_DEBUG_CHANNEL
 - debug-unit.c, 516
 - debug.c, 790
 - debug.h, 792
 - sched.c, 806
 - sprs.c, 315
- decode_execute
 - execute.c, 363
- decode_execute_wrapper
 - execute.c, 363
- DEF_1T_OP
 - def_op_t.h, 320
 - dyn_rec.c, 330
- DEF_2T_OP
 - def_op_t.h, 320
 - dyn_rec.c, 330
- DEF_2T_OP_NEQ
 - def_op_t.h, 320
 - dyn_rec.c, 330
- DEF_3T_OP
 - def_op_t.h, 320
 - dyn_rec.c, 330
- DEF_3T_OP_NEQ
 - def_op_t.h, 321
 - dyn_rec.c, 330
- DEF_GPR_OP
 - def_op_t.h, 321
 - dyn_rec.c, 330
- def_op_t.h
 - DEF_1T_OP, 320
 - DEF_2T_OP, 320
 - DEF_2T_OP_NEQ, 320
 - DEF_3T_OP, 320
 - DEF_3T_OP_NEQ, 321
 - DEF_GPR_OP, 321
 - GPR_T, 321
- DEFAULT_BAUD
 - tty.c, 643
- DEFAULT_DEBUG_CHANNEL
 - 16450.c, 573
 - atadevice.c, 597
 - atadevice_cmdi.c, 607
 - atahost.c, 619
 - debug.h, 792
 - dma.c, 659
 - dmmu.c, 540
 - eth.c, 673
 - except.c, 286
 - gpio.c, 704
 - immu.c, 549
 - mc.c, 714
 - pic.c, 738
 - sched.c, 806
 - sim-config.c, 774
 - simprintf.c, 810
 - spr-dump.c, 302
 - sprs.c, 315
 - tick.c, 814
 - vapi.c, 840
- DEFAULT_MEMORY_LEN
 - abstract.h, 224
- DEFAULT_MEMORY_START
 - abstract.h, 224
- DEFAULT_TTY_DEVICE
 - tty.c, 643
- delay
 - cuc_timing_table, 65
- delay_insn
 - cpu_state, 58
- delayi
 - cuc_timing_table, 65
- delayr
 - dyn_page, 86
 - mem_config, 147
 - mem_ops, 149
- delayw

- mem_config, 147
- mem_ops, 149
- dep
 - cuc_insn, 62
- dep_list
 - cuc.h, 465
- depend
 - dstats_entry, 85
 - fstats_entry, 118
- depend_operands
 - execute.h, 256
- dependstats
 - config, 52
- desc
 - dma_channel, 78
- destination
 - dma_channel, 78
- destination_mask
 - dma_channel, 78
- destruct_automata
 - or32.c, 428
- detect_bb
 - bb.c, 439
 - cuc.h, 468
- detect_locals
 - load.c, 492
- detect_max_values
 - adv.c, 437
 - cuc.h, 468
- dev
 - ata_device, 20
- dev_generic
 - GENERIC_BYTE, 73
 - GENERIC_HW, 73
 - GENERIC_READ, 73
 - GENERIC_WORD, 73
 - GENERIC_WRITE, 73
- dev_16450, 68
 - baseaddr, 71
 - channel, 71
 - channel_str, 71
 - char_clks, 71
 - dlh, 71
 - dll, 71
 - enabled, 71
 - fcr, 71
 - fifo_len, 71
 - ier, 71
 - iir, 71
 - ints, 71
 - iregs, 71
 - irq, 71
 - istat, 71
 - jitter, 71
 - lcr, 71
 - loopback, 71
 - lsr, 71
 - mcr, 71
 - msr, 71
 - receiveing, 71
 - rcv_break, 71
 - regs, 71
 - rxbuf, 71
 - rxbuf_full, 71
 - rxbuf_head, 71
 - rxbuf_tail, 71
 - rxser, 71
 - scr, 71
 - skew, 71
 - txbuf, 71
 - txbuf_full, 71
 - txbuf_head, 71
 - txbuf_tail, 71
 - txser, 71
 - uart16550, 71
 - vapi, 71
 - vapi_buf, 71
 - vapi_buf_head_ptr, 71
 - vapi_buf_tail_ptr, 71
 - vapi_id, 71
- dev_generic, 73
 - baseaddr, 74
 - byte_enabled, 74
 - enabled, 74
 - hw_enabled, 74
 - name, 74
 - size, 74
 - trans_direction, 74
 - trans_size, 74
 - value, 74
 - word_enabled, 74
- dev_id
 - ata_host, 24
- dev_list
 - abstract.c, 221
- dev_memarea, 75
 - addr_compare, 76
 - addr_mask, 76
 - direct_ops, 76
 - log, 76
 - next, 76
 - ops, 76
 - size, 76
 - size_mask, 76
 - valid, 76
- dev_sel
 - ata_host, 24
- DEVELOPINT_MAX

- debug-unit.c, 512
- DEVELOPINT_RISCOPE
 - debug-unit.c, 512
- development
 - debug-unit.c, 518
- development_interface_address_space
 - debug-unit.c, 512
- device
 - ata_devices, 22
- device_control
 - ata_device, 20
- device_head
 - ata_device, 20
- DEVICE_RESET
 - atacmd.h, 593
- devices
 - ata_host, 24
- direct_ops
 - dev_memarea, 76
- dirty
 - dyn_page, 86
- dirtyfy_page
 - dyn_rec.c, 330
- DISABLE_ADVANCED_POWER_-
MANAGEMENT
 - atacmd.h, 593
- DISABLE_MEDIA_STATUS_NOTIFICATION
 - atacmd.h, 593
- DISABLE_POWERUP_IN_STANDBY_-
FEATURE_SET
 - atacmd.h, 593
- DISABLE_READ_LOOKAHEAD
 - atacmd.h, 593
- DISABLE_RELEASE_INTERRUPT
 - atacmd.h, 593
- DISABLE_REVERTING_TO_POWERON_-
DEFAULTS
 - atacmd.h, 593
- DISABLE_SERVICE_INTERRUPT
 - atacmd.h, 593
- DISABLE_WRITE_CACHE
 - atacmd.h, 593
- disasm
 - cuc_insn, 62
- disassemble_index
 - or32.c, 429
- disassemble_insn
 - or32.c, 429
- disassemble_memory
 - abstract.c, 207
 - abstract.h, 224
- disassembled
 - or32.c, 431
- disassembled_str
 - or32.c, 431
- dispose_list
 - cuc.h, 468
 - insn.c, 481
- DISWIDTH
 - dumpverilog.c, 796
- dlh
 - dev_16450, 71
- dll
 - dev_16450, 71
- dma
 - eth_device, 110
- dma-defs.h
 - DMA_ADDR_SPACE, 652
 - DMA_CH_A0, 652
 - DMA_CH_A0_ADDR_OFFSET, 652
 - DMA_CH_A0_ADDR_WIDTH, 652
 - DMA_CH_A1, 652
 - DMA_CH_A1_ADDR_OFFSET, 652
 - DMA_CH_A1_ADDR_WIDTH, 652
 - DMA_CH_AM0, 652
 - DMA_CH_AM0_MASK_OFFSET, 652
 - DMA_CH_AM0_MASK_WIDTH, 652
 - DMA_CH_AM1, 652
 - DMA_CH_AM1_MASK_OFFSET, 652
 - DMA_CH_AM1_MASK_WIDTH, 652
 - DMA_CH_BASE, 652
 - DMA_CH_CSR, 652
 - DMA_CH_CSR_ARS_OFFSET, 654
 - DMA_CH_CSR_BUSY_OFFSET, 654
 - DMA_CH_CSR_CH_EN_OFFSET, 654
 - DMA_CH_CSR_DONE_OFFSET, 654
 - DMA_CH_CSR_DST_SEL_OFFSET, 654
 - DMA_CH_CSR_ERR_OFFSET, 654
 - DMA_CH_CSR_INC_DST_OFFSET, 654
 - DMA_CH_CSR_INC_SRC_OFFSET, 654
 - DMA_CH_CSR_INE_CHK_DONE_-
OFFSET, 654
 - DMA_CH_CSR_INE_DONE_OFFSET, 654
 - DMA_CH_CSR_INE_ERR_OFFSET, 654
 - DMA_CH_CSR_INT_CHUNK_DONE_-
OFFSET, 654
 - DMA_CH_CSR_INT_DONE_OFFSET, 654
 - DMA_CH_CSR_INT_ERR_OFFSET, 654
 - DMA_CH_CSR_MODE_OFFSET, 654
 - DMA_CH_CSR_PRIORITY_OFFSET, 654
 - DMA_CH_CSR_PRIORITY_WIDTH, 654
 - DMA_CH_CSR_RESERVED_OFFSET, 654
 - DMA_CH_CSR_RESERVED_WIDTH, 654
 - DMA_CH_CSR_REST_EN_OFFSET, 654
 - DMA_CH_CSR_SRC_SEL_OFFSET, 654
 - DMA_CH_CSR_STOP_OFFSET, 654
 - DMA_CH_CSR_SZ_WB_OFFSET, 654
 - DMA_CH_CSR_USE_ED_OFFSET, 654

- DMA_CH_CSR_WRITE_MASK, 654
- DMA_CH_DESC, 654
- DMA_CH_DESC_ADDR_OFFSET, 654
- DMA_CH_DESC_ADDR_WIDTH, 654
- DMA_CH_SIZE, 654
- DMA_CH_SWPTR, 654
- DMA_CH_SWPTR_EN_OFFSET, 656
- DMA_CH_SWPTR_PTR_OFFSET, 656
- DMA_CH_SWPTR_PTR_WIDTH, 656
- DMA_CH_SZ, 656
- DMA_CH_SZ_CHK_SZ_OFFSET, 656
- DMA_CH_SZ_CHK_SZ_WIDTH, 656
- DMA_CH_SZ_TOT_SZ_OFFSET, 656
- DMA_CH_SZ_TOT_SZ_WIDTH, 656
- DMA_CSR, 656
- DMA_CSR_PAUSE_OFFSET, 656
- DMA_DESC_ADR0, 656
- DMA_DESC_ADR1, 656
- DMA_DESC_CSR, 656
- DMA_DESC_CSR_DST_SEL_OFFSET, 656
- DMA_DESC_CSR_EOL_OFFSET, 656
- DMA_DESC_CSR_INC_DST_OFFSET, 656
- DMA_DESC_CSR_INC_SRC_OFFSET, 656
- DMA_DESC_CSR_SRC_SEL_OFFSET, 656
- DMA_DESC_CSR_TOT_SZ_OFFSET, 656
- DMA_DESC_CSR_TOT_SZ_WIDTH, 656
- DMA_DESC_NEXT, 656
- DMA_INT_MSK_A, 656
- DMA_INT_MSK_B, 656
- DMA_INT_SRC_A, 656
- DMA_INT_SRC_B, 656
- DMA_NUM_CHANNELS, 656
- dma.c
 - CHANNEL_ND_I, 659
 - check_dma_ack_o, 659
 - clear_dma_nd_i, 659
 - clear_dma_req_i, 659
 - DEFAULT_DEBUG_CHANNEL, 659
 - dma_baseaddr, 659
 - dma_channel_clock, 659
 - dma_channel_terminate_transfer, 659
 - dma_enabled, 660
 - dma_init_transfer, 660
 - dma_irq, 660
 - dma_load_descriptor, 660
 - dma_read32, 660
 - dma_read_ch_csr, 661
 - dma_reset, 661
 - dma_sec_end, 661
 - dma_sec_start, 661
 - dma_status, 661
 - dma_vapi_id, 662
 - dma_write32, 662
 - dma_write_ch_csr, 662
- dma.h
 - check_dma_ack_o, 665
 - clear_dma_nd_i, 665
 - clear_dma_req_i, 665
 - find_dma_controller_ch, 665
 - reg_dma_sec, 665
 - set_dma_nd_i, 665
 - set_dma_req_i, 665
- dma_ack_o
 - dma_channel, 78
- DMA_ADDR_SPACE
 - dma-defs.h, 652
- dma_baseaddr
 - dma.c, 659
- DMA_CH_A0
 - dma-defs.h, 652
- DMA_CH_A0_ADDR_OFFSET
 - dma-defs.h, 652
- DMA_CH_A0_ADDR_WIDTH
 - dma-defs.h, 652
- DMA_CH_A1
 - dma-defs.h, 652
- DMA_CH_A1_ADDR_OFFSET
 - dma-defs.h, 652
- DMA_CH_A1_ADDR_WIDTH
 - dma-defs.h, 652
- DMA_CH_AM0
 - dma-defs.h, 652
- DMA_CH_AM0_MASK_OFFSET
 - dma-defs.h, 652
- DMA_CH_AM0_MASK_WIDTH
 - dma-defs.h, 652
- DMA_CH_AM1
 - dma-defs.h, 652
- DMA_CH_AM1_MASK_OFFSET
 - dma-defs.h, 652
- DMA_CH_AM1_MASK_WIDTH
 - dma-defs.h, 652
- DMA_CH_BASE
 - dma-defs.h, 652
- DMA_CH_CSR
 - dma-defs.h, 652
- DMA_CH_CSR_ARS_OFFSET
 - dma-defs.h, 654
- DMA_CH_CSR_BUSY_OFFSET
 - dma-defs.h, 654
- DMA_CH_CSR_CH_EN_OFFSET
 - dma-defs.h, 654
- dmass, 663
- find_dma_controller_ch, 662
- masked_increase, 663
- reg_dma_sec, 663
- set_dma_nd_i, 663
- set_dma_req_i, 663

- DMA_CH_CSR_DONE_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_DST_SEL_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_ERR_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_INC_DST_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_INC_SRC_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_INE_CHK_DONE_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_INE_DONE_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_INE_ERR_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_INT_CHUNK_DONE_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_INT_DONE_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_INT_ERR_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_MODE_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_PRIORITY_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_PRIORITY_WIDTH
dma-defs.h, 654
- DMA_CH_CSR_RESERVED_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_RESERVED_WIDTH
dma-defs.h, 654
- DMA_CH_CSR_REST_EN_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_SRC_SEL_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_STOP_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_SZ_WB_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_USE_ED_OFFSET
dma-defs.h, 654
- DMA_CH_CSR_WRITE_MASK
dma-defs.h, 654
- DMA_CH_DESC
dma-defs.h, 654
- DMA_CH_DESC_ADDR_OFFSET
dma-defs.h, 654
- DMA_CH_DESC_ADDR_WIDTH
dma-defs.h, 654
- DMA_CH_SIZE
dma-defs.h, 654
- DMA_CH_SWPTR
dma-defs.h, 654
- DMA_CH_SWPTR_EN_OFFSET
dma-defs.h, 656
- DMA_CH_SWPTR_PTR_OFFSET
dma-defs.h, 656
- DMA_CH_SWPTR_PTR_WIDTH
dma-defs.h, 656
- DMA_CH_SZ
dma-defs.h, 656
- DMA_CH_SZ_CHK_SZ_OFFSET
dma-defs.h, 656
- DMA_CH_SZ_CHK_SZ_WIDTH
dma-defs.h, 656
- DMA_CH_SZ_TOT_SZ_OFFSET
dma-defs.h, 656
- DMA_CH_SZ_TOT_SZ_WIDTH
dma-defs.h, 656
- dma_channel, 77
 - a0, 78
 - a1, 78
 - am0, 78
 - am1, 78
 - channel_mask, 78
 - channel_number, 78
 - chunk_size, 78
 - controller, 78
 - csr, 78
 - current_descriptor, 78
 - desc, 78
 - destination, 78
 - destination_mask, 78
 - dma_ack_o, 78
 - dma_nd_i, 78
 - dma_req_i, 78
 - load_next_descriptor_when_done, 78
 - referenced, 78
 - regs, 78
 - source, 78
 - source_mask, 78
 - swptr, 78
 - sz, 78
 - total_size, 78
 - words_transferred, 78
- dma_channel_clock
dma.c, 659
- dma_channel_terminate_transfer
dma.c, 659
- dma_controller, 80
 - baseaddr, 81
 - ch, 81
 - csr, 81
 - enabled, 81
 - int_msk_a, 81
 - int_msk_b, 81
 - int_src_a, 81

- int_src_b, 81
- irq, 81
- next, 81
- regs, 81
- vapi_id, 81
- DMA_CSR
 - dma-defs.h, 656
- DMA_CSR_PAUSE_OFFSET
 - dma-defs.h, 656
- DMA_DESC_ADR0
 - dma-defs.h, 656
- DMA_DESC_ADR1
 - dma-defs.h, 656
- DMA_DESC_CSR
 - dma-defs.h, 656
- DMA_DESC_CSR_DST_SEL_OFFSET
 - dma-defs.h, 656
- DMA_DESC_CSR_EOL_OFFSET
 - dma-defs.h, 656
- DMA_DESC_CSR_INC_DST_OFFSET
 - dma-defs.h, 656
- DMA_DESC_CSR_INC_SRC_OFFSET
 - dma-defs.h, 656
- DMA_DESC_CSR_SRC_SEL_OFFSET
 - dma-defs.h, 656
- DMA_DESC_CSR_TOT_SZ_OFFSET
 - dma-defs.h, 656
- DMA_DESC_CSR_TOT_SZ_WIDTH
 - dma-defs.h, 656
- DMA_DESC_NEXT
 - dma-defs.h, 656
- dma_enabled
 - dma.c, 660
- dma_init_transfer
 - dma.c, 660
- DMA_INT_MSK_A
 - dma-defs.h, 656
- DMA_INT_MSK_B
 - dma-defs.h, 656
- DMA_INT_SRC_A
 - dma-defs.h, 656
- DMA_INT_SRC_B
 - dma-defs.h, 656
- dma_irq
 - dma.c, 660
- dma_load_descriptor
 - dma.c, 660
- dma_mode
 - ata_device, 20
- DMA_MODE0_TD
 - atahost.c, 615
 - atahost_define.h, 629
- dma_mode0_td
 - ata_host, 24
- DMA_MODE0_TEOC
 - atahost.c, 615
 - atahost_define.h, 629
- dma_mode0_tec
 - ata_host, 24
- DMA_MODE0_TM
 - atahost.c, 615
 - atahost_define.h, 629
- dma_mode0_tm
 - ata_host, 24
- dma_nd_i
 - dma_channel, 78
- DMA_NUM_CHANNELS
 - dma-defs.h, 656
- dma_read32
 - dma.c, 660
- dma_read_ch_csr
 - dma.c, 661
- dma_req_i
 - dma_channel, 78
- dma_reset
 - dma.c, 661
- dma_sec_end
 - dma.c, 661
- dma_sec_start
 - dma.c, 661
- dma_status
 - dma.c, 661
- dma_vapi_id
 - dma.c, 662
- dma_write32
 - dma.c, 662
- dma_write_ch_csr
 - dma.c, 662
- dmarq
 - ata_device, 20
- dmass
 - dma.c, 663
- dmmu, 82
 - enabled, 83
 - entrysize, 83
 - hitdelay, 83
 - lru_reload, 83
 - missdelay, 83
 - nsets, 83
 - nways, 83
 - page_mask, 83
 - page_offset_mask, 83
 - pagesize, 83
 - pagesize_log2, 83
 - set_mask, 83
 - ustates, 83
 - vpn_mask, 83
- dmmu.c

- DEFAULT_DEBUG_CHANNEL, 540
- dmmu_enabled, 540
- dmmu_end_sec, 540
- dmmu_entsize, 540
- dmmu_find_tlbmr, 541
- dmmu_hitdelay, 541
- dmmu_missdelay, 541
- dmmu_nsets, 541
- dmmu_nways, 541
- dmmu_pagesize, 541
- dmmu_start_sec, 542
- dmmu_state, 544
- dmmu_translate, 542
- dmmu_ustates, 542
- dtlb_status, 543
- peek_into_dtlb, 543
- reg_dmmu_sec, 543
- dmmu.h
 - DADDR_PAGE, 546
 - dmmu_simulate_tlb, 546
 - dmmu_state, 547
 - dmmu_translate, 546
 - peek_into_dtlb, 546
 - reg_dmmu_sec, 546
- dmmu_enabled
 - dmmu.c, 540
- dmmu_end_sec
 - dmmu.c, 540
- dmmu_entsize
 - dmmu.c, 540
- dmmu_find_tlbmr
 - dmmu.c, 541
- dmmu_hitdelay
 - dmmu.c, 541
- dmmu_missdelay
 - dmmu.c, 541
- dmmu_nsets
 - dmmu.c, 541
- dmmu_nways
 - dmmu.c, 541
- dmmu_pagesize
 - dmmu.c, 541
- dmmu_simulate_tlb
 - dmmu.h, 546
- dmmu_start_sec
 - dmmu.c, 542
- dmmu_state
 - dmmu.c, 544
 - dmmu.h, 547
- dmmu_stats
 - stats.c, 275
 - stats.h, 278
- dmmu_translate
 - dmmu.c, 542
 - dmmu.h, 546
- dmmu_ustates
 - dmmu.c, 542
- dmmustats_entry, 84
 - loads_pagefaults, 84
 - loads_tlbhit, 84
 - loads_tlbmiss, 84
 - stores_pagefaults, 84
 - stores_tlbhit, 84
 - stores_tlbmiss, 84
- do_jump
 - op_support.c, 417
 - op_support.h, 419
- do_sched_wrap
 - op.c, 394
- do_sched_wrap_delay
 - op.c, 395
- do_scheduler
 - sched.c, 806
 - sched.h, 808
- do_stats
 - dyn_rec.c, 346
 - execute.c, 368
 - execute.h, 260
 - sim-config.h, 787
- done_memory_table
 - abstract.c, 207
 - abstract.h, 225
- DOWNLOAD_MICROCODE
 - atacmd.h, 593
- dsiz
 - COFF_AOUTHDR, 38
- dstats
 - stats.c, 275
- dstats_entry, 85
 - cnt_dynamic, 85
 - depend, 85
 - insn1, 85
 - insn2, 85
- DSTATS_LEN
 - stats.c, 275
- DT_DEBUG
 - elf.h, 250
- DT_FINI
 - elf.h, 250
- DT_HASH
 - elf.h, 250
- DT_HIPROC
 - elf.h, 250
- DT_INIT
 - elf.h, 250
- DT_JMPREL
 - elf.h, 250
- DT_LOPROC

- elf.h, 250
- DT_NEEDED
 - elf.h, 250
- DT_NULL
 - elf.h, 250
- DT_PLTGOT
 - elf.h, 250
- DT_PLTREL
 - elf.h, 250
- DT_PLTRELSZ
 - elf.h, 250
- DT_REL
 - elf.h, 250
- DT_RELA
 - elf.h, 250
- DT_RELAENT
 - elf.h, 250
- DT_RELASZ
 - elf.h, 250
- DT_RELENT
 - elf.h, 250
- DT_RELSZ
 - elf.h, 250
- DT_RPATH
 - elf.h, 250
- DT_SONAME
 - elf.h, 250
- DT_STRSZ
 - elf.h, 250
- DT_STARTAB
 - elf.h, 250
- DT_SYMBOLIC
 - elf.h, 250
- DT_SYMENT
 - elf.h, 250
- DT_SYMTAB
 - elf.h, 250
- DT_TEXTREL
 - elf.h, 250
- dtlb_status
 - dmmu.c, 543
- dtr0
 - ata_host, 24
- dtr1
 - ata_host, 24
- du_reset
 - debug-unit.c, 516
 - debug-unit.h, 522
- dump_exe_log
 - execute.c, 364
 - execute.h, 256
- dump_memory
 - abstract.c, 207
 - abstract.h, 225
- dump_spr
 - spr-dump.c, 302
 - spr-dump.h, 313
 - sprs.h, 317
- dumphex
 - dumpverilog.c, 797
 - dumpverilog.h, 799
- dumpreg
 - execute.c, 364
 - execute.h, 257
- dumpverilog
 - dumpverilog.c, 797
 - dumpverilog.h, 799
- dumpverilog.c
 - DISWIDTH, 796
 - dumphex, 797
 - dumpverilog, 797
 - DW, 796
 - DWQ, 796
 - LABELEND_CHAR, 796
 - OR1K_MEM_VERILOG_FOOTER, 796
 - OR1K_MEM_VERILOG_HEADER, 796
- dumpverilog.h
 - dumphex, 799
 - dumpverilog, 799
- dup_func
 - bb.c, 440
 - cuc.h, 469
- DW
 - dumpverilog.c, 796
- DWQ
 - dumpverilog.c, 796
- dyn32_defs.h
 - l_none, 322
 - PARAMS, 322
- dyn_checkwrite
 - dyn_rec.c, 331
 - dyn_rec.h, 348
- dyn_main
 - dyn_rec.c, 331
 - dyn_rec.h, 348
- dyn_page, 86
 - delayr, 86
 - dirty, 86
 - host_len, 86
 - host_page, 86
 - insn_indexs, 86
 - insns, 86
 - locs, 86
 - or_page, 86
 - ts_bound, 86
- dyn_rec.c
 - __op_param1, 346
 - __op_param2, 346

- __op_param3, 346
- add_to_op_params, 330
- add_to_opq, 330
- DEF_1T_OP, 330
- DEF_2T_OP, 330
- DEF_2T_OP_NEQ, 330
- DEF_3T_OP, 330
- DEF_3T_OP_NEQ, 330
- DEF_GPR_OP, 330
- dirtyfy_page, 330
- do_stats, 346
- dyn_checkwrite, 331
- dyn_main, 331
- dyn_ret_stack_prot, 331
- dyn_sigsegv_debug, 332
- enough_host_page, 332
- eval_insn_ops, 332
- find_jump_loc, 332
- find_t, 332
- gen_j_imm, 333
- gen_j_reg, 333
- gen_l_add, 333
- gen_l_addc, 333
- gen_l_and, 333
- gen_l_bf, 333
- gen_l_bnf, 333
- gen_l_cmov, 334
- gen_l_cust1, 335
- gen_l_cust2, 335
- gen_l_cust3, 335
- gen_l_cust4, 335
- gen_l_cust5, 335
- gen_l_cust6, 335
- gen_l_cust7, 335
- gen_l_cust8, 335
- gen_l_div, 335
- gen_l_divu, 335
- gen_l_extbs, 335
- gen_l_extbz, 335
- gen_l_extbs, 335
- gen_l_exthz, 335
- gen_l_extws, 335
- gen_l_extwz, 335
- gen_l_ffl, 335
- gen_l_invalid, 335
- gen_l_j, 335
- gen_l_jal, 335
- gen_l_jalr, 336
- gen_l_jr, 336
- gen_l_lbs, 336
- gen_l_lbz, 337
- gen_l_lhs, 337
- gen_l_lhz, 337
- gen_l_lws, 337
- gen_l_lwz, 337
- gen_l_mac, 337
- gen_l_macrc, 338
- gen_l_mfspr, 338
- gen_l_movhi, 338
- gen_l_msb, 338
- gen_l_mtspr, 338
- gen_l_mul, 338
- gen_l_mulu, 338
- gen_l_nop, 338
- gen_l_or, 338
- gen_l_rfe, 338
- gen_l_sb, 338
- gen_l_sfeq, 338
- gen_l_sfges, 339
- gen_l_sfgeu, 339
- gen_l_sfgts, 339
- gen_l_sfgtu, 339
- gen_l_sfles, 339
- gen_l_sfleu, 339
- gen_l_sflts, 339
- gen_l_sfltu, 339
- gen_l_sfne, 339
- gen_l_sh, 339
- gen_l_sll, 339
- gen_l_sra, 339
- gen_l_srl, 339
- gen_l_sub, 339
- gen_l_sw, 339
- gen_l_sys, 339
- gen_l_trap, 340
- gen_l_xor, 340
- gen_lf_add_s, 340
- gen_lf_div_s, 340
- gen_lf_ftoi_s, 340
- gen_lf_itof_s, 340
- gen_lf_madd_s, 340
- gen_lf_mul_s, 340
- gen_lf_rem_s, 341
- gen_lf_sfeq_s, 341
- gen_lf_sfge_s, 341
- gen_lf_sfgt_s, 341
- gen_lf_sfle_s, 341
- gen_lf_sflt_s, 341
- gen_lf_sfne_s, 342
- gen_lf_sub_s, 342
- gen_op_mark_loc, 342
- generic_gen_op, 330
- imm_gen_op, 330
- immu_retranslate, 342
- init_dyn_recomp, 343
- new_dp, 343
- OPS_ENLARGE_BY, 330
- RECED_PAGE_ENLARGE_BY, 330

- recheck_immu, 344
- recompile_delay_insn, 344
- recompile_insn, 344
- recompile_page, 344
- run_sched_out_of_line, 345
- set_pc_delay_gpr, 346
- ship_gprs_out_t, 345
- ship_t_out, 346
- sigsegv_addr, 346
- sigsegv_state, 346
- T_NONE, 330
- TFLAG_DST, 330
- TFLAG_SAVED, 330
- TFLAG_SOURCED, 330
- TFLAG_SRC, 330
- dyn_rec.h
 - add_to_op_params, 348
 - add_to_opq, 348
 - dyn_checkwrite, 348
 - dyn_main, 348
 - enough_host_page, 349
 - enter_dyn_code, 349
 - gen_code_ent, 348
 - glue, 348
 - IMMU_GOT_DISABLED, 348
 - IMMU_GOT_ENABLED, 348
 - init_dyn_recomp, 349
 - new_dp, 349
 - rec_stack_base, 351
 - recheck_immu, 350
 - recompile_page, 350
 - run_sched_out_of_line, 350
 - xglue, 348
- dyn_ret_stack_prot
 - dyn_rec.c, 331
- dyn_sigsegv_debug
 - dyn_rec.c, 332
- dynamic, 87
 - d_ptr, 87
 - d_tag, 87
 - d_un, 87
 - d_val, 87
- dyngen.c
 - c_file_head, 353
 - c_rel_file_head, 353
 - c_rel_file_tail, 353
 - c_sw_file_head, 353
 - c_sw_file_tail, 354
 - gen_code_proto, 354
 - gen_func_proto, 353
 - main, 353
 - MAX_PARAMS, 353
 - OP_FUNC_PARAM_PREFIX, 353
 - OP_FUNC_PREFIX, 353
- dyngen.h
 - archfs, 355
 - bffs, 355
- dyngen_elf.c
 - bffs, 357
 - elf_close_obj, 356
 - elf_find_func, 356
 - elf_get_func_len, 356
 - elf_get_func_name, 357
 - elf_get_func_reloc, 357
 - elf_get_func_start, 357
 - elf_get_sym_name, 357
 - elf_open_obj, 357
- dyngen_i386.c
 - archfs, 358
 - i386_gen_func_reloc, 358
 - i386_gen_reloc, 358
 - i386_get_real_func_len, 358
 - RET_OPCODE, 358
- e
 - COFF_syment, 48
- E_DIMNUM
 - coff.h, 242
- e_ehsize
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_entry
 - elf32_hdr, 89
 - elf64_hdr, 98
- E_FILNMLEN
 - coff.h, 242
- e_flags
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_hdr
 - elf_obj, 106
- e_ident
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_machine
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_name
 - COFF_syment, 48
- e_numaux
 - COFF_syment, 48
- e_offset
 - COFF_syment, 48
- e_phentsize
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_phnum
 - elf32_hdr, 89

- elf64_hdr, 98
- e_phoff
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_rel_num
 - elf_obj, 106
- e_rel_sec
 - elf_obj, 106
- e_rel_sym
 - elf_obj, 106
- e_rela_num
 - elf_obj, 106
- e_rela_sec
 - elf_obj, 106
- e_rela_sym
 - elf_obj, 106
- e_relas
 - elf_obj, 106
- e_rels
 - elf_obj, 106
- e_sclass
 - COFF_syment, 48
- e_scnnum
 - COFF_syment, 48
- e_sections
 - elf_obj, 106
- e_shdrs
 - elf_obj, 106
- e_shentsize
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_shnum
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_shoff
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_shstrndx
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_sym_num
 - elf_obj, 106
- e_sym_str_tab
 - elf_obj, 106
- E_SYMNMLEN
 - coff.h, 242
- e_syms
 - elf_obj, 106
- e_type
 - COFF_syment, 48
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_value
 - COFF_syment, 48
- e_version
 - elf32_hdr, 89
 - elf64_hdr, 98
- e_zeroes
 - COFF_syment, 48
- edge_trigger
 - config::pic, 54
- EF
 - or32.c, 428
- EFI
 - or32.c, 428
- EFN
 - or32.c, 428
- EI_CLASS
 - elf.h, 250
- EI_DATA
 - elf.h, 250
- EI_MAG0
 - elf.h, 250
- EI_MAG1
 - elf.h, 250
- EI_MAG2
 - elf.h, 250
- EI_MAG3
 - elf.h, 250
- EI_NIDENT
 - elf.h, 250
- EI_PAD
 - elf.h, 250
- EI_VERSION
 - elf.h, 250
- elf.h
 - _DYNAMIC, 253
 - AT_BASE, 250
 - AT_EGID, 250
 - AT_ENTRY, 250
 - AT_EUID, 250
 - AT_EXECFD, 250
 - AT_FLAGS, 250
 - AT_GID, 250
 - AT_IGNORE, 250
 - AT_NOTELF, 250
 - AT_NULL, 250
 - AT_PAGESZ, 250
 - AT_PHDR, 250
 - AT_PHEXT, 250
 - AT_PHNUM, 250
 - AT_UID, 250
 - DT_DEBUG, 250
 - DT_FINI, 250
 - DT_HASH, 250
 - DT_HIPROC, 250
 - DT_INIT, 250
 - DT_JMPREL, 250

DT_LOPROC, 250
DT_NEEDED, 250
DT_NULL, 250
DT_PLTGOT, 250
DT_PLTREL, 250
DT_PLTRELSZ, 250
DT_REL, 250
DT_RELA, 250
DT_RELAENT, 250
DT_RELASZ, 250
DT_RELENT, 250
DT_RELSZ, 250
DT_RPATH, 250
DT_SONAME, 250
DT_STRSZ, 250
DT_STRTAB, 250
DT_SYMBOLIC, 250
DT_SYMENT, 250
DT_SYMTAB, 250
DT_TEXTREL, 250
EI_CLASS, 250
EI_DATA, 250
EI_MAG0, 250
EI_MAG1, 250
EI_MAG2, 250
EI_MAG3, 250
EI_NIDENT, 250
EI_PAD, 250
EI_VERSION, 250
Elf32_Addr, 253
Elf32_Dyn, 253
Elf32_Ehdr, 253
Elf32_Half, 253
Elf32_Nhdr, 253
Elf32_Off, 253
Elf32_Phdr, 253
ELF32_R_SYM, 250
ELF32_R_TYPE, 250
Elf32_Rel, 253
Elf32_Rela, 253
Elf32_Shdr, 253
ELF32_ST_BIND, 250
ELF32_ST_TYPE, 250
Elf32_Sword, 253
Elf32_Sym, 253
Elf32_Word, 253
Elf64_Ehdr, 253
Elf64_Nhdr, 253
Elf64_Phdr, 253
Elf64_Rel, 253
Elf64_Rela, 253
Elf64_Shdr, 253
Elf64_Sym, 253
ELF_LONG_H, 250
elf_note, 251
elf_phdr, 251
ELF_SHORT_H, 251
ELFCLASS32, 251
ELFCLASS64, 253
ELFCLASSNONE, 253
ELFCLASSNUM, 253
ELFDATA2LSB, 253
ELFDATA2MSB, 253
ELFDATANONE, 253
elfhdr, 253
ELFMAG, 253
ELFMAG0, 253
ELFMAG1, 253
ELFMAG2, 253
ELFMAG3, 253
EM_386, 253
EM_486, 253
EM_68K, 253
EM_860, 253
EM_88K, 253
EM_ALPHA, 253
EM_M32, 253
EM_MIPS, 253
EM_MIPS_RS4_BE, 253
EM_NONE, 253
EM_PARISC, 253
EM_PPC, 253
EM_SPARC, 253
EM_SPARC32PLUS, 253
EM_SPARC64, 253
ET_CORE, 253
ET_DYN, 253
ET_EXEC, 253
ET_HIPROC, 253
ET_LOPROC, 253
ET_NONE, 253
ET_REL, 253
EV_CURRENT, 253
EV_NONE, 253
EV_NUM, 253
NT_PRFPREG, 253
NT_PRPSINFO, 253
NT_PRSTATUS, 253
NT_TASKSTRUCT, 253
PF_R, 253
PF_W, 253
PF_X, 253
PT_DYNAMIC, 253
PT_HIPROC, 253
PT_INTERP, 253
PT_LOAD, 253
PT_LOPROC, 253
PT_NOTE, 253

- PT_NULL, 253
- PT_PHDR, 253
- PT_SHLIB, 253
- R_386_32, 253
- R_386_COPY, 253
- R_386_GLOB_DAT, 253
- R_386_GOT32, 253
- R_386_GOTOFF, 253
- R_386_GOTPC, 253
- R_386_JMP_SLOT, 253
- R_386_NONE, 253
- R_386_NUM, 253
- R_386_PC32, 253
- R_386_PLT32, 253
- R_386_RELATIVE, 253
- R_68K_16, 253
- R_68K_32, 253
- R_68K_8, 253
- R_68K_COPY, 253
- R_68K_GLOB_DAT, 253
- R_68K_GOT16, 253
- R_68K_GOT160, 253
- R_68K_GOT32, 253
- R_68K_GOT320, 253
- R_68K_GOT8, 253
- R_68K_GOT80, 253
- R_68K_JMP_SLOT, 253
- R_68K_NONE, 253
- R_68K_PC16, 253
- R_68K_PC32, 253
- R_68K_PC8, 253
- R_68K_PLT16, 253
- R_68K_PLT160, 253
- R_68K_PLT32, 253
- R_68K_PLT320, 253
- R_68K_PLT8, 253
- R_68K_PLT80, 253
- R_68K_RELATIVE, 253
- SELMAG, 253
- SHF_ALLOC, 253
- SHF_EXECINSTR, 253
- SHF_MASKPROC, 253
- SHF_WRITE, 253
- SHN_ABS, 253
- SHN_COMMON, 253
- SHN_HIPROC, 253
- SHN_HIRESERVE, 253
- SHN_LOPROC, 253
- SHN_LORESERVE, 253
- SHN_UNDEF, 253
- SHT_DYNAMIC, 253
- SHT_DYNSYM, 253
- SHT_HASH, 253
- SHT_HIPROC, 253
- SHT_HIUSER, 253
- SHT_LOPROC, 253
- SHT_LOUSER, 253
- SHT_NOBITS, 253
- SHT_NOTE, 253
- SHT_NULL, 253
- SHT_NUM, 253
- SHT_PROGBITS, 253
- SHT_REL, 253
- SHT_RELA, 253
- SHT_SHLIB, 253
- SHT_STRTAB, 253
- SHT_SYMTAB, 253
- STB_GLOBAL, 253
- STB_LOCAL, 253
- STB_WEAK, 253
- STT_FILE, 253
- STT_FUNC, 253
- STT_NOTYPE, 253
- STT_OBJECT, 253
- STT_SECTION, 253
- Elf32_Addr
 - elf.h, 253
- Elf32_Dyn
 - elf.h, 253
- Elf32_Ehdr
 - elf.h, 253
- Elf32_Half
 - elf.h, 253
- elf32_hdr, 88
 - e_ehsize, 89
 - e_entry, 89
 - e_flags, 89
 - e_ident, 89
 - e_machine, 89
 - e_phentsize, 89
 - e_phnum, 89
 - e_phoff, 89
 - e_shentsize, 89
 - e_shnum, 89
 - e_shoff, 89
 - e_shstrndx, 89
 - e_type, 89
 - e_version, 89
- Elf32_Nhdr
 - elf.h, 253
- elf32_note, 90
 - n_descsz, 90
 - n_namesz, 90
 - n_type, 90
- Elf32_Off
 - elf.h, 253
- Elf32_Phdr
 - elf.h, 253

- elf32_phdr, 91
 - p_align, 91
 - p_filesz, 91
 - p_flags, 91
 - p_memsz, 91
 - p_offset, 91
 - p_paddr, 91
 - p_type, 91
 - p_vaddr, 91
- ELF32_R_SYM
 - elf.h, 250
- ELF32_R_TYPE
 - elf.h, 250
- Elf32_Rel
 - elf.h, 253
- elf32_rel, 92
 - r_info, 92
 - r_offset, 92
- Elf32_Rela
 - elf.h, 253
- elf32_rela, 93
 - r_addend, 93
 - r_info, 93
 - r_offset, 93
- Elf32_Shdr
 - elf.h, 253
- elf32_shdr, 94
 - sh_addr, 94
 - sh_addralign, 94
 - sh_entsize, 94
 - sh_flags, 94
 - sh_info, 94
 - sh_link, 94
 - sh_name, 94
 - sh_offset, 94
 - sh_size, 94
 - sh_type, 94
- ELF32_ST_BIND
 - elf.h, 250
- ELF32_ST_TYPE
 - elf.h, 250
- Elf32_Sword
 - elf.h, 253
- Elf32_Sym
 - elf.h, 253
- elf32_sym, 95
 - st_info, 95
 - st_name, 95
 - st_other, 95
 - st_shndx, 95
 - st_size, 95
 - st_value, 95
- Elf32_Word
 - elf.h, 253
- Elf64_Dyn, 96
 - d_ptr, 96
 - d_tag, 96
 - d_un, 96
 - d_val, 96
- Elf64_Ehdr
 - elf.h, 253
- elf64_hdr, 97
 - e_ehsize, 98
 - e_entry, 98
 - e_flags, 98
 - e_ident, 98
 - e_machine, 98
 - e_phentsize, 98
 - e_phnum, 98
 - e_phoff, 98
 - e_shentsize, 98
 - e_shnum, 98
 - e_shoff, 98
 - e_shstrndx, 98
 - e_type, 98
 - e_version, 98
- Elf64_Nhdr
 - elf.h, 253
- elf64_note, 99
 - n_descsz, 99
 - n_namesz, 99
 - n_type, 99
- Elf64_Phdr
 - elf.h, 253
- elf64_phdr, 100
 - p_align, 100
 - p_filesz, 100
 - p_flags, 100
 - p_memsz, 100
 - p_offset, 100
 - p_paddr, 100
 - p_type, 100
 - p_vaddr, 100
- Elf64_Rel
 - elf.h, 253
- elf64_rel, 101
 - r_info, 101
 - r_offset, 101
- Elf64_Rela
 - elf.h, 253
- elf64_rela, 102
 - r_addend, 102
 - r_info, 102
 - r_offset, 102
- Elf64_Shdr
 - elf.h, 253
- elf64_shdr, 103
 - sh_addr, 103

- sh_addralign, 103
- sh_entsize, 103
- sh_flags, 103
- sh_info, 103
- sh_link, 103
- sh_name, 103
- sh_offset, 103
- sh_size, 103
- sh_type, 103
- Elf64_Sym
 - elf.h, 253
- elf64_sym, 104
 - st_info, 104
 - st_name, 104
 - st_other, 104
 - st_shndx, 104
 - st_size, 104
 - st_value, 104
- elf_close_obj
 - dyngen_elf.c, 356
- elf_find_func
 - dyngen_elf.c, 356
- elf_get_func_len
 - dyngen_elf.c, 356
- elf_get_func_name
 - dyngen_elf.c, 357
- elf_get_func_reloc
 - dyngen_elf.c, 357
- elf_get_func_start
 - dyngen_elf.c, 357
- elf_get_sym_name
 - dyngen_elf.c, 357
- ELF_LONG_H
 - elf.h, 250
- elf_note
 - elf.h, 251
- elf_obj, 105
 - e_hdr, 106
 - e_rel_num, 106
 - e_rel_sec, 106
 - e_rel_sym, 106
 - e_rela_num, 106
 - e_rela_sec, 106
 - e_rela_sym, 106
 - e_relas, 106
 - e_rels, 106
 - e_sections, 106
 - e_shdrs, 106
 - e_sym_num, 106
 - e_sym_str_tab, 106
 - e_syms, 106
- elf_open_obj
 - dyngen_elf.c, 357
- elf_phdr
 - elf.h, 251
- ELF_SHORT_H
 - elf.h, 251
- ELFCLASS32
 - elf.h, 251
- ELFCLASS64
 - elf.h, 253
- ELFCLASSNONE
 - elf.h, 253
- ELFCLASSNUM
 - elf.h, 253
- ELFDATA2LSB
 - elf.h, 253
- ELFDATA2MSB
 - elf.h, 253
- ELFDATANONE
 - elf.h, 253
- elfhdr
 - elf.h, 253
- ELFMAG
 - elf.h, 253
- ELFMAG0
 - elf.h, 253
- ELFMAG1
 - elf.h, 253
- ELFMAG2
 - elf.h, 253
- ELFMAG3
 - elf.h, 253
- EM_386
 - elf.h, 253
- EM_486
 - elf.h, 253
- EM_68K
 - elf.h, 253
- EM_860
 - elf.h, 253
- EM_88K
 - elf.h, 253
- EM_ALPHA
 - elf.h, 253
- EM_M32
 - elf.h, 253
- EM_MIPS
 - elf.h, 253
- EM_MIPS_RS4_BE
 - elf.h, 253
- EM_NONE
 - elf.h, 253
- EM_PARISC
 - elf.h, 253
- EM_PPC
 - elf.h, 253
- EM_SPARC

- elf.h, 253
- EM_SPARC32PLUS
 - elf.h, 253
- EM_SPARC64
 - elf.h, 253
- ENABLE_ADVANCED_POWER_-
MANAGEMENT
 - atacmd.h, 593
- enable_bursts
 - config, 52
- ENABLE_MEDIA_STATUS_NOTIFICATION
 - atacmd.h, 593
- ENABLE_POWERUP_IN_STANDBY_-
FEATURE_SET
 - atacmd.h, 593
- ENABLE_READ_LOOKAHEAD_FEATURE
 - atacmd.h, 593
- ENABLE_RELEASE_INTERRUPT
 - atacmd.h, 593
- ENABLE_REVERTING_TO_POWERON_-
DEFAULTS
 - atacmd.h, 593
- ENABLE_SERVICE_INTERRUPT
 - atacmd.h, 593
- ENABLE_WRITE_CACHE
 - atacmd.h, 593
- enabled
 - ata_host, 24
 - config, 52
 - config::pic, 54
 - dev_16450, 71
 - dev_generic, 74
 - dma_controller, 81
 - dmmu, 83
 - eth_device, 110
 - fb_state, 115
 - gpio_device, 121
 - ic, 124
 - immu, 126
 - kbd_state, 141
 - mc, 144
 - runtime, 160
 - vga_state, 175
- end_addr
 - _cuc_func, 14
 - mprofiler.c, 558
- end_cycles
 - runtime, 160
- end_t_func
 - ata_device, 20
- enough_host_page
 - dyn_rec.c, 332
 - dyn_rec.h, 349
- enter_dyn_code
 - dyn_rec.h, 349
 - op.c, 395
- entry
 - COFF_AOUTHDR, 38
- entrysize
 - dmmu, 83
 - immu, 126
- ERR
 - debug.h, 792
- ERR_CRC
 - gdb.h, 524
- ERR_MEM
 - gdb.h, 524
- ERR_NONE
 - gdb.h, 524
- ERR_
 - debug.h, 793
- ERR_ON
 - debug.h, 793
- error
 - ata_device, 20
 - eth_device, 110
- ET_CORE
 - elf.h, 253
- ET_DYN
 - elf.h, 253
- ET_EXEC
 - elf.h, 253
- ET_HIPROC
 - elf.h, 253
- ET_LOPROC
 - elf.h, 253
- ET_NONE
 - elf.h, 253
- ET_REL
 - elf.h, 253
- eth.c
 - DEFAULT_DEBUG_CHANNEL, 673
 - ETH_NUM_VAPI_IDS, 673
 - ETH_VAPI_CTRL, 672
 - ETH_VAPI_DATA, 672
 - ETH_ADDR_SPACE, 672
 - ETH_ALEN, 672
 - eth_baseaddr, 673
 - ETH_BD_BASE, 672
 - ETH_BD_COUNT, 672
 - ETH_BD_SPACE, 672
 - ETH_CMODER_PASSALL_OFFSET, 672
 - ETH_CMODER_RXFLOW_OFFSET, 672
 - ETH_CMODER_TXFLOW_OFFSET, 672
 - ETH_COLLCONF, 672
 - ETH_COLLCONF_COLLVALID_OFFSET, 672

- ETH_COLLCONF_COLLVALID_WIDTH, [672](#)
- ETH_COLLCONF_MAXRET_OFFSET, [672](#)
- ETH_COLLCONF_MAXRET_WIDTH, [672](#)
- eth_controller_rx_clock, [673](#)
- eth_controller_tx_clock, [673](#)
- ETH_CTRLMODER, [672](#)
- eth_dma, [674](#)
- ETH_DMA_RX_TX, [672](#)
- eth_enabled, [674](#)
- ETH_HASH0, [672](#)
- ETH_HASH1, [672](#)
- ETH_INT_MASK, [672](#)
- ETH_INT_MASK_BUSY_M_OFFSET, [672](#)
- ETH_INT_MASK_RXB_M_OFFSET, [672](#)
- ETH_INT_MASK_RXC_M_OFFSET, [672](#)
- ETH_INT_MASK_RXE_M_OFFSET, [672](#)
- ETH_INT_MASK_TXB_M_OFFSET, [672](#)
- ETH_INT_MASK_TXC_M_OFFSET, [672](#)
- ETH_INT_MASK_TXE_M_OFFSET, [672](#)
- ETH_INT_SOURCE, [672](#)
- ETH_INT_SOURCE_BUSY_OFFSET, [672](#)
- ETH_INT_SOURCE_RXB_OFFSET, [672](#)
- ETH_INT_SOURCE_RXC_OFFSET, [672](#)
- ETH_INT_SOURCE_RXE_OFFSET, [672](#)
- ETH_INT_SOURCE_TXB_OFFSET, [672](#)
- ETH_INT_SOURCE_TXC_OFFSET, [672](#)
- ETH_INT_SOURCE_TXE_OFFSET, [672](#)
- ETH_IPGR1, [672](#)
- ETH_IPGR2, [672](#)
- ETH_IPGT, [672](#)
- eth_irq, [674](#)
- ETH_MAC_ADDR0, [672](#)
- ETH_MAC_ADDR1, [672](#)
- ETH_MAXPL, [672](#)
- ETH_MIIADDR_FIAD_OFFSET, [672](#)
- ETH_MIIADDR_FIAD_WIDTH, [672](#)
- ETH_MIIADDR_RGAD_OFFSET, [672](#)
- ETH_MIIADDR_RGAD_WIDTH, [672](#)
- ETH_MIIADDRESS, [672](#)
- ETH_MIICOMM_RSTAT_OFFSET, [672](#)
- ETH_MIICOMM_SCANS_OFFSET, [672](#)
- ETH_MIICOMM_WCDATA_OFFSET, [672](#)
- ETH_MIICOMMAND, [672](#)
- ETH_MIIMODER, [672](#)
- ETH_MIIMODER_CLKDIV_OFFSET, [672](#)
- ETH_MIIMODER_CLKDIV_WIDTH, [672](#)
- ETH_MIIMODER_MRST_OFFSET, [672](#)
- ETH_MIIMODER_NOPRE_OFFSET, [672](#)
- ETH_MIIRX_DATA, [672](#)
- ETH_MIISTAT_BUSY_OFFSET, [672](#)
- ETH_MIISTAT_FAIL_OFFSET, [672](#)
- ETH_MIISTAT_NVALID_OFFSET, [672](#)
- ETH_MIISTATUS, [672](#)
- ETH_MIITX_DATA, [672](#)
- ETH_MODER, [672](#)
- ETH_MODER_BRO_OFFSET, [672](#)
- ETH_MODER_CRCEN_OFFSET, [672](#)
- ETH_MODER_DLYCRCEN_OFFSET, [672](#)
- ETH_MODER_DMAEN_OFFSET, [672](#)
- ETH_MODER_EXDFREN_OFFSET, [672](#)
- ETH_MODER_FULLD_OFFSET, [672](#)
- ETH_MODER_HUGEN_OFFSET, [672](#)
- ETH_MODER_IAM_OFFSET, [672](#)
- ETH_MODER_IFG_OFFSET, [672](#)
- ETH_MODER_LOOPBCK_OFFSET, [672](#)
- ETH_MODER_NOBCKOF_OFFSET, [672](#)
- ETH_MODER_NOPRE_OFFSET, [672](#)
- ETH_MODER_PAD_OFFSET, [672](#)
- ETH_MODER_PRO_OFFSET, [672](#)
- ETH_MODER_RECSCALL_OFFSET, [672](#)
- ETH_MODER_RST_OFFSET, [672](#)
- ETH_MODER_RXEN_OFFSET, [672](#)
- ETH_MODER_TXEN_OFFSET, [672](#)
- ETH_PACKETLEN, [672](#)
- ETH_PACKETLEN_MAXFL_OFFSET, [672](#)
- ETH_PACKETLEN_MAXFL_WIDTH, [672](#)
- ETH_PACKETLEN_MINFL_OFFSET, [672](#)
- ETH_PACKETLEN_MINFL_WIDTH, [672](#)
- eth_read32, [674](#)
- eth_read_rx_file, [675](#)
- eth_reset, [675](#)
- ETH_RTX_FILE, [672](#)
- ETH_RTX SOCK, [672](#)
- eth_rtx_type, [675](#)
- ETH_RTX_VAPI, [672](#)
- ETH_RX_BD_COLLISION_OFFSET, [672](#)
- ETH_RX_BD_CRC_OFFSET, [672](#)
- ETH_RX_BD_DRIBBLE_OFFSET, [672](#)
- ETH_RX_BD_INVALID_OFFSET, [672](#)
- ETH_RX_BD_IRQ_OFFSET, [672](#)
- ETH_RX_BD_LENGTH_OFFSET, [672](#)
- ETH_RX_BD_LENGTH_WIDTH, [672](#)
- ETH_RX_BD_MISS_OFFSET, [672](#)
- ETH_RX_BD_READY_OFFSET, [672](#)
- ETH_RX_BD_TOOBIG_OFFSET, [672](#)
- ETH_RX_BD_TOOSHORT_OFFSET, [672](#)
- ETH_RX_BD_UVERRUN_OFFSET, [672](#)
- ETH_RX_BD_WRAP_OFFSET, [672](#)
- eth_rx_channel, [675](#)
- eth_rx_next_packet, [675](#)
- eth_rxfile, [675](#)
- ETH_RXSTATE_IDLE, [672](#)
- ETH_RXSTATE_RECV, [672](#)
- ETH_RXSTATE_WAIT4BD, [672](#)
- ETH_RXSTATE_WRITEFIFO, [672](#)
- eth_sec_end, [676](#)
- eth_sec_start, [679](#)

- eth_skip_rx_file, 679
- eth_sockif, 679
- eth_status, 679
- ETH_TX_BD_COLLISION_OFFSET, 672
- ETH_TX_BD_CRC_OFFSET, 672
- ETH_TX_BD_DEFER_OFFSET, 672
- ETH_TX_BD_IRQ_OFFSET, 672
- ETH_TX_BD_LAST_OFFSET, 672
- ETH_TX_BD_LENGTH_OFFSET, 672
- ETH_TX_BD_LENGTH_WIDTH, 672
- ETH_TX_BD_NO_CARRIER_OFFSET, 672
- ETH_TX_BD_NUM, 672
- ETH_TX_BD_PAD_OFFSET, 672
- ETH_TX_BD_PAUSE_OFFSET, 672
- ETH_TX_BD_READY_OFFSET, 672
- ETH_TX_BD_RETRANSMIT_OFFSET, 672
- ETH_TX_BD_RETRY_OFFSET, 672
- ETH_TX_BD_RETRY_WIDTH, 672
- ETH_TX_BD_UNDERRUN_OFFSET, 672
- ETH_TX_BD_WRAP_OFFSET, 672
- eth_tx_channel, 679
- eth_txfile, 679
- ETH_TXSTATE_IDLE, 672
- ETH_TXSTATE_READFIFO, 672
- ETH_TXSTATE_TRANSMIT, 672
- ETH_TXSTATE_WAIT4BD, 672
- eth_vapi_id, 679
- eth_vapi_read, 680
- eth_write32, 680
- eth_write_tx_bd_num, 680
- ETHER_ADDR_LEN, 672
- ETHER_CRC_LEN, 672
- ETHER_HDR_LEN, 672
- ETHER_IS_VALID_LEN, 672
- ETHER_MAX_LEN, 672
- ETHER_MIN_LEN, 672
- ETHER_TYPE_LEN, 672
- ETHERMIN, 672
- ETHERMTU, 672
- ETHERTYPE_ARP, 672
- ETHERTYPE_IP, 672
- ETHERTYPE_NTRAILER, 672
- ETHERTYPE_PUP, 672
- ETHERTYPE_REVARP, 672
- ETHERTYPE_TRAIL, 672
- reg_ethernet_sec, 680
- eth.h
 - reg_ethernet_sec, 682
- ETH_NUM_VAPI_IDS
 - eth.c, 673
- ETH_VAPI_CTRL
 - eth.c, 672
- ETH_VAPI_DATA
 - eth.c, 672
- ETH_ADDR_SPACE
 - eth.c, 672
- ETH_ALEN
 - eth.c, 672
- eth_baseaddr
 - eth.c, 673
- ETH_BD_BASE
 - eth.c, 672
- ETH_BD_COUNT
 - eth.c, 672
- ETH_BD_SPACE
 - eth.c, 672
- ETH_CMODER_PASSALL_OFFSET
 - eth.c, 672
- ETH_CMODER_RXFLOW_OFFSET
 - eth.c, 672
- ETH_CMODER_TXFLOW_OFFSET
 - eth.c, 672
- ETH_COLLCONF
 - eth.c, 672
- ETH_COLLCONF_COLLVALID_OFFSET
 - eth.c, 672
- ETH_COLLCONF_COLLVALID_WIDTH
 - eth.c, 672
- ETH_COLLCONF_MAXRET_OFFSET
 - eth.c, 672
- ETH_COLLCONF_MAXRET_WIDTH
 - eth.c, 672
- eth_controller_rx_clock
 - eth.c, 673
- eth_controller_tx_clock
 - eth.c, 673
- ETH_CTRLMODER
 - eth.c, 672
- eth_device, 107
 - add_crc, 110
 - base_vapi_id, 110
 - baseaddr, 110
 - bd, 110
 - bd_addr, 110
 - bd_index, 110
 - bd_ram, 110
 - bytes_left, 110
 - bytes_read, 110
 - bytes_sent, 110
 - collconf, 110
 - controlmoder, 110
 - crc_dly, 110
 - crc_value, 110
 - dma, 110
 - enabled, 110
 - error, 110
 - fd, 110
 - hash0, 110

- hash1, 110
- ifr, 110
- int_mask, 110
- int_source, 110
- ipgr1, 110
- ipgr2, 110
- ipgt, 110
- lo_buff, 110
- loopback_offset, 110
- mac_address, 110
- mac_int, 110
- maximum_length, 110
- miiaddress, 110
- miicommand, 110
- miimoder, 110
- miirx_data, 110
- miistatus, 110
- miitx_data, 110
- minimum_length, 110
- moder, 110
- offset, 110
- packet_length, 110
- packetlen, 110
- regs, 110
- rfds, 110
- rtx_sock, 110
- rtx_type, 110
- rx, 110
- rx_buff, 110
- rx_channel, 110
- rxfd, 110
- rxfile, 110
- sockif, 110
- state, 110
- tx, 110
- tx_bd_num, 110
- tx_buff, 110
- tx_channel, 110
- txfd, 110
- txfile, 110
- waiting_for_dma, 110
- wfds, 110
- working, 110
- eth_dma
 - eth.c, 674
- ETH_DMA_RX_TX
 - eth.c, 672
- eth_enabled
 - eth.c, 674
- ETH_HASH0
 - eth.c, 672
- ETH_HASH1
 - eth.c, 672
- ETH_INT_MASK
 - eth.c, 672
- ETH_INT_MASK_BUSY_M_OFFSET
 - eth.c, 672
- ETH_INT_MASK_RXB_M_OFFSET
 - eth.c, 672
- ETH_INT_MASK_RXC_M_OFFSET
 - eth.c, 672
- ETH_INT_MASK_RXE_M_OFFSET
 - eth.c, 672
- ETH_INT_MASK_TXB_M_OFFSET
 - eth.c, 672
- ETH_INT_MASK_TXC_M_OFFSET
 - eth.c, 672
- ETH_INT_MASK_TXE_M_OFFSET
 - eth.c, 672
- ETH_INT_SOURCE
 - eth.c, 672
- ETH_INT_SOURCE_BUSY_OFFSET
 - eth.c, 672
- ETH_INT_SOURCE_RXB_OFFSET
 - eth.c, 672
- ETH_INT_SOURCE_RXC_OFFSET
 - eth.c, 672
- ETH_INT_SOURCE_RXE_OFFSET
 - eth.c, 672
- ETH_INT_SOURCE_TXB_OFFSET
 - eth.c, 672
- ETH_INT_SOURCE_TXC_OFFSET
 - eth.c, 672
- ETH_INT_SOURCE_TXE_OFFSET
 - eth.c, 672
- ETH_IPGR1
 - eth.c, 672
- ETH_IPGR2
 - eth.c, 672
- ETH_IPGT
 - eth.c, 672
- eth_irq
 - eth.c, 674
- ETH_MAC_ADDR0
 - eth.c, 672
- ETH_MAC_ADDR1
 - eth.c, 672
- ETH_MAXPL
 - eth.c, 672
- ETH_MIIADDR_FIAD_OFFSET
 - eth.c, 672
- ETH_MIIADDR_FIAD_WIDTH
 - eth.c, 672
- ETH_MIIADDR_RGAD_OFFSET
 - eth.c, 672
- ETH_MIIADDR_RGAD_WIDTH
 - eth.c, 672
- ETH_MIIADDRESS

- eth.c, [672](#)
- ETH_MIICOMM_RSTAT_OFFSET
 - eth.c, [672](#)
- ETH_MIICOMM_SCANS_OFFSET
 - eth.c, [672](#)
- ETH_MIICOMM_WCDATA_OFFSET
 - eth.c, [672](#)
- ETH_MIICOMMAND
 - eth.c, [672](#)
- ETH_MIIMODER
 - eth.c, [672](#)
- ETH_MIIMODER_CLKDIV_OFFSET
 - eth.c, [672](#)
- ETH_MIIMODER_CLKDIV_WIDTH
 - eth.c, [672](#)
- ETH_MIIMODER_MRST_OFFSET
 - eth.c, [672](#)
- ETH_MIIMODER_NOPRE_OFFSET
 - eth.c, [672](#)
- ETH_MIIRX_DATA
 - eth.c, [672](#)
- ETH_MIISTAT_BUSY_OFFSET
 - eth.c, [672](#)
- ETH_MIISTAT_FAIL_OFFSET
 - eth.c, [672](#)
- ETH_MIISTAT_NVALID_OFFSET
 - eth.c, [672](#)
- ETH_MIISTATUS
 - eth.c, [672](#)
- ETH_MIITX_DATA
 - eth.c, [672](#)
- ETH_MODER
 - eth.c, [672](#)
- ETH_MODER_BRO_OFFSET
 - eth.c, [672](#)
- ETH_MODER_CRCEN_OFFSET
 - eth.c, [672](#)
- ETH_MODER_DLYCRCEN_OFFSET
 - eth.c, [672](#)
- ETH_MODER_DMAEN_OFFSET
 - eth.c, [672](#)
- ETH_MODER_EXDFREN_OFFSET
 - eth.c, [672](#)
- ETH_MODER_FULLD_OFFSET
 - eth.c, [672](#)
- ETH_MODER_HUGEN_OFFSET
 - eth.c, [672](#)
- ETH_MODER_IAM_OFFSET
 - eth.c, [672](#)
- ETH_MODER_IFG_OFFSET
 - eth.c, [672](#)
- ETH_MODER_LOOPBCK_OFFSET
 - eth.c, [672](#)
- ETH_MODER_NOBCKOF_OFFSET
 - eth.c, [672](#)
- eth.c, [672](#)
- ETH_MODER_NOPRE_OFFSET
 - eth.c, [672](#)
- ETH_MODER_PAD_OFFSET
 - eth.c, [672](#)
- ETH_MODER_PRO_OFFSET
 - eth.c, [672](#)
- ETH_MODER_RECSCALL_OFFSET
 - eth.c, [672](#)
- ETH_MODER_RST_OFFSET
 - eth.c, [672](#)
- ETH_MODER_RXEN_OFFSET
 - eth.c, [672](#)
- ETH_MODER_TXEN_OFFSET
 - eth.c, [672](#)
- ETH_PACKETLEN
 - eth.c, [672](#)
- ETH_PACKETLEN_MAXFL_OFFSET
 - eth.c, [672](#)
- ETH_PACKETLEN_MAXFL_WIDTH
 - eth.c, [672](#)
- ETH_PACKETLEN_MINFL_OFFSET
 - eth.c, [672](#)
- ETH_PACKETLEN_MINFL_WIDTH
 - eth.c, [672](#)
- eth_read32
 - eth.c, [674](#)
- eth_read_rx_file
 - eth.c, [675](#)
- eth_reset
 - eth.c, [675](#)
- ETH_RTX_FILE
 - eth.c, [672](#)
- ETH_RTX SOCK
 - eth.c, [672](#)
- eth_rtx_type
 - eth.c, [675](#)
- ETH_RTX_VAPI
 - eth.c, [672](#)
- ETH_RX_BD_COLLISION_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_CRC_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_DRIBBLE_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_INVALID_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_IRQ_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_LENGTH_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_LENGTH_WIDTH
 - eth.c, [672](#)
- ETH_RX_BD_MISS_OFFSET
 - eth.c, [672](#)

- eth.c, [672](#)
- ETH_RX_BD_READY_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_TOOBIG_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_TOOSHORT_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_UVERRUN_OFFSET
 - eth.c, [672](#)
- ETH_RX_BD_WRAP_OFFSET
 - eth.c, [672](#)
- eth_rx_channel
 - eth.c, [675](#)
- eth_rx_next_packet
 - eth.c, [675](#)
- eth_rxfile
 - eth.c, [675](#)
- ETH_RXSTATE_IDLE
 - eth.c, [672](#)
- ETH_RXSTATE_RECVC
 - eth.c, [672](#)
- ETH_RXSTATE_WAIT4BD
 - eth.c, [672](#)
- ETH_RXSTATE_WRITEFIFO
 - eth.c, [672](#)
- eth_sec_end
 - eth.c, [676](#)
- eth_sec_start
 - eth.c, [679](#)
- eth_skip_rx_file
 - eth.c, [679](#)
- eth_sockif
 - eth.c, [679](#)
- eth_status
 - eth.c, [679](#)
- ETH_TX_BD_COLLISION_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_CRC_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_DEFER_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_IRQ_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_LAST_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_LENGTH_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_LENGTH_WIDTH
 - eth.c, [672](#)
- ETH_TX_BD_NO_CARRIER_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_NUM
 - eth.c, [672](#)
- ETH_TX_BD_PAD_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_PAUSE_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_READY_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_RETRANSMIT_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_RETRY_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_RETRY_WIDTH
 - eth.c, [672](#)
- ETH_TX_BD_UNDERRUN_OFFSET
 - eth.c, [672](#)
- ETH_TX_BD_WRAP_OFFSET
 - eth.c, [672](#)
- eth_tx_channel
 - eth.c, [679](#)
- eth_txfile
 - eth.c, [679](#)
- ETH_TXSTATE_IDLE
 - eth.c, [672](#)
- ETH_TXSTATE_READFIFO
 - eth.c, [672](#)
- ETH_TXSTATE_TRANSMIT
 - eth.c, [672](#)
- ETH_TXSTATE_WAIT4BD
 - eth.c, [672](#)
- eth_vapi_id
 - eth.c, [679](#)
- eth_vapi_read
 - eth.c, [680](#)
- eth_write32
 - eth.c, [680](#)
- eth_write_tx_bd_num
 - eth.c, [680](#)
- ether_addr, [112](#)
 - ether_addr_octet, [112](#)
- ETHER_ADDR_LEN
 - eth.c, [672](#)
- ether_addr_octet
 - ether_addr, [112](#)
- ETHER_CRC_LEN
 - eth.c, [672](#)
- ether_dhost
 - ether_header, [113](#)
- ETHER_HDR_LEN
 - eth.c, [672](#)
- ether_header, [113](#)
 - ether_dhost, [113](#)
 - ether_shost, [113](#)
 - ether_type, [113](#)
- ETHER_IS_VALID_LEN
 - eth.c, [672](#)
- ETHER_MAX_LEN

- eth.c, [672](#)
- ETHER_MIN_LEN
 - eth.c, [672](#)
- ether_shost
 - ether_header, [113](#)
- ether_type
 - ether_header, [113](#)
- ETHER_TYPE_LEN
 - eth.c, [672](#)
- ETHERMIN
 - eth.c, [672](#)
- ETHERMTU
 - eth.c, [672](#)
- ETHERTYPE_ARP
 - eth.c, [672](#)
- ETHERTYPE_IP
 - eth.c, [672](#)
- ETHERTYPE_NTRAILER
 - eth.c, [672](#)
- ETHERTYPE_PUP
 - eth.c, [672](#)
- ETHERTYPE_REVARP
 - eth.c, [672](#)
- ETHERTYPE_TRAIL
 - eth.c, [672](#)
- EV_CURRENT
 - elf.h, [253](#)
- EV_NONE
 - elf.h, [253](#)
- EV_NUM
 - elf.h, [253](#)
- eval_direct16
 - abstract.c, [208](#)
 - abstract.h, [225](#)
- eval_direct32
 - abstract.c, [208](#)
 - abstract.h, [226](#)
- eval_direct8
 - abstract.c, [208](#)
 - abstract.h, [226](#)
- eval_insn
 - abstract.c, [209](#)
 - abstract.h, [227](#)
- eval_insn_ops
 - dyn_rec.c, [332](#)
- eval_label
 - labels.c, [262](#)
 - labels.h, [264](#)
- eval_mem16
 - abstract.c, [209](#)
 - abstract.h, [227](#)
- eval_mem32
 - abstract.c, [209](#)
 - abstract.h, [228](#)
- eval_mem8
 - abstract.c, [210](#)
 - abstract.h, [228](#)
- eval_mem_16_inv
 - abstract.c, [210](#)
- eval_mem_16_inv_direct
 - abstract.c, [211](#)
- eval_mem_32_inv
 - abstract.c, [211](#)
- eval_mem_32_inv_direct
 - abstract.c, [211](#)
- eval_mem_8_inv
 - abstract.c, [212](#)
- eval_mem_8_inv_direct
 - abstract.c, [212](#)
- eval_operand_val
 - execute.c, [365](#)
 - execute.h, [257](#)
- evalsim_mem16
 - abstract.c, [212](#)
 - abstract.h, [228](#)
- evalsim_mem32
 - abstract.c, [212](#)
 - abstract.h, [229](#)
- evalsim_mem8
 - abstract.c, [213](#)
 - abstract.h, [229](#)
- evalsim_reg
 - execute.c, [365](#)
 - execute.h, [258](#)
- except.c
 - DEFAULT_DEBUG_CHANNEL, [286](#)
 - except_handle, [286](#)
 - except_name, [286](#)
 - except_names, [286](#)
 - except_pending, [286](#)
- except.h
 - EXCEPT_ALIGN, [289](#)
 - EXCEPT_BUSERR, [289](#)
 - EXCEPT_DPF, [289](#)
 - EXCEPT_DTLBMISS, [289](#)
 - except_handle, [289](#)
 - EXCEPT_ILLEGAL, [289](#)
 - EXCEPT_INT, [289](#)
 - EXCEPT_IPF, [289](#)
 - EXCEPT_ITLBMISS, [289](#)
 - except_pending, [290](#)
 - EXCEPT_RANGE, [289](#)
 - EXCEPT_RESET, [289](#)
 - EXCEPT_SYSCALL, [289](#)
 - EXCEPT_TICK, [289](#)
 - EXCEPT_TRAP, [289](#)
- EXCEPT_ALIGN
 - except.h, [289](#)

- EXCEPT_BUSERR
 - except.h, 289
- EXCEPT_DPF
 - except.h, 289
- EXCEPT_DTLBMISS
 - except.h, 289
- except_handle
 - except.c, 286
 - except.h, 289
- EXCEPT_ILLEGAL
 - except.h, 289
- EXCEPT_INT
 - except.h, 289
- EXCEPT_IPF
 - except.h, 289
- EXCEPT_ITLBMISS
 - except.h, 289
- except_name
 - except.c, 286
- except_names
 - except.c, 286
- except_pending
 - except.c, 286
 - except.h, 290
- EXCEPT_RANGE
 - except.h, 289
- EXCEPT_RESET
 - except.h, 289
- EXCEPT_SYSCALL
 - except.h, 289
- EXCEPT_TICK
 - except.h, 289
- EXCEPT_TRAP
 - except.h, 289
- exe_log
 - config, 52
- exe_log_end
 - config, 52
- exe_log_fn
 - config, 52
- EXE_LOG_HARDWARE
 - sim-config.h, 782
- exe_log_marker
 - config, 52
- EXE_LOG_SIMPLE
 - sim-config.h, 782
- EXE_LOG_SOFTWARE
 - sim-config.h, 782
- exe_log_start
 - config, 52
- exe_log_type
 - config, 52
- exec_main
 - execute.c, 366
- execute.h, 258
- execute.c
 - analysis, 361
 - breakpoint, 368
 - check_depend, 362
 - cpu_clock, 362
 - cpu_reset, 363
 - cpu_state, 368
 - decode_execute, 363
 - decode_execute_wrapper, 363
 - do_stats, 368
 - dump_exe_log, 364
 - dumpreg, 364
 - eval_operand_val, 365
 - evalsim_reg, 365
 - exec_main, 366
 - fetch, 366
 - hist_exec_tail, 368
 - issued_per_cycle, 368
 - l_invalid, 367
 - multitissue, 369
 - next_delay_insn, 369
 - pnnext, 369
 - sbuf_buf, 369
 - sbuf_count, 369
 - sbuf_head, 369
 - sbuf_load, 367
 - sbuf_prev_cycles, 369
 - sbuf_store, 367
 - sbuf_tail, 369
 - sbuf_total_cyc, 369
 - sbuf_wait_cyc, 369
 - SET_OV_FLAG_FN, 361
 - setsim_reg, 368
 - update_pc, 368
- execute.h
 - analysis, 255
 - cpu_clock, 255
 - cpu_reset, 256
 - cpu_state, 260
 - CURINSN, 255
 - depend_operands, 256
 - do_stats, 260
 - dump_exe_log, 256
 - dumpreg, 257
 - eval_operand_val, 257
 - evalsim_reg, 258
 - exec_main, 258
 - hist_exec_tail, 260
 - pnnext, 260
 - sbuf_total_cyc, 260
 - sbuf_wait_cyc, 260
 - setsim_reg, 259
- EXECUTE_DEVICE_DIAGNOSTICS

- atacmd.h, 593
- expand_bb
 - bb.c, 440
 - cuc.h, 469
- expand_branch
 - load.c, 492
- expand_calls
 - load.c, 492
- expand_memory
 - load.c, 493
- expand_signed
 - load.c, 493
- ext
 - config, 52
- EXT_CAST
 - op.c, 394
- ext_int
 - runtime, 160
- EXT_NAME
 - op.c, 394
- ext_read
 - generic.c, 694
- EXT_TYPE
 - op.c, 394
- ext_write
 - generic.c, 694
- extend_imm
 - or32.c, 429
- external_clock
 - gpio_device, 121
- extract_function
 - cuc.c, 450
- f_flags
 - COFF_filehdr, 43
- f_magic
 - COFF_filehdr, 43
- f_nscns
 - COFF_filehdr, 43
- f_nsyms
 - COFF_filehdr, 43
- f_opthdr
 - COFF_filehdr, 43
- f_symptr
 - COFF_filehdr, 43
- f_timdat
 - COFF_filehdr, 43
- FALSE
 - gdbcomm.h, 530
- fb.c
 - CAM_SIZEX, 685
 - CAM_SIZEY, 685
 - change_buf_addr, 686
 - CNV16, 685
 - CNV32, 685
 - fb_baseaddr, 686
 - FB_BUFADDR, 685
 - FB_CAMBUFADDR, 685
 - FB_CAMPOSADDR, 685
 - FB_CTRL, 685
 - fb_dump_image24, 686
 - fb_dump_image8, 686
 - fb_enabled, 686
 - fb_filename, 686
 - fb_job, 686
 - FB_PAL, 685
 - fb_read32, 687
 - fb_refresh_rate, 687
 - fb_reset, 687
 - fb_sec_end, 687
 - fb_sec_start, 688
 - FB_SIZEX, 685
 - FB_SIZEY, 685
 - FB_WRAP, 685
 - fb_write32, 688
 - REFRESH_DIVIDER, 685
 - reg_fb_sec, 688
- fb.h
 - reg_fb_sec, 690
- fb_baseaddr
 - fb.c, 686
- FB_BUFADDR
 - fb.c, 685
- FB_CAMBUFADDR
 - fb.c, 685
- FB_CAMPOSADDR
 - fb.c, 685
- FB_CTRL
 - fb.c, 685
- fb_dump_image24
 - fb.c, 686
- fb_dump_image8
 - fb.c, 686
- fb_enabled
 - fb.c, 686
- fb_filename
 - fb.c, 686
- fb_job
 - fb.c, 686
- FB_PAL
 - fb.c, 685
- fb_read32
 - fb.c, 687
- fb_refresh_rate
 - fb.c, 687
- fb_reset
 - fb.c, 687
- fb_sec_end

- fb.c, 687
- fb_sec_start
 - fb.c, 688
- FB_SIZEX
 - fb.c, 685
- FB_SIZEY
 - fb.c, 685
- fb_state, 114
 - addr, 115
 - baseaddr, 115
 - cam_addr, 115
 - camera_pos, 115
 - camerax, 115
 - cameray, 115
 - ctrl, 115
 - enabled, 115
 - filename, 115
 - in_refresh, 115
 - pal, 115
 - pic, 115
 - refresh, 115
 - refresh_count, 115
 - refresh_rate, 115
- FB_WRAP
 - fb.c, 685
- fb_write32
 - fb.c, 688
- fcr
 - dev_16450, 71
- fd
 - eth_device, 110
 - vapi_handler, 173
- fd.c
 - fd_channel_ops, 634
 - fd_init, 634
 - fd_isok, 634
 - fd_read, 634
 - fd_status, 634
 - fd_status_fd, 634
 - fd_write, 634
- fd.h
 - fd_channel_ops, 635
 - fd_read, 635
 - fd_write, 635
- fd_channel, 116
 - fdin, 116
 - fdout, 116
- fd_channel_ops
 - fd.c, 634
 - fd.h, 635
- fd_init
 - fd.c, 634
- fd_isok
 - fd.c, 634
- fd_read
 - fd.c, 634
 - fd.h, 635
- fd_status
 - fd.c, 634
- fd_status_fd
 - fd.c, 634
- fd_write
 - fd.c, 634
 - fd.h, 635
- fdeps
 - _cuc_func, 14
- fdin
 - fd_channel, 116
- fdout
 - fd_channel, 116
- fds
 - file_channel, 117
 - tcp_channel, 171
 - tty_channel, 172
 - vapi.c, 844
 - xterm_channel, 176
- features
 - ata_device, 20
- fetch
 - execute.c, 366
- fetch_pagefaults
 - immustats_entry, 127
- fetch_tlbhit
 - immustats_entry, 127
- fetch_tlbmiss
 - immustats_entry, 127
- fxe_log
 - runtime, 160
- FIELD_MASK
 - fields.h, 692
- FIELD_SHIFT
 - fields.h, 692
- fields.h
 - ASSIGN_FLAG, 691
 - CLEAR_FLAG, 691
 - FIELD_MASK, 692
 - FIELD_SHIFT, 692
 - FLAG_MASK, 692
 - FLAG_SHIFT, 692
 - GET_FIELD, 692
 - SET_FIELD, 692
 - SET_FLAG, 692
 - TEST_FLAG, 692
- fifo_len
 - dev_16450, 71
- file
 - ata_device, 20
- file.c

- file_channel_ops, 637
- file_close, 637
- file_free, 637
- file_init, 637
- file_open, 637
- file.h
 - file_channel_ops, 638
- file_channel, 117
 - fds, 117
 - namein, 117
 - nameout, 117
- file_channel_ops
 - file.c, 637
 - file.h, 638
- file_close
 - file.c, 637
- file_free
 - file.c, 637
- file_init
 - file.c, 637
- file_open
 - file.c, 637
- filename
 - fb_state, 115
 - runtime, 160
 - vga_state, 175
- find_channel_factory
 - channel.c, 631
- find_dma_controller_ch
 - dma.c, 662
 - dma.h, 665
- find_handler
 - vapi.c, 840
- find_jump_loc
 - dyn_rec.c, 332
- find_label
 - labels.c, 262
 - labels.h, 264
- find_lsc_index
 - verilog.c, 507
- find_t
 - dyn_rec.c, 332
- firmware
 - ata_device, 20
- first
 - cuc_bb, 60
- FIXME
 - debug.h, 793
- FIXME_
 - debug.h, 793
- FIXME_ON
 - debug.h, 793
- FLAG_MASK
 - fields.h, 692
- FLAG_REG
 - cuc.h, 465
- FLAG_SHIFT
 - fields.h, 692
- flog
 - cuc.c, 459
 - cuc.h, 477
- FLUSH_CACHE
 - atacmd.h, 593
- fmprof
 - runtime, 160
- FMTLEN
 - simprintf.c, 810
- fmtstr
 - simprintf.c, 810
- fo
 - sprs.c, 316
- FORCE_RET
 - op_i386.h, 412
- format_func_options
 - cuc.c, 450
- forward
 - branchstat, 30
- fout
 - runtime, 160
- fprof
 - mprofiler.c, 558
 - profiler.c, 751
 - runtime, 160
- free
 - channel_ops, 37
- free_func
 - bb.c, 440
 - cuc.h, 469
- free_job_queue
 - scheduler_struct, 163
- freemem
 - parse.c, 270
- from
 - _csm_list, 12
 - cuc_conv, 61
- from_spr
 - spr_def, 168
- fstats
 - stats.c, 275
- fstats_entry, 118
 - cnt_dynamic, 118
 - depend, 118
 - insn1, 118
 - insn2, 118
- FSTATS_LEN
 - stats.c, 275
- func
 - config_param, 55

- cuc.c, 459
- sched_entry, 162
- func_index
 - verilog.c, 507
- func_offset
 - reloc, 157
- func_struct, 119
 - addr, 119
 - calls, 119
 - cum_cycles, 119
 - name, 119
- func_unit_str
 - stats.c, 276
- func_v
 - cuc.c, 459
- gdb.h
 - ERR_CRC, 524
 - ERR_MEM, 524
 - ERR_NONE, 524
 - JTAG_PROXY_ACCESS_EXCEPTION, 525
 - JTAG_PROXY_COMMAND_NOT_IMPLEMENTED, 525
 - JTAG_PROXY_INVALID_ADDRESS, 525
 - JTAG_PROXY_INVALID_CHAIN, 525
 - JTAG_PROXY_INVALID_COMMAND, 525
 - JTAG_PROXY_INVALID_LENGTH, 525
 - JTAG_PROXY_NO_CONNECTION, 525
 - JTAG_PROXY_OUT_OF_MEMORY, 525
 - JTAG_PROXY_PROTOCOL_ERROR, 525
 - JTAG_PROXY_SERVER_TERMINATED, 525
 - OR1K_JTAG_COMMAND_CHAIN, 525
 - OR1K_JTAG_COMMAND_READ, 525
 - OR1K_JTAG_COMMAND_READ_BLOCK, 525
 - OR1K_JTAG_COMMAND_WRITE, 525
 - OR1K_JTAG_COMMAND_WRITE_BLOCK, 525
 - or1k_jtag_errors, 524
 - or1k_jtag_proxy_protocol_commands, 525
- gdb_enabled
 - config, 52
- gdb_fd
 - gdbcomm.c, 529
- gdb_read
 - gdbcomm.c, 527
- gdb_request
 - gdbcomm.c, 527
- gdb_write
 - gdbcomm.c, 527
- gdbcomm.c
 - block_jtag, 527
 - gdb_fd, 529
 - gdb_read, 527
 - gdb_request, 527
 - gdb_write, 527
 - gdbcomm_init, 528
 - get_server_socket, 528
 - handle_server_socket, 528
 - jtag_request, 528
 - protocol_clean, 529
 - server_fd, 529
 - server_ip, 529
 - server_port, 529
 - tcp_level, 529
- gdbcomm.h
 - block_jtag, 530
 - boolean, 530
 - FALSE, 530
 - gdbcomm_init, 530
 - handle_server_socket, 530
 - TRUE, 530
- gdbcomm_init
 - gdbcomm.c, 528
 - gdbcomm.h, 530
- GEN
 - verilog.c, 507
- gen_code_ent
 - dyn_rec.h, 348
- gen_code_proto
 - dyngen.c, 354
- gen_eval_operands
 - generate.c, 370
- gen_func_proto
 - dyngen.c, 353
- gen_func_reloc
 - archf, 16
- gen_j_imm
 - dyn_rec.c, 333
- gen_j_reg
 - dyn_rec.c, 333
- gen_l_add
 - dyn_rec.c, 333
- gen_l_addc
 - dyn_rec.c, 333
- gen_l_and
 - dyn_rec.c, 333
- gen_l_bf
 - dyn_rec.c, 333
- gen_l_bnf
 - dyn_rec.c, 333
- gen_l_cmov
 - dyn_rec.c, 334
- gen_l_cust1
 - dyn_rec.c, 335
- gen_l_cust2
 - dyn_rec.c, 335

- gen_l_cust3
 - dyn_rec.c, 335
- gen_l_cust4
 - dyn_rec.c, 335
- gen_l_cust5
 - dyn_rec.c, 335
- gen_l_cust6
 - dyn_rec.c, 335
- gen_l_cust7
 - dyn_rec.c, 335
- gen_l_cust8
 - dyn_rec.c, 335
- gen_l_div
 - dyn_rec.c, 335
- gen_l_divu
 - dyn_rec.c, 335
- gen_l_extbs
 - dyn_rec.c, 335
- gen_l_extbz
 - dyn_rec.c, 335
- gen_l_extbs
 - dyn_rec.c, 335
- gen_l_exths
 - dyn_rec.c, 335
- gen_l_exthz
 - dyn_rec.c, 335
- gen_l_extws
 - dyn_rec.c, 335
- gen_l_extwz
 - dyn_rec.c, 335
- gen_l_ffl
 - dyn_rec.c, 335
- gen_l_invalid
 - dyn_rec.c, 335
- gen_l_j
 - dyn_rec.c, 335
- gen_l_jal
 - dyn_rec.c, 335
- gen_l_jalr
 - dyn_rec.c, 336
- gen_l_jr
 - dyn_rec.c, 336
- gen_l_lbs
 - dyn_rec.c, 336
- gen_l_lbz
 - dyn_rec.c, 337
- gen_l_lhs
 - dyn_rec.c, 337
- gen_l_lhz
 - dyn_rec.c, 337
- gen_l_lws
 - dyn_rec.c, 337
- gen_l_lwz
 - dyn_rec.c, 337
- gen_l_mac
 - dyn_rec.c, 337
- gen_l_macrc
 - dyn_rec.c, 338
- gen_l_mfspr
 - dyn_rec.c, 338
- gen_l_movhi
 - dyn_rec.c, 338
- gen_l_msb
 - dyn_rec.c, 338
- gen_l_mtspr
 - dyn_rec.c, 338
- gen_l_mul
 - dyn_rec.c, 338
- gen_l_mulu
 - dyn_rec.c, 338
- gen_l_nop
 - dyn_rec.c, 338
- gen_l_or
 - dyn_rec.c, 338
- gen_l_rfe
 - dyn_rec.c, 338
- gen_l_sb
 - dyn_rec.c, 338
- gen_l_sfeq
 - dyn_rec.c, 338
- gen_l_sfges
 - dyn_rec.c, 339
- gen_l_sfges
 - dyn_rec.c, 339
- gen_l_sfgts
 - dyn_rec.c, 339
- gen_l_sfgtu
 - dyn_rec.c, 339
- gen_l_sfles
 - dyn_rec.c, 339
- gen_l_sfleu
 - dyn_rec.c, 339
- gen_l_sflts
 - dyn_rec.c, 339
- gen_l_sfltu
 - dyn_rec.c, 339
- gen_l_sfne
 - dyn_rec.c, 339
- gen_l_sh
 - dyn_rec.c, 339
- gen_l_sll
 - dyn_rec.c, 339
- gen_l_sra
 - dyn_rec.c, 339
- gen_l_srl
 - dyn_rec.c, 339
- gen_l_sub
 - dyn_rec.c, 339
- gen_l_sw
 - dyn_rec.c, 339

- gen_l_sys
 - dyn_rec.c, 339
- gen_l_trap
 - dyn_rec.c, 340
- gen_l_xor
 - dyn_rec.c, 340
- gen_lf_add_s
 - dyn_rec.c, 340
- gen_lf_div_s
 - dyn_rec.c, 340
- gen_lf_ftoi_s
 - dyn_rec.c, 340
- gen_lf_itof_s
 - dyn_rec.c, 340
- gen_lf_madd_s
 - dyn_rec.c, 340
- gen_lf_mul_s
 - dyn_rec.c, 340
- gen_lf_rem_s
 - dyn_rec.c, 341
- gen_lf_sfreq_s
 - dyn_rec.c, 341
- gen_lf_sfge_s
 - dyn_rec.c, 341
- gen_lf_sfgt_s
 - dyn_rec.c, 341
- gen_lf_sfle_s
 - dyn_rec.c, 341
- gen_lf_sflt_s
 - dyn_rec.c, 341
- gen_lf_sfne_s
 - dyn_rec.c, 342
- gen_lf_sub_s
 - dyn_rec.c, 342
- gen_op_mark_loc
 - dyn_rec.c, 342
- gen_option
 - cuc.c, 450
- gen_reloc
 - archf, 16
- generate.c
 - gen_eval_operands, 370
 - generate_body, 371
 - generate_footer, 371
 - generate_header, 371
 - in_file, 372
 - main, 371
 - out_file, 372
 - out_lines, 372
 - output_call, 371
 - output_function, 372
 - shift_fprintf, 372
 - write_to_reg, 372
- generate_bb_seq
 - bb.c, 440
 - cuc.h, 469
- generate_body
 - generate.c, 371
- generate_footer
 - generate.c, 371
- generate_function
 - cuc.c, 452
- generate_header
 - generate.c, 371
- generate_main
 - verilog.c, 507
 - verilog.h, 509
- generate_time_pretty
 - abstract.c, 213
 - abstract.h, 230
- generic.c
 - ext_read, 694
 - ext_write, 694
 - generic_baseaddr, 694
 - generic_byte_enabled, 694
 - generic_enabled, 694
 - generic_hw_enabled, 694
 - generic_name, 694
 - generic_read_byte, 694
 - generic_read_hw, 694
 - generic_read_word, 694
 - generic_reset, 695
 - generic_sec_end, 695
 - generic_sec_start, 695
 - generic_size, 696
 - generic_status, 696
 - generic_word_enabled, 696
 - generic_write_byte, 696
 - generic_write_hw, 696
 - generic_write_word, 696
 - reg_generic_sec, 696
- generic.h
 - reg_generic_sec, 699
- GENERIC_BYTE
 - dev_generic, 73
- GENERIC_HW
 - dev_generic, 73
- GENERIC_READ
 - dev_generic, 73
- GENERIC_WORD
 - dev_generic, 73
- GENERIC_WRITE
 - dev_generic, 73
- generic_baseaddr
 - generic.c, 694
- generic_byte_enabled
 - generic.c, 694
- generic_close

- channels/generic.c, 698
- channels/generic.h, 700
- generic_enabled
 - generic.c, 694
- generic_free
 - channels/generic.c, 698
 - channels/generic.h, 700
- generic_gen_op
 - dyn_rec.c, 330
- generic_hw_enabled
 - generic.c, 694
- generic_name
 - generic.c, 694
- generic_open
 - channels/generic.c, 698
 - channels/generic.h, 700
- generic_read_byte
 - generic.c, 694
- generic_read_hw
 - generic.c, 694
- generic_read_word
 - generic.c, 694
- generic_reset
 - generic.c, 695
- generic_sec_end
 - generic.c, 695
- generic_sec_start
 - generic.c, 695
- generic_size
 - generic.c, 696
- generic_status
 - generic.c, 696
- generic_word_enabled
 - generic.c, 696
- generic_write_byte
 - generic.c, 696
- generic_write_hw
 - generic.c, 696
- generic_write_word
 - generic.c, 696
- get_devint_reg
 - debug-unit.c, 516, 517
- GET_FIELD
 - fields.h, 692
- get_func_len
 - bff, 26
- get_func_name
 - bff, 26
- get_func_reloc
 - bff, 26
- get_func_start
 - bff, 26
- get_label
 - labels.c, 262
- labels.h, 264
- GET_MEDIA_STATUS
 - atacmd.h, 593
- get_param_str
 - sim-config.c, 774
- get_pc
 - common_i386.h, 319
- get_real_func_len
 - archf, 16
- get_server_socket
 - gdbcomm.c, 528
 - vapi.c, 840
- get_sp
 - rec_i386.h, 433
- glue
 - dyn_rec.h, 348
 - op_1t_op.h, 403
 - op_2t_op.h, 405
 - op_3t_op.h, 407
 - op_extend_op.h, 410
 - op_ff1_op.h, 411
 - op_mac_op.h, 414
 - op_swhb_op.h, 421
 - op_t_reg_mov_op.h, 425
- gpio.c
 - DEFAULT_DEBUG_CHANNEL, 704
 - GPIO_NUM_VAPI_IDS, 703
 - GPIO_VAPI_AUX, 703
 - GPIO_VAPI_CLOCK, 703
 - GPIO_VAPI_DATA, 703
 - GPIO_VAPI_RGPIO_AUX, 703
 - GPIO_VAPI_RGPIO_CTRL, 703
 - GPIO_VAPI_RGPIO_INTE, 703
 - GPIO_VAPI_RGPIO_OE, 703
 - GPIO_VAPI_RGPIO_PTRIG, 703
 - GPIO_ADDR_SPACE, 703
 - gpio_base_vapi_id, 704
 - gpio_baseaddr, 704
 - gpio_clock, 704
 - gpio_device_clock, 704
 - gpio_do_int, 704
 - gpio_enabled, 705
 - gpio_external_clock, 705
 - gpio_irq, 705
 - gpio_read32, 706
 - gpio_reset, 706
 - gpio_sec_end, 706
 - gpio_sec_start, 706
 - gpio_status, 707
 - gpio_vapi_read, 707
 - gpio_write32, 707
 - reg_gpio_sec, 707
 - RGPIO_AUX, 703
 - RGPIO_CTRL, 703

- RGPIO_CTRL_ECLK, 703
- RGPIO_CTRL_INTE, 703
- RGPIO_CTRL_INTS, 703
- RGPIO_CTRL_NEC, 703
- RGPIO_IN, 703
- RGPIO_INTE, 703
- RGPIO_INTS, 703
- RGPIO_OE, 703
- RGPIO_OUT, 703
- RGPIO_PTRIG, 703
- gpio.h
 - reg_gpio_sec, 709
- GPIO_NUM_VAPI_IDS
 - gpio.c, 703
- GPIO_VAPI_AUX
 - gpio.c, 703
- GPIO_VAPI_CLOCK
 - gpio.c, 703
- GPIO_VAPI_DATA
 - gpio.c, 703
- GPIO_VAPI_RGPIO_AUX
 - gpio.c, 703
- GPIO_VAPI_RGPIO_CTRL
 - gpio.c, 703
- GPIO_VAPI_RGPIO_INTE
 - gpio.c, 703
- GPIO_VAPI_RGPIO_OE
 - gpio.c, 703
- GPIO_VAPI_RGPIO_PTRIG
 - gpio.c, 703
- GPIO_ADDR_SPACE
 - gpio.c, 703
- gpio_base_vapi_id
 - gpio.c, 704
- gpio_baseaddr
 - gpio.c, 704
- gpio_clock
 - gpio.c, 704
- gpio_device, 120
 - aux, 121
 - auxiliary_inputs, 121
 - base_vapi_id, 121
 - baseaddr, 121
 - ctrl, 121
 - curr, 121
 - enabled, 121
 - external_clock, 121
 - gpio_number, 121
 - in, 121
 - inte, 121
 - ints, 121
 - irq, 121
 - next, 121
 - oe, 121
 - out, 121
 - ptrig, 121
- gpio_device_clock
 - gpio.c, 704
- gpio_do_int
 - gpio.c, 704
- gpio_enabled
 - gpio.c, 705
- gpio_external_clock
 - gpio.c, 705
- gpio_irq
 - gpio.c, 705
- gpio_number
 - gpio_device, 121
- gpio_read32
 - gpio.c, 706
- gpio_reset
 - gpio.c, 706
- gpio_sec_end
 - gpio.c, 706
- gpio_sec_start
 - gpio.c, 706
- gpio_status
 - gpio.c, 707
- gpio_vapi_read
 - gpio.c, 707
- gpio_write32
 - gpio.c, 707
- GPR_T
 - def_op_t.h, 321
- group_bits
 - mprofiler.c, 558
- handle_server_socket
 - gdbcomm.c, 528
 - gdbcomm.h, 530
- handle_sim_command
 - sim-cmd.c, 757
 - sim-cmd.h, 771
- handler_fits_id
 - vapi.c, 841
- has_breakpoint
 - labels.c, 262
 - labels.h, 264
- hash
 - mprofiler.c, 558
- hash0
 - eth_device, 110
- hash1
 - eth_device, 110
- hash_add
 - mprofiler.c, 557
- HASH_FUNC
 - mprofiler.c, 557

- hash_get
 - mprofiler.c, 557
- HASH_SIZE
 - mprofiler.c, 557
- hazards
 - config, 52
- hazardwait
 - runtime, 160
- head
 - channel.c, 631
- heads
 - ata_device, 20
- heads_per_cylinder
 - ata_device, 20
- height
 - INFOHEADER, 128
- hide_device_id
 - config, 52
- high32
 - common_i386.h, 319
- hist_exec, 122
 - addr, 122
 - next, 122
 - prev, 122
- hist_exec_tail
 - abstract.h, 236
 - execute.c, 368
 - execute.h, 260
- HISTEXEC_LEN
 - abstract.h, 224
- history
 - config, 52
- hit
 - bpbstat, 29
 - bticstat, 33
- hitdelay
 - config, 52
 - dmmu, 83
 - ic, 124
 - immu, 126
- hlen
 - vga_state, 175
- host
 - ata_device, 20
- host_len
 - dyn_page, 86
- host_page
 - dyn_page, 86
- htim
 - vga_state, 175
- hush
 - runtime, 160
- hw_enabled
 - dev_generic, 74
- i386_gen_func_reloc
 - dynngen_i386.c, 358
- i386_gen_reloc
 - dynngen_i386.c, 358
- i386_get_real_func_len
 - dynngen_i386.c, 358
- i386_regs.h
 - CPU_STATE_REG, 373
 - NUM_T_REGS, 373
 - T0_REG, 373
 - T1_REG, 373
 - T2_REG, 373
- IADDR_PAGE
 - immu.h, 554
- ic, 123
 - block_mask, 124
 - block_offset_mask, 124
 - blocksize, 124
 - blocksize_log2, 124
 - enabled, 124
 - hitdelay, 124
 - last_way, 124
 - lrus, 124
 - mem, 124
 - missdelay, 124
 - nsets, 124
 - nways, 124
 - set_mask, 124
 - tagaddr_mask, 124
 - tags, 124
 - ustates, 124
 - ustates_reload, 124
- ic_blocksize
 - icache-model.c, 193
- ic_enabled
 - icache-model.c, 193
- ic_end_sec
 - icache-model.c, 193
- ic_hitdelay
 - icache-model.c, 194
- ic_info
 - icache-model.c, 194
- ic_inv
 - icache-model.c, 194
 - icache-model.h, 198
- ic_missdelay
 - icache-model.c, 194
- ic_nsets
 - icache-model.c, 194
- ic_nways
 - icache-model.c, 194
- ic_simulate_fetch
 - icache-model.c, 195
 - icache-model.h, 198

- ic_start_sec
 - icache-model.c, 195
- ic_state
 - icache-model.c, 196
 - icache-model.h, 199
- ic_stats
 - stats.c, 276
 - stats.h, 278
- ic_ustates
 - icache-model.c, 195
- icache-model.c
 - ic_blocksize, 193
 - ic_enabled, 193
 - ic_end_sec, 193
 - ic_hitdelay, 194
 - ic_info, 194
 - ic_inv, 194
 - ic_missdelay, 194
 - ic_nsets, 194
 - ic_nways, 194
 - ic_simulate_fetch, 195
 - ic_start_sec, 195
 - ic_state, 196
 - ic_ustates, 195
 - MAX_IC_BLOCK_SIZE, 193
 - MAX_IC_SETS, 193
 - MAX_IC_WAYS, 193
 - MIN_IC_BLOCK_SIZE, 193
 - reg_ic_sec, 196
- icache-model.h
 - ic_inv, 198
 - ic_simulate_fetch, 198
 - ic_state, 199
 - reg_ic_sec, 198
- icomplet
 - cpu_state, 58
- IDENTIFY_DEVICE
 - atacmd.h, 593
- IDENTIFY_PACKET_DEVICE
 - atacmd.h, 593
- identifyfile
 - parse.c, 266
- IDLE
 - atacmd.h, 593
- IDLE_IMMEDIATE
 - atacmd.h, 593
- ier
 - dev_16450, 71
- ifr
 - eth_device, 110
- II_ADD
 - insn.h, 486
- II_AND
 - insn.h, 486
- II_BF
 - insn.h, 486
- II_CALL
 - insn.h, 486
- II_CMOV
 - insn.h, 486
- II_IS_LOAD
 - insn.h, 486
- II_IS_STORE
 - insn.h, 486
- II_LAST
 - insn.h, 486
- II_LB
 - insn.h, 486
- II_LH
 - insn.h, 486
- II_LRBB
 - insn.h, 486
- II_LW
 - insn.h, 486
- II_MASK
 - insn.h, 486
- II_MEM
 - insn.h, 486
- II_MEM_WIDTH
 - insn.h, 486
- II_MUL
 - insn.h, 486
- II_NOP
 - insn.h, 488
- II_OR
 - insn.h, 488
- II_REG
 - insn.h, 488
- II_SB
 - insn.h, 488
- II_SFEQ
 - insn.h, 488
- II_SFGE
 - insn.h, 488
- II_SFGT
 - insn.h, 488
- II_SFLE
 - insn.h, 488
- II_SFLT
 - insn.h, 488
- II_SFNE
 - insn.h, 488
- II_SH
 - insn.h, 488
- II_SIGNED
 - insn.h, 488
- ii_size
 - insn.h, 488

- timings.c, 504
- II_SLL
 - insn.h, 488
- II_SRA
 - insn.h, 488
- II_SRL
 - insn.h, 488
- II_SUB
 - insn.h, 488
- II_SW
 - insn.h, 488
- II_XOR
 - insn.h, 488
- iir
 - dev_16450, 71
- imagesize
 - INFOHEADER, 128
- imm_gen_op
 - dyn_rec.c, 330
- IMM_STATS
 - parse.c, 266
- immu, 125
 - enabled, 126
 - entrysize, 126
 - hitdelay, 126
 - lru_reload, 126
 - missdelay, 126
 - nsets, 126
 - nways, 126
 - page_mask, 126
 - page_offset_mask, 126
 - pagesize, 126
 - pagesize_log2, 126
 - set_mask, 126
 - ustates, 126
 - vpn_mask, 126
- immu.c
 - DEFAULT_DEBUG_CHANNEL, 549
 - immu_enabled, 549
 - immu_end_sec, 549
 - immu_entrysize, 549
 - immu_find_tlbmr, 549
 - immu_hitdelay, 550
 - immu_missdelay, 550
 - immu_nsets, 550
 - immu_nways, 550
 - immu_pagesize, 550
 - immu_start_sec, 551
 - immu_state, 552
 - immu_translate, 551
 - immu_ustates, 551
 - itlb_status, 551
 - peek_into_itlb, 552
 - reg_immu_sec, 552
- immu.h
 - IADDR_PAGE, 554
 - immu_simulate_tlb, 554
 - immu_state, 555
 - immu_translate, 554
 - peek_into_itlb, 554
 - reg_immu_sec, 554
- immu_enabled
 - immu.c, 549
- immu_end_sec
 - immu.c, 549
- immu_entrysize
 - immu.c, 549
- immu_find_tlbmr
 - immu.c, 549
- IMMU_GOT_DISABLED
 - dyn_rec.h, 348
- IMMU_GOT_ENABLED
 - dyn_rec.h, 348
- immu_hitdelay
 - immu.c, 550
- immu_missdelay
 - immu.c, 550
- immu_nsets
 - immu.c, 550
- immu_nways
 - immu.c, 550
- immu_pagesize
 - immu.c, 550
- immu_retranslate
 - dyn_rec.c, 342
- immu_simulate_tlb
 - immu.h, 554
- immu_start_sec
 - immu.c, 551
- immu_state
 - immu.c, 552
 - immu.h, 555
- immu_stats
 - stats.c, 276
 - stats.h, 278
- immu_translate
 - immu.c, 551
 - immu.h, 554
- immu_ustates
 - immu.c, 551
- immustats_entry, 127
 - fetch_pagefaults, 127
 - fetch_tlbhit, 127
 - fetch_tlbmiss, 127
- importantcolours
 - INFOHEADER, 128
- in
 - gpio_device, 121

- in_file
 - generate.c, 372
- in_refresh
 - fb_state, 115
- in_reset
 - debug-unit.c, 518
- incorrect
 - bpbstat, 29
- index
 - cuc_insn, 62
 - mc, 144
- INFOHEADER, 128
 - bits, 128
 - compression, 128
 - height, 128
 - imagesize, 128
 - importantcolours, 128
 - ncolours, 128
 - planes, 128
 - size, 128
 - width, 128
 - xresolution, 128
 - yresolution, 128
- init
 - channel_ops, 37
 - mprofiler.c, 557
 - runtime, 160
- init_bb_reloc
 - _cuc_func, 14
- init_breakpoints
 - labels.c, 262
 - labels.h, 264
- init_defconfig
 - sim-config.c, 774
 - sim-config.h, 783
- init_dyn_recomp
 - dyn_rec.c, 343
 - dyn_rec.h, 349
- init_labels
 - labels.c, 262
 - labels.h, 264
- INITIALIZE_DEVICE_PARAMETERS
 - atacmd.h, 593
- initstats
 - stats.c, 275
 - stats.h, 278
- insert_conditional_facts
 - adv.c, 437
 - cuc.h, 469
- insert_insns
 - cuc.h, 470
 - insn.c, 481
- INSN
 - cuc.h, 465
- insn
 - btic_entry, 32
 - cuc.h, 477
 - cuc_bb, 60
 - cuc_insn, 62
 - iqueue_entry, 129
 - load.c, 494
 - op_queue, 154
 - sstats_entry, 169
- insn.c
 - add_data_dep, 479
 - add_dep, 479
 - add_latches, 479
 - apply_edge_condition, 479
 - change_insn_type, 480
 - cmov_needed, 480
 - count_cmovs, 480
 - cse, 480
 - csm, 480
 - csm_gen, 480
 - cuc_insn_name, 480
 - dispose_list, 481
 - insert_insns, 481
 - insn_uses, 481
 - iteration, 483
 - known, 483
 - main_list, 483
 - optimize_cmov_more, 481
 - optimize_cmovs, 481
 - optimize_tree, 481
 - print_insns, 482
 - print_shared, 482
 - remove_dead, 482
 - remove_nops, 482
 - remove_trivial_regs, 483
 - search_csm, 483
 - set_io, 483
 - tmp_op, 483
 - tmp_opt, 483
 - unmark_tree, 483
- insn.h
 - change_insn_type, 488
 - cuc_insn_name, 488
 - II_ADD, 486
 - II_AND, 486
 - II_BF, 486
 - II_CALL, 486
 - II_CMOV, 486
 - II_IS_LOAD, 486
 - II_IS_STORE, 486
 - II_LAST, 486
 - II_LB, 486
 - II_LH, 486
 - II_LRBB, 486

- II_LW, 486
- II_MASK, 486
- II_MEM, 486
- II_MEM_WIDTH, 486
- II_MUL, 486
- II_NOP, 488
- II_OR, 488
- II_REG, 488
- II_SB, 488
- II_SFEQ, 488
- II_SFGE, 488
- II_SFGT, 488
- II_SFLE, 488
- II_SFLT, 488
- II_SFNE, 488
- II_SH, 488
- II_SIGNED, 488
- ii_size, 488
- II_SLL, 488
- II_SRA, 488
- II_SRL, 488
- II_SUB, 488
- II_SW, 488
- II_XOR, 488
- insn_size, 488
- insn_time, 488
- known, 489
- load_timing_table, 488
- print_shared, 488
- insn1
 - dstats_entry, 85
 - fstats_entry, 118
- insn2
 - dstats_entry, 85
 - fstats_entry, 118
- insn_addr
 - iqueue_entry, 129
 - op_queue, 154
- insn_ci
 - abstract.c, 221
 - abstract.h, 236
- insn_decode
 - or32.c, 429
- insn_ea
 - cpu_state, 57
- insn_extract
 - or32.c, 430
- insn_index
 - iqueue_entry, 129
 - op_queue, 154
 - or32.c, 430
- insn_indexs
 - dyn_page, 86
- insn_len
 - or32.c, 430
- insn_name
 - or32.c, 430
- insn_size
 - insn.h, 488
 - timings.c, 505
- insn_time
 - insn.h, 488
 - timings.c, 505
- insn_uses
 - insn.c, 481
- INSNAME_LEN
 - abstract.h, 224
- insns
 - dyn_page, 86
- insnset.c
 - INSTRUCTION, 377–386
- INSTRUCTION
 - insnset.c, 377–386
- instructions
 - runtime, 160
- int_mask
 - eth_device, 110
- int_msk_a
 - dma_controller, 81
- int_msk_b
 - dma_controller, 81
- int_source
 - eth_device, 110
- int_src_a
 - dma_controller, 81
- int_src_b
 - dma_controller, 81
- int_val
 - param_val, 155
- inte
 - gpio_device, 121
- internal_or1ksim_time
 - libtoplevel.c, 533
- internals
 - ata_device, 20
- intrq
 - ata_device, 20
- ints
 - dev_16450, 71
 - gpio_device, 121
- iordy
 - ata_device, 20
- ipgr1
 - eth_device, 110
- ipgr2
 - eth_device, 110
- ipgt
 - eth_device, 110

- iprompt
 - runtime, 160
- iprompt_run
 - runtime, 160
- iqueue
 - cpu_state, 58
- iqueue_entry, 129
 - insn, 129
 - insn_addr, 129
 - insn_index, 129
- iregs
 - dev_16450, 71
- irq
 - ata_host, 24
 - dev_16450, 71
 - dma_controller, 81
 - gpio_device, 121
 - kbd_state, 141
 - vga_state, 175
- is_ata_hostadr
 - atahost.h, 626
- is_power2
 - misc.c, 800
 - misc.h, 801
- isblank
 - isblank.c, 746
 - port.h, 747
- isblank.c
 - isblank, 746
- isok
 - channel_ops, 37
- issued_per_cycle
 - execute.c, 368
- istat
 - dev_16450, 71
- IT_BBEND
 - cuc.h, 465
- IT_BBSTART
 - cuc.h, 465
- IT_BRANCH
 - cuc.h, 465
- IT_COND
 - cuc.h, 465
- IT_CUT
 - cuc.h, 465
- IT_FLAG1
 - cuc.h, 465
- IT_FLAG2
 - cuc.h, 465
- IT_INDELAY
 - cuc.h, 465
- IT_LATCHED
 - cuc.h, 465
- IT_MEMADD
 - cuc.h, 465
- IT_MEMORY
 - cuc.h, 465
- IT_OUTPUT
 - cuc.h, 465
- IT_SIGNED
 - cuc.h, 465
- IT_UNUSED
 - cuc.h, 465
- IT_VOLATILE
 - cuc.h, 465
- iteration
 - insn.c, 483
- itlb_status
 - immu.c, 551
- jitter
 - dev_16450, 71
- job_queue
 - scheduler_struct, 163
- join_bb
 - bb.c, 440
- join_transfers
 - cuc/memory.c, 496
- JTAG_CHAIN_DEBUG_UNIT
 - debug-unit.h, 519
- JTAG_CHAIN_DEVELOPMENT
 - debug-unit.h, 519
- JTAG_CHAIN_GLOBAL
 - debug-unit.h, 519
- JTAG_CHAIN_TRACE
 - debug-unit.h, 519
- JTAG_CHAIN_WISHBONE
 - debug-unit.h, 519
- JTAG_PROXY_ACCESS_EXCEPTION
 - gdb.h, 525
- JTAG_PROXY_COMMAND_NOT_-
IMPLEMENTED
 - gdb.h, 525
- JTAG_PROXY_INVALID_ADDRESS
 - gdb.h, 525
- JTAG_PROXY_INVALID_CHAIN
 - gdb.h, 525
- JTAG_PROXY_INVALID_COMMAND
 - gdb.h, 525
- JTAG_PROXY_INVALID_LENGTH
 - gdb.h, 525
- JTAG_PROXY_NO_CONNECTION
 - gdb.h, 525
- JTAG_PROXY_OUT_OF_MEMORY
 - gdb.h, 525
- JTAG_PROXY_PROTOCOL_ERROR
 - gdb.h, 525
- JTAG_PROXY_SERVER_TERMINATED

- ps2kbd.c, 726
- kbd_read8
 - ps2kbd.c, 726
- kbd_reset
 - ps2kbd.c, 726
- kbd_rxfile
 - ps2kbd.c, 726
- kbd_sec_end
 - ps2kbd.c, 726
- kbd_sec_start
 - ps2kbd.c, 727
- KBD_SPACE
 - ps2kbd.c, 725
- kbd_state, 141
 - baseaddr, 141
 - buf, 141
 - buf_count, 141
 - buf_head, 141
 - buf_tail, 141
 - ccmd, 141
 - ccmdbyte, 141
 - enabled, 141
 - irq, 141
 - kcmd, 141
 - kresp, 141
 - rxfile, 141
 - rxfs, 141
 - slowdown, 141
- KBD_STATUS_A2
 - ps2kbd.c, 725
- KBD_STATUS_IBF
 - ps2kbd.c, 725
- KBD_STATUS_INH
 - ps2kbd.c, 725
- KBD_STATUS_MOBF
 - ps2kbd.c, 725
- KBD_STATUS_OBF
 - ps2kbd.c, 725
- KBD_STATUS_PERR
 - ps2kbd.c, 725
- KBD_STATUS_SYS
 - ps2kbd.c, 725
- KBD_STATUS_TO
 - ps2kbd.c, 725
- kbd_write8
 - ps2kbd.c, 727
- kcmd
 - kbd_state, 141
- KEEP_ENDIAN_LONG
 - coff.h, 242
- KEEP_ENDIAN_SHORT
 - coff.h, 243
- known
 - insn.c, 483
- insn.h, 489
- kresp
 - kbd_state, 141
- l_addr
 - COFF_lineno, 44
- l_invalid
 - execute.c, 367
- l_inno
 - COFF_lineno, 44
- l_none
 - dyn32_defs.h, 322
 - or32.c, 430
- l_paddr
 - COFF_lineno, 44
- l_symndx
 - COFF_lineno, 44
- label_entry, 142
 - addr, 142
 - name, 142
 - next, 142
- label_hash
 - labels.c, 262
- LABELEND_CHAR
 - dumpverilog.c, 796
- LABELNAME_LEN
 - abstract.h, 224
- labels.c
 - add_breakpoint, 262
 - add_label, 262
 - breakpoints, 262
 - eval_label, 262
 - find_label, 262
 - get_label, 262
 - has_breakpoint, 262
 - init_breakpoints, 262
 - init_labels, 262
 - label_hash, 262
 - LABELS_HASH_SIZE, 262
 - print_breakpoints, 262
 - remove_breakpoint, 262
- labels.h
 - add_breakpoint, 264
 - add_label, 264
 - breakpoints, 264
 - eval_label, 264
 - find_label, 264
 - get_label, 264
 - has_breakpoint, 264
 - init_breakpoints, 264
 - init_labels, 264
 - print_breakpoints, 264
 - remove_breakpoint, 264
- LABELS_HASH_SIZE

- labels.c, 262
- last
 - cuc_bb, 60
- last_used_reg
 - cuc_bb, 60
- last_way
 - ic, 124
- lba
 - ata_device, 20
- lcr
 - dev_16450, 71
- LE16
 - abstract.h, 224
- length
 - jtr_chain_message, 130
 - jtr_read_block_message, 133
 - jtr_read_message, 135
 - jtr_write_block_message, 137
 - jtr_write_message, 139
- letter_range
 - or32.c, 430
- letter_signed
 - or32.c, 430
- libtoplevel.c, 532
 - internal_orlksim_time, 533
 - orlksim_clock_rate, 533
 - orlksim_get_time_period, 533
 - orlksim_init, 533
 - orlksim_interrupt, 536
 - orlksim_is_le, 536
 - orlksim_reset_duration, 536
 - orlksim_run, 536
 - orlksim_set_time_point, 537
- line
 - dc_set, 67
- LINK_REGNO
 - arch.h, 283
- lo_buff
 - eth_device, 110
- load.c
 - build_insn, 491
 - conv, 494
 - cuc_load, 491
 - detect_locals, 492
 - expand_branch, 492
 - expand_calls, 492
 - expand_memory, 493
 - expand_signed, 493
 - insn, 494
 - negate_conditional, 493
 - num_insn, 494
 - print_cuc_insns, 493
 - reloc, 494
 - remove_dslots, 493
 - xchg_insn, 493
- load_hitdelay
 - config, 52
- load_missdelay
 - config, 52
- load_next_descriptor_when_done
 - dma_channel, 78
- load_timing_table
 - insn.h, 488
 - timings.c, 505
- loadcode
 - parse.c, 267
 - parse.h, 271
- loadcycles
 - runtime, 160
- loads_pagefaults
 - dmmustats_entry, 84
- loads_tlbhit
 - dmmustats_entry, 84
- loads_tlbmiss
 - dmmustats_entry, 84
- locs
 - dyn_page, 86
- log
 - cuc.h, 465
 - dev_memarea, 76
 - mem_config, 147
 - mem_ops, 149
- log2_int
 - misc.c, 800
 - misc.h, 801
- log_enabled
 - config, 52
- LONGEST
 - abstract.h, 224
- longlong_val
 - param_val, 155
- loopback
 - dev_16450, 71
- loopback_offset
 - eth_device, 110
- low32
 - common_i386.h, 319
- LRBB_REG
 - cuc.h, 465
- lru
 - bpb_entry, 28
 - btic_entry, 32
 - dc_set, 67
- lru_reload
 - dmmu, 83
 - immu, 126
- lrus
 - ic, 124

- LS_OP_CAST
 - op.c, 394
- LS_OP_FUNC
 - op.c, 394
- LS_OP_NAME
 - op.c, 394
- lsr
 - dev_16450, 71
- lur
 - _cuc_func, 14
- mac_address
 - eth_device, 110
- mac_int
 - eth_device, 110
- magic
 - COFF_AOUTHDR, 38
- main
 - dyngen.c, 353
 - generate.c, 371
 - toplevel-mprofile.c, 821
 - toplevel-profile.c, 824
 - toplevel.c, 836
- main_cuc
 - cuc.c, 453
 - cuc.h, 470
- main_list
 - insn.c, 483
- main_mprofiler
 - mprofiler.c, 557
 - mprofiler.h, 560
- main_profiler
 - profiler.c, 750
 - profiler.h, 753
- mainpage, 538
- mark_cut
 - cuc.h, 473
 - timings.c, 505
- mark_successors
 - adv.c, 437
- mask
 - adv.c, 437
 - spr_bit_def, 167
- masked_increase
 - dma.c, 663
- MATCHPOINTS_TO_NDP
 - spr-defs.h, 298
- MAX
 - cuc.h, 465
- max
 - cuc_insn, 62
- MAX_AUTOMATA_SIZE
 - or32.c, 428
- MAX_BB
 - cuc.h, 465
- max_bb_delay
 - timings.c, 505
- MAX_DC_BLOCK_SIZE
 - dcache-model.h, 190
- MAX_DC_SETS
 - dcache-model.h, 190
- MAX_DC_WAYS
 - dcache-model.h, 190
- max_delay
 - timings.c, 505
- MAX_FUNCS
 - profiler.h, 753
- MAX_GRP_S
 - spr-defs.h, 298
- MAX_IC_BLOCK_SIZE
 - icache-model.c, 193
- MAX_IC_SETS
 - icache-model.c, 193
- MAX_IC_WAYS
 - icache-model.c, 193
- MAX_INSNS
 - cuc.h, 465
- MAX_LEN
 - or32.c, 428
- MAX_MATCHPOINTS
 - spr-defs.h, 300
- max_op
 - adv.c, 437
- MAX_OP_TABLE_SIZE
 - or32.c, 428
- MAX_OPERANDS
 - abstract.h, 224
- MAX_PARAMS
 - dyngen.c, 353
- MAX_PREROLL
 - cuc.h, 465
- MAX_REGS
 - cuc.h, 465
- MAX_SBUF_LEN
 - sim-config.h, 782
- MAX_SKEW
 - 16450.c, 567
- MAX_SPRS
 - spr-defs.h, 300
- MAX_SPRS_PER_GRP
 - spr-defs.h, 300
- MAX_SPRS_PER_GRP_BITS
 - spr-defs.h, 300
- MAX_STACK
 - profiler.c, 750
- MAX_UNROLL
 - cuc.h, 465
- MAX_WATCHPOINTS

- spr-defs.h, 300
- MAX_XTERM_ARGS
 - xterm.c, 646
- maximum_length
 - eth_device, 110
- maxstack
 - profiler.c, 751
- mc, 143
 - ba_mask, 144
 - baseaddr, 144
 - csc, 144
 - csr, 144
 - enabled, 144
 - index, 144
 - mc_areas, 144
 - mc_area, 145
 - mem_config, 147
 - next, 144
 - poc, 144
 - tms, 144
- mc.c
 - DEFAULT_DEBUG_CHANNEL, 714
 - MC_ADDR_SPACE, 714
 - mc_areas, 717
 - MC_BA_MASK, 714
 - MC_BA_MASK_VALID, 714
 - mc_baseaddr, 714
 - MC_CSC, 714
 - MC_CSC_BAS_OFFSET, 714
 - MC_CSC_BW_OFFSET, 714
 - MC_CSC_BW_WIDTH, 714
 - MC_CSC_EN_OFFSET, 714
 - MC_CSC_KRO_OFFSET, 714
 - MC_CSC_MEMTYPE_ASYNC, 714
 - MC_CSC_MEMTYPE_OFFSET, 714
 - MC_CSC_MEMTYPE_SDRAM, 714
 - MC_CSC_MEMTYPE_SSRAM, 714
 - MC_CSC_MEMTYPE_SYNC, 714
 - MC_CSC_MEMTYPE_WIDTH, 714
 - MC_CSC_MS_OFFSET, 714
 - MC_CSC_MS_WIDTH, 714
 - MC_CSC_PEN_OFFSET, 714
 - MC_CSC_SEL_OFFSET, 714
 - MC_CSC_SEL_WIDTH, 714
 - MC_CSC_VALID, 714
 - MC_CSC_WP_OFFSET, 714
 - MC_CSR, 714
 - MC_CSR_VALID, 714
 - mc_done, 714
 - mc_enabled, 714
 - mc_index, 715
 - MC_POC, 714
 - mc_poc, 715
 - MC_POC_EN_BW_OFFSET, 714
 - MC_POC_EN_BW_WIDTH, 714
 - MC_POC_EN_MEMTYPE_OFFSET, 714
 - MC_POC_EN_MEMTYPE_WIDTH, 714
 - MC_POC_VALID, 714
 - mc_read_word, 715
 - mc_reg_mem_area, 715
 - mc_reset, 715
 - mc_sec_end, 715
 - mc_sec_start, 716
 - mc_status, 716
 - MC_TMS, 714
 - MC_TMS_ASYNC_TRDV_OFFSET, 714
 - MC_TMS_ASYNC_TRDV_WIDTH, 714
 - MC_TMS_ASYNC_TRDZ_OFFSET, 714
 - MC_TMS_ASYNC_TRDZ_WIDTH, 714
 - MC_TMS_ASYNC_TWD_OFFSET, 714
 - MC_TMS_ASYNC_TWD_WIDTH, 714
 - MC_TMS_ASYNC_TWPW_OFFSET, 714
 - MC_TMS_ASYNC_TWPW_WIDTH, 714
 - MC_TMS_ASYNC_TWWD_OFFSET, 714
 - MC_TMS_ASYNC_TWWD_WIDTH, 714
 - MC_TMS_ASYNC_VALID, 714
 - MC_TMS_SDRAM_BL_OFFSET, 714
 - MC_TMS_SDRAM_BL_WIDTH, 714
 - MC_TMS_SDRAM_BT_OFFSET, 714
 - MC_TMS_SDRAM_CL_OFFSET, 714
 - MC_TMS_SDRAM_CL_WIDTH, 714
 - MC_TMS_SDRAM_OM_OFFSET, 714
 - MC_TMS_SDRAM_OM_WIDTH, 714
 - MC_TMS_SDRAM_TRCD_OFFSET, 714
 - MC_TMS_SDRAM_TRCD_WIDTH, 714
 - MC_TMS_SDRAM_TRFC_OFFSET, 714
 - MC_TMS_SDRAM_TRFC_WIDTH, 714
 - MC_TMS_SDRAM_TRP_OFFSET, 714
 - MC_TMS_SDRAM_TRP_WIDTH, 714
 - MC_TMS_SDRAM_TWR_OFFSET, 714
 - MC_TMS_SDRAM_TWR_WIDTH, 714
 - MC_TMS_SDRAM_VALID, 714
 - MC_TMS_SDRAM_WBL_OFFSET, 714
 - MC_TMS_SSRAM_VALID, 714
 - MC_TMS_SYNC_TRDV_OFFSET, 714
 - MC_TMS_SYNC_TRDV_WIDTH, 714
 - MC_TMS_SYNC_TRDZ_OFFSET, 714
 - MC_TMS_SYNC_TRDZ_WIDTH, 714
 - MC_TMS_SYNC_TTO_OFFSET, 714
 - MC_TMS_SYNC_TTO_WIDTH, 714
 - MC_TMS_SYNC_TWR_OFFSET, 714
 - MC_TMS_SYNC_TWR_WIDTH, 714
 - MC_TMS_SYNC_VALID, 714
 - MC_TMS_VALID, 714
 - mc_write_word, 716
 - mcs, 717
 - N_CE, 714
 - reg_mc_sec, 716

- set_csc_tms, 717
- mc.h
 - mc_done, 718
 - mc_reg_mem_area, 718
 - reg_mc_sec, 719
- MC_ADDR_SPACE
 - mc.c, 714
- mc_area, 145
 - abstract.c, 221
 - cs, 145
 - mc, 145
 - mem, 145
 - next, 145
- mc_areas
 - mc, 144
 - mc.c, 717
- MC_BA_MASK
 - mc.c, 714
- MC_BA_MASK_VALID
 - mc.c, 714
- mc_baseaddr
 - mc.c, 714
- MC_CSC
 - mc.c, 714
- MC_CSC_BAS_OFFSET
 - mc.c, 714
- MC_CSC_BW_OFFSET
 - mc.c, 714
- MC_CSC_BW_WIDTH
 - mc.c, 714
- MC_CSC_EN_OFFSET
 - mc.c, 714
- MC_CSC_KRO_OFFSET
 - mc.c, 714
- MC_CSC_MEMTYPE_ASYNC
 - mc.c, 714
- MC_CSC_MEMTYPE_OFFSET
 - mc.c, 714
- MC_CSC_MEMTYPE_SDRAM
 - mc.c, 714
- MC_CSC_MEMTYPE_SSRAM
 - mc.c, 714
- MC_CSC_MEMTYPE_SYNC
 - mc.c, 714
- MC_CSC_MEMTYPE_WIDTH
 - mc.c, 714
- MC_CSC_MS_OFFSET
 - mc.c, 714
- MC_CSC_MS_WIDTH
 - mc.c, 714
- MC_CSC_PEN_OFFSET
 - mc.c, 714
- MC_CSC_SEL_OFFSET
 - mc.c, 714
- MC_CSC_SEL_WIDTH
 - mc.c, 714
- MC_CSC_VALID
 - mc.c, 714
- MC_CSC_WP_OFFSET
 - mc.c, 714
- MC_CSR
 - mc.c, 714
- MC_CSR_VALID
 - mc.c, 714
- mc_done
 - mc.c, 714
 - mc.h, 718
- mc_enabled
 - mc.c, 714
- mc_index
 - mc.c, 715
- MC_POC
 - mc.c, 714
- mc_poc
 - mc.c, 715
- MC_POC_EN_BW_OFFSET
 - mc.c, 714
- MC_POC_EN_BW_WIDTH
 - mc.c, 714
- MC_POC_EN_MEMTYPE_OFFSET
 - mc.c, 714
- MC_POC_EN_MEMTYPE_WIDTH
 - mc.c, 714
- MC_POC_VALID
 - mc.c, 714
- mc_read_word
 - mc.c, 715
- mc_reg_mem_area
 - mc.c, 715
 - mc.h, 718
- mc_reset
 - mc.c, 715
- mc_sec_end
 - mc.c, 715
- mc_sec_start
 - mc.c, 716
- mc_status
 - mc.c, 716
- MC_TMS
 - mc.c, 714
- MC_TMS_ASYNC_TRDV_OFFSET
 - mc.c, 714
- MC_TMS_ASYNC_TRDV_WIDTH
 - mc.c, 714
- MC_TMS_ASYNC_TRDZ_OFFSET
 - mc.c, 714
- MC_TMS_ASYNC_TRDZ_WIDTH
 - mc.c, 714

- MC_TMS_ASYNC_TWD_OFFSET
mc.c, [714](#)
- MC_TMS_ASYNC_TWD_WIDTH
mc.c, [714](#)
- MC_TMS_ASYNC_TWPW_OFFSET
mc.c, [714](#)
- MC_TMS_ASYNC_TWPW_WIDTH
mc.c, [714](#)
- MC_TMS_ASYNC_TWWD_OFFSET
mc.c, [714](#)
- MC_TMS_ASYNC_TWWD_WIDTH
mc.c, [714](#)
- MC_TMS_ASYNC_VALID
mc.c, [714](#)
- MC_TMS_SDRAM_BL_OFFSET
mc.c, [714](#)
- MC_TMS_SDRAM_BL_WIDTH
mc.c, [714](#)
- MC_TMS_SDRAM_BT_OFFSET
mc.c, [714](#)
- MC_TMS_SDRAM_CL_OFFSET
mc.c, [714](#)
- MC_TMS_SDRAM_CL_WIDTH
mc.c, [714](#)
- MC_TMS_SDRAM_OM_OFFSET
mc.c, [714](#)
- MC_TMS_SDRAM_OM_WIDTH
mc.c, [714](#)
- MC_TMS_SDRAM_TRCD_OFFSET
mc.c, [714](#)
- MC_TMS_SDRAM_TRCD_WIDTH
mc.c, [714](#)
- MC_TMS_SDRAM_TRFC_OFFSET
mc.c, [714](#)
- MC_TMS_SDRAM_TRFC_WIDTH
mc.c, [714](#)
- MC_TMS_SDRAM_TRP_OFFSET
mc.c, [714](#)
- MC_TMS_SDRAM_TRP_WIDTH
mc.c, [714](#)
- MC_TMS_SDRAM_TWR_OFFSET
mc.c, [714](#)
- MC_TMS_SDRAM_TWR_WIDTH
mc.c, [714](#)
- MC_TMS_SDRAM_VALID
mc.c, [714](#)
- MC_TMS_SDRAM_WBL_OFFSET
mc.c, [714](#)
- MC_TMS_SSRAM_VALID
mc.c, [714](#)
- MC_TMS_SYNC_TRDV_OFFSET
mc.c, [714](#)
- MC_TMS_SYNC_TRDV_WIDTH
mc.c, [714](#)
- MC_TMS_SYNC_TRDZ_OFFSET
mc.c, [714](#)
- MC_TMS_SYNC_TRDZ_WIDTH
mc.c, [714](#)
- MC_TMS_SYNC_TTO_OFFSET
mc.c, [714](#)
- MC_TMS_SYNC_TTO_WIDTH
mc.c, [714](#)
- MC_TMS_SYNC_TWR_OFFSET
mc.c, [714](#)
- MC_TMS_SYNC_TWR_WIDTH
mc.c, [714](#)
- MC_TMS_SYNC_VALID
mc.c, [714](#)
- MC_TMS_VALID
mc.c, [714](#)
- mc_write_word
mc.c, [716](#)
- mcr
dev_16450, [71](#)
- mcs
mc.c, [717](#)
- mdelay
runtime, [160](#)
- mdep
cuc_bb, [60](#)
- MEDIA_EJECT
atacmd.h, [593](#)
- MEDIA_LOCK
atacmd.h, [593](#)
- MEDIA_UNLOCK
atacmd.h, [593](#)
- mem
ata_host, [24](#)
ic, [124](#)
mc_area, [145](#)
mem_config, [147](#)
- mem_config
MT_PATTERN, [146](#)
MT_RANDOM, [146](#)
MT_UNKNOWN, [146](#)
- mem_config, [146](#)
baseaddr, [147](#)
ce, [147](#)
delayr, [147](#)
delayw, [147](#)
log, [147](#)
mc, [147](#)
mem, [147](#)
name, [147](#)
pattern, [147](#)
random_seed, [147](#)
size, [147](#)
type, [147](#)

- mem_cycles
 - runtime, 160
- mem_ops, 148
 - delayr, 149
 - delayw, 149
 - log, 149
 - read_dat16, 149
 - read_dat32, 149
 - read_dat8, 149
 - readfunc16, 149
 - readfunc32, 149
 - readfunc8, 149
 - write_dat16, 149
 - write_dat32, 149
 - write_dat8, 149
 - writfunc16, 149
 - writfunc32, 149
 - writfunc8, 149
 - writeprog32, 149
 - writeprog32_dat, 149
 - writeprog8, 149
 - writeprog8_dat, 149
- mem_ordering_cmp
 - cuc/memory.c, 496
- mem_reset
 - peripheral/memory.c, 499
- memory.h
 - reg_memory_sec, 721
- memory_baseaddr
 - peripheral/memory.c, 499
- memory_ce
 - peripheral/memory.c, 499
- memory_delay
 - timings.c, 505
- memory_delayr
 - peripheral/memory.c, 499
- memory_delayw
 - peripheral/memory.c, 499
- memory_hash, 150
 - addr, 150
 - cnt, 150
 - next, 150
- MEMORY_LEN
 - parse.c, 266
- memory_log
 - peripheral/memory.c, 499
- memory_mc
 - peripheral/memory.c, 499
- memory_name
 - peripheral/memory.c, 499
- memory_order
 - _cuc_func, 14
 - config, 52
- memory_pattern
 - peripheral/memory.c, 499
- memory_random_seed
 - peripheral/memory.c, 499
- memory_sec_end
 - peripheral/memory.c, 500
- memory_sec_start
 - peripheral/memory.c, 500
- memory_size
 - peripheral/memory.c, 500
- memory_table_status
 - abstract.c, 214
 - abstract.h, 230
- memory_type
 - peripheral/memory.c, 500
- MERROR
 - sim-config.c, 774
- mfspr
 - sprs.c, 315
 - sprs.h, 317
- miiaddress
 - eth_device, 110
- miicommand
 - eth_device, 110
- miimoder
 - eth_device, 110
- miirx_data
 - eth_device, 110
- miistatus
 - eth_device, 110
- miitx_data
 - eth_device, 110
- MIN
 - 16450.c, 567
 - cuc.h, 465
 - or32.c, 428
- MIN_DC_BLOCK_SIZE
 - dcache-model.h, 190
- MIN_IC_BLOCK_SIZE
 - icache-model.c, 193
- MIN_MWDMA_CYCLE_TIME
 - atadevice_cmdi.h, 611
- MIN_PIO_CYCLE_TIME_IORDY
 - atadevice_cmdi.h, 611
- MIN_PIO_CYCLE_TIME_NO_IORDY
 - atadevice_cmdi.h, 611
- minimum_length
 - eth_device, 110
- misc.c
 - is_power2, 800
 - log2_int, 800
- misc.h
 - is_power2, 801
 - log2_int, 801
- miss

- bpbstat, 29
 - bticstat, 33
- missdelay
 - config, 52
 - dmmu, 83
 - ic, 124
 - immu, 126
- mmu/dmmu.c, 539
- mmu/dmmu.h, 545
- mmu/immu.c, 548
- mmu/immu.h, 553
- MO_EXACT
 - cuc.h, 465
- MO_NONE
 - cuc.h, 465
- MO_STRONG
 - cuc.h, 465
- MO_WEAK
 - cuc.h, 465
- MODE_ACCESS
 - mprofiler.c, 557
- MODE_DETAIL
 - mprofiler.c, 557
- MODE_PRETTY
 - mprofiler.c, 557
- MODE_WIDTH
 - mprofiler.c, 557
- moder
 - eth_device, 110
- MPROF_16
 - profile.h, 804
- MPROF_32
 - profile.h, 804
- MPROF_8
 - profile.h, 804
- MPROF_FETCH
 - profile.h, 804
- mprof_fn
 - config, 52
- MPROF_READ
 - profile.h, 804
- MPROF_WRITE
 - profile.h, 804
- mprofentry_struct, 151
 - addr, 151
 - type, 151
- mprofile
 - config, 52
 - profile.c, 802
 - profile.h, 804
- mprofiler.c, 556
 - BUF_SIZE, 557
 - end_addr, 558
 - fprof, 558
 - group_bits, 558
 - hash, 558
 - hash_add, 557
 - HASH_FUNC, 557
 - hash_get, 557
 - HASH_SIZE, 557
 - init, 557
 - main_mprofiler, 557
 - MODE_ACCESS, 557
 - MODE_DETAIL, 557
 - MODE_PRETTY, 557
 - MODE_WIDTH, 557
 - nbits, 558
 - printout, 558
 - read_file, 558
 - start_addr, 558
- mprofiler.h, 560
 - main_mprofiler, 560
- msched
 - _cuc_func, 14
- mshr
 - dev_16450, 71
- mstats_entry, 152
 - bf, 152
 - bnf, 152
 - bpb, 152
 - btic, 152
 - byteadd, 152
- MT_PATTERN
 - mem_config, 146
- MT_RANDOM
 - mem_config, 146
- MT_UNKNOWN
 - mem_config, 146
- MT_BURST
 - cuc.h, 465
- MT_BURSTE
 - cuc.h, 465
- MT_CALL
 - cuc.h, 465
- MT_LOAD
 - cuc.h, 465
- MT_SIGNED
 - cuc.h, 465
- MT_STORE
 - cuc.h, 465
- MT_WIDTH
 - cuc.h, 465
- mtspr
 - sprs.c, 315
 - sprs.h, 318
- mtype
 - _cuc_func, 14
- multitissue

- execute.c, 369
- mwdma
 - ata_device, 20
- N_CE
 - mc.c, 714
- n_descsz
 - elf32_note, 90
 - elf64_note, 99
- n_namesz
 - elf32_note, 90
 - elf64_note, 99
- n_type
 - elf32_note, 90
 - elf64_note, 99
- name
 - channel_factory, 36
 - config_param, 55
 - config_section, 56
 - cuc_known_insn, 63
 - dev_generic, 74
 - func_struct, 119
 - label_entry, 142
 - mem_config, 147
 - reloc, 157
 - sim_command, 164
 - spr_bit_def, 167
 - spr_def, 168
 - stack_struct, 170
 - tty.c, 643
- namein
 - file_channel, 117
- nameout
 - file_channel, 117
- nbits
 - mprofiler.c, 558
- ncolours
 - INFOHEADER, 128
- negate_conditional
 - cuc.h, 473
 - load.c, 493
- new_bb_cycles
 - timings.c, 505
- new_dp
 - dyn_rec.c, 343
 - dyn_rec.h, 349
- new_time
 - cuc_timings, 66
- next
 - _csm_list, 12
 - _dep_list_t, 15
 - breakpoint_entry, 31
 - channel_factory, 36
 - config_param, 55
 - config_section, 56
 - cuc_bb, 60
 - dev_memarea, 76
 - dma_controller, 81
 - gpio_device, 121
 - hist_exec, 122
 - label_entry, 142
 - mc, 144
 - mc_area, 145
 - memory_hash, 150
 - op_queue, 154
 - sched_entry, 162
 - sim_reset_hook, 165
 - sim_stat, 166
 - vapi_handler, 173
- next_delay_insn
 - execute.c, 369
- nfdeps
 - _cuc_func, 14
- nfds
 - vapi.c, 844
- nfunccalls
 - profiler.c, 751
- nhandlers
 - vapi.c, 844
- ninsn
 - _csm_list, 12
 - cuc_bb, 60
- nmemory
 - cuc_bb, 60
- nmsched
 - _cuc_func, 14
- no_multicycle
 - config, 52
- nonblocking
 - tcp_channel, 171
- NOP
 - atacmd.h, 593
- NOP_CNT_RESET
 - spr-defs.h, 300
- NOP_EXIT
 - spr-defs.h, 300
- NOP_NOP
 - spr-defs.h, 300
- NOP_PRINTF
 - spr-defs.h, 300
- NOP_PUTC
 - spr-defs.h, 300
- NOP_REPORT
 - spr-defs.h, 300
- NOP_REPORT_FIRST
 - spr-defs.h, 300
- NOP_REPORT_LAST
 - spr-defs.h, 300

- not_jump_loc
 - op_queue, 154
- nottaken
 - branchstat, 30
- nr_sect
 - ata_device, 20
- nsets
 - config, 52
 - dmmu, 83
 - ic, 124
 - immu, 126
- nshared
 - cuc_timings, 66
- nstack
 - profiler.c, 751
- NT_PRFPREG
 - elf.h, 253
- NT_PRPSINFO
 - elf.h, 253
- NT_PRSTATUS
 - elf.h, 253
- NT_TASKSTRUCT
 - elf.h, 253
- ntim
 - cuc_bb, 60
- ntotcalls
 - profiler.c, 751
- num_bb
 - _cuc_func, 14
- num_ids
 - vapi_handler, 173
- num_init_bb
 - _cuc_func, 14
- num_insn
 - cuc.h, 477
 - load.c, 494
- num_ones
 - or32.c, 430
- num_opcodes
 - or32.c, 431
- num_ops
 - op_queue, 154
- num_ops_param
 - op_queue, 154
- num_regs
 - jtr_read_block_message, 133
 - jtr_read_block_response, 134
 - jtr_write_block_message, 137
- num_runs
 - _cuc_func, 14
- NUM_T_REGS
 - i386_regs.h, 373
- nuncovered
 - or32.c, 431
- nways
 - config, 52
 - dmmu, 83
 - ic, 124
 - immu, 126
- oe
 - gpio_device, 121
- offset
 - BMP_HEADER, 27
 - eth_device, 110
- OP
 - op.c, 394
- op
 - cuc_insn, 62
- op.c
 - __op_param1, 401
 - __op_param2, 401
 - __op_param3, 401
 - __or_dynop, 394
 - asm, 394
 - COMP, 394
 - COMP_CAST, 394
 - COMP_NAME, 394
 - do_sched_wrap, 394
 - do_sched_wrap_delay, 395
 - enter_dyn_code, 395
 - EXT_CAST, 394
 - EXT_NAME, 394
 - EXT_TYPE, 394
 - LS_OP_CAST, 394
 - LS_OP_FUNC, 394
 - LS_OP_NAME, 394
 - OP, 394
 - op_add_pc, 395
 - op_analysis, 395
 - OP_CAST, 394
 - op_check_delay_slot, 395
 - op_check_flag, 395
 - op_check_flag_delay, 395
 - op_check_not_flag, 396
 - op_check_not_flag_delay, 396
 - op_clear_delay_insn, 396
 - op_clear_flag, 396
 - op_clear_pc_delay, 396
 - op_do_jump_delay, 396
 - op_do_sched, 396
 - op_do_sched_delay, 396
 - OP_EXTRA, 394
 - OP_FILE, 394
 - op_illegal, 396
 - op_illegal_delay, 398
 - op_jump_imm, 398
 - op_join_mem_cycles, 398

- op_macc, 398
- op_move_gpr10_pc_delay, 398
- op_move_gpr11_pc_delay, 398
- op_move_gpr12_pc_delay, 398
- op_move_gpr13_pc_delay, 398
- op_move_gpr14_pc_delay, 398
- op_move_gpr15_pc_delay, 398
- op_move_gpr16_pc_delay, 398
- op_move_gpr17_pc_delay, 398
- op_move_gpr18_pc_delay, 398
- op_move_gpr19_pc_delay, 398
- op_move_gpr1_pc_delay, 398
- op_move_gpr20_pc_delay, 398
- op_move_gpr21_pc_delay, 398
- op_move_gpr22_pc_delay, 398
- op_move_gpr23_pc_delay, 398
- op_move_gpr24_pc_delay, 398
- op_move_gpr25_pc_delay, 398
- op_move_gpr26_pc_delay, 398
- op_move_gpr27_pc_delay, 398
- op_move_gpr28_pc_delay, 398
- op_move_gpr29_pc_delay, 398
- op_move_gpr2_pc_delay, 398
- op_move_gpr30_pc_delay, 398
- op_move_gpr31_pc_delay, 398
- op_move_gpr3_pc_delay, 398
- op_move_gpr4_pc_delay, 398
- op_move_gpr5_pc_delay, 398
- op_move_gpr6_pc_delay, 398
- op_move_gpr7_pc_delay, 398
- op_move_gpr8_pc_delay, 398
- op_move_gpr9_pc_delay, 398
- OP_NAME, 394
- op_nop_exit, 398
- op_nop_printf, 399
- op_nop_report, 399
- op_nop_report_imm, 399
- op_nop_reset, 399
- OP_PARAM1, 394
- OP_PARAM2, 394
- OP_PARAM3, 394
- op_prep_rfe, 399
- op_prep_sys, 400
- op_prep_sys_delay, 400
- op_prep_trap, 400
- op_prep_trap_delay, 400
- op_set_delay_insn, 400
- op_set_flag, 401
- op_set_pc_delay_imm, 401
- op_set_pc_delay_pc, 401
- op_set_pc_pc_delay, 401
- op_store_insn_ea, 401
- op_store_link_addr_gpr, 401
- prep_except, 401
- S_FUNC, 394
- S_OP_NAME, 394
- save_t_bound, 401
- OP_1T
 - op_1t.h, 402
- op_1t.h
 - OP_1T, 402
 - T, 402
- op_1t_op.h
 - glue, 403
- OP_2T
 - op_2t.h, 404
- op_2t.h
 - OP_2T, 404
 - T, 404
- op_2t_op.h
 - glue, 405
- OP_3T
 - op_3t.h, 406
- op_3t.h
 - OP_3T, 406
 - T, 406
- op_3t_op.h
 - glue, 407
- op_add_pc
 - op.c, 395
- op_analysis
 - op.c, 395
- OP_CAST
 - op.c, 394
- op_check_delay_slot
 - op.c, 395
- op_check_flag
 - op.c, 395
- op_check_flag_delay
 - op.c, 395
- op_check_not_flag
 - op.c, 396
- op_check_not_flag_delay
 - op.c, 396
- op_clear_delay_insn
 - op.c, 396
- op_clear_flag
 - op.c, 396
- op_clear_pc_delay
 - op.c, 396
- op_data
 - or32.c, 431
- op_do_jump_delay
 - op.c, 396
- op_do_sched
 - op.c, 396
- op_do_sched_delay
 - op.c, 396

op_extend_op.h
 glue, 410
OP_EXTRA
 op.c, 394
op_ff1_op.h
 glue, 411
OP_FILE
 op.c, 394
OP_FUNC_PARAM_PREFIX
 dyngen.c, 353
OP_FUNC_PREFIX
 dyngen.c, 353
op_i386.h
 asm, 412
 FORCE_RET, 412
 OP_JUMP, 412
 SPEEDY_CALL, 412
op_illegal
 op.c, 396
op_illegal_delay
 op.c, 398
op_jump_imm
 op.c, 398
op_join_mem_cycles
 op.c, 398
OP_JUMP
 op_i386.h, 412
op_mac_op.h
 glue, 414
op_macc
 op.c, 398
OP_MEM_ACCESS
 abstract.h, 224
op_mftspr_op.h
 op_mtspr_imm_clear, 415
op_move_gpr10_pc_delay
 op.c, 398
op_move_gpr11_pc_delay
 op.c, 398
op_move_gpr12_pc_delay
 op.c, 398
op_move_gpr13_pc_delay
 op.c, 398
op_move_gpr14_pc_delay
 op.c, 398
op_move_gpr15_pc_delay
 op.c, 398
op_move_gpr16_pc_delay
 op.c, 398
op_move_gpr17_pc_delay
 op.c, 398
op_move_gpr18_pc_delay
 op.c, 398
op_move_gpr19_pc_delay
 op.c, 398
op_move_gpr20_pc_delay
 op.c, 398
op_move_gpr21_pc_delay
 op.c, 398
op_move_gpr22_pc_delay
 op.c, 398
op_move_gpr23_pc_delay
 op.c, 398
op_move_gpr24_pc_delay
 op.c, 398
op_move_gpr25_pc_delay
 op.c, 398
op_move_gpr26_pc_delay
 op.c, 398
op_move_gpr27_pc_delay
 op.c, 398
op_move_gpr28_pc_delay
 op.c, 398
op_move_gpr29_pc_delay
 op.c, 398
op_move_gpr2_pc_delay
 op.c, 398
op_move_gpr30_pc_delay
 op.c, 398
op_move_gpr31_pc_delay
 op.c, 398
op_move_gpr3_pc_delay
 op.c, 398
op_move_gpr4_pc_delay
 op.c, 398
op_move_gpr5_pc_delay
 op.c, 398
op_move_gpr6_pc_delay
 op.c, 398
op_move_gpr7_pc_delay
 op.c, 398
op_move_gpr8_pc_delay
 op.c, 398
op_move_gpr9_pc_delay
 op.c, 398
op_mtspr_imm_clear
 op_mftspr_op.h, 415
OP_NAME
 op.c, 394
op_nop_exit
 op.c, 398
op_nop_printf
 op.c, 399
op_nop_report
 op.c, 399
op_nop_report_imm

- op.c, 399
- op_nop_reset
 - op.c, 399
- OP_PARAM1
 - op.c, 394
- OP_PARAM2
 - op.c, 394
- OP_PARAM3
 - op.c, 394
- op_prep_rfe
 - op.c, 399
- op_prep_sys
 - op.c, 400
- op_prep_sys_delay
 - op.c, 400
- op_prep_trap
 - op.c, 400
- op_prep_trap_delay
 - op.c, 400
- op_queue, 153
 - insn, 154
 - insn_addr, 154
 - insn_index, 154
 - jump_local, 154
 - jump_local_loc, 154
 - next, 154
 - not_jump_loc, 154
 - num_ops, 154
 - num_ops_param, 154
 - ops, 154
 - ops_len, 154
 - ops_param, 154
 - ops_param_len, 154
 - param, 154
 - param_num, 154
 - param_type, 154
 - prev, 154
 - reg_t, 154
 - tflags, 154
 - xref, 154
- op_set_delay_insn
 - op.c, 400
- op_set_flag
 - op.c, 401
- op_set_pc_delay_imm
 - op.c, 401
- op_set_pc_delay_pc
 - op.c, 401
- op_set_pc_pc_delay
 - op.c, 401
- op_start
 - or32.c, 431
- op_store_insn_ea
 - op.c, 401
- op_store_link_addr_gpr
 - op.c, 401
- op_support.c
 - do_jump, 417
 - op_support_analysis, 417
 - op_support_nop_exit, 417
 - op_support_nop_printf, 417
 - op_support_nop_report, 417
 - op_support_nop_report_imm, 418
 - op_support_nop_reset, 418
- op_support.h
 - do_jump, 419
 - op_support_analysis, 419
 - op_support_nop_exit, 419
 - op_support_nop_printf, 420
 - op_support_nop_report, 420
 - op_support_nop_report_imm, 420
 - op_support_nop_reset, 420
 - upd_reg_from_t, 420
- op_support_analysis
 - op_support.c, 417
 - op_support.h, 419
- op_support_nop_exit
 - op_support.c, 417
 - op_support.h, 419
- op_support_nop_printf
 - op_support.c, 417
 - op_support.h, 420
- op_support_nop_report
 - op_support.c, 417
 - op_support.h, 420
- op_support_nop_report_imm
 - op_support.c, 418
 - op_support.h, 420
- op_support_nop_reset
 - op_support.c, 418
 - op_support.h, 420
- op_swhb_op.h
 - glue, 421
- op_t_reg_mov_op.h
 - glue, 425
- open
 - channel_ops, 37
- open_file
 - atadevice.c, 597
- open_local
 - atadevice.c, 597
- open_obj
 - bff, 26
- OPERANDNAME_LEN
 - abstract.h, 224
- ops
 - channel, 35
 - channel_factory, 36

- dev_memarea, 76
- op_queue, 154
- OPS_ENLARGE_BY
 - dyn_rec.c, 330
- ops_len
 - op_queue, 154
- ops_param
 - op_queue, 154
- ops_param_len
 - op_queue, 154
- opt
 - cuc_insn, 62
- OPT_BB
 - cuc.h, 465
- OPT_CONST
 - cuc.h, 465
- OPT_DEST
 - cuc.h, 465
- OPT_JUMP
 - cuc.h, 465
- OPT_LRBB
 - cuc.h, 465
- OPT_NONE
 - cuc.h, 465
- OPT_REF
 - cuc.h, 465
- OPT_REGISTER
 - cuc.h, 465
- optimize_bb
 - bb.c, 441
 - cuc.h, 473
- optimize_cmov_more
 - insn.c, 481
- optimize_cmovs
 - cuc.h, 473
 - insn.c, 481
- optimize_tree
 - cuc.h, 473
 - insn.c, 481
- option_char
 - cuc.c, 459
- options_cmd
 - cuc.c, 455
- OR1K_JTAG_COMMAND_CHAIN
 - gdb.h, 525
- OR1K_JTAG_COMMAND_READ
 - gdb.h, 525
- OR1K_JTAG_COMMAND_READ_BLOCK
 - gdb.h, 525
- OR1K_JTAG_COMMAND_WRITE
 - gdb.h, 525
- OR1K_JTAG_COMMAND_WRITE_BLOCK
 - gdb.h, 525
- or1k_jtag_errors
 - gdb.h, 524
- or1k_jtag_proxy_protocol_commands
 - gdb.h, 525
- OR1K_MEM_VERILOG_FOOTER
 - dumpverilog.c, 796
- OR1K_MEM_VERILOG_HEADER
 - dumpverilog.c, 796
- or1k_mstats
 - stats.c, 276
 - stats.h, 278
- or1ksim.h, 561
 - OR1KSIM_RC_BADINIT, 561
 - OR1KSIM_RC_BRKPT, 561
 - OR1KSIM_RC_OK, 561
 - or1ksim_clock_rate, 561
 - or1ksim_get_time_period, 561
 - or1ksim_init, 562
 - or1ksim_interrupt, 562
 - or1ksim_is_le, 562
 - or1ksim_rc, 561
 - or1ksim_reset_duration, 562
 - or1ksim_run, 562
 - or1ksim_set_time_point, 563
- OR1KSIM_RC_BADINIT
 - or1ksim.h, 561
- OR1KSIM_RC_BRKPT
 - or1ksim.h, 561
- OR1KSIM_RC_OK
 - or1ksim.h, 561
- or1ksim_clock_rate
 - libtoplevel.c, 533
 - or1ksim.h, 561
- or1ksim_get_time_period
 - libtoplevel.c, 533
 - or1ksim.h, 561
- or1ksim_init
 - libtoplevel.c, 533
 - or1ksim.h, 562
- or1ksim_interrupt
 - libtoplevel.c, 536
 - or1ksim.h, 562
- or1ksim_is_le
 - libtoplevel.c, 536
 - or1ksim.h, 562
- or1ksim_rc
 - or1ksim.h, 561
- or1ksim_reset_duration
 - libtoplevel.c, 536
 - or1ksim.h, 562
- or1ksim_run
 - libtoplevel.c, 536
 - or1ksim.h, 562
- or1ksim_set_time_point
 - libtoplevel.c, 537

- or1ksim.h, 563
- or32.c
 - automata, 431
 - build_automata, 428
 - cover_insn, 428
 - curpass, 431
 - destruct_automata, 428
 - disassemble_index, 429
 - disassemble_insn, 429
 - disassembled, 431
 - disassembled_str, 431
 - EF, 428
 - EFI, 428
 - EFN, 428
 - extend_imm, 429
 - insn_decode, 429
 - insn_extract, 430
 - insn_index, 430
 - insn_len, 430
 - insn_name, 430
 - l_none, 430
 - letter_range, 430
 - letter_signed, 430
 - MAX_AUTOMATA_SIZE, 428
 - MAX_LEN, 428
 - MAX_OP_TABLE_SIZE, 428
 - MIN, 428
 - num_ones, 430
 - num_opcodes, 431
 - nuncovered, 431
 - op_data, 431
 - op_start, 431
 - or32_debug, 430
 - or32_extract, 430
 - or32_letters, 431
 - or32_opcodes, 431
 - or32_print_immediate, 430
 - or32_print_register, 430
 - parse_params, 430
 - range_cache, 432
 - ti, 432
- or32_debug
 - or32.c, 430
- or32_extract
 - or32.c, 430
- or32_letters
 - or32.c, 431
- or32_opcodes
 - or32.c, 431
- or32_print_immediate
 - or32.c, 430
- or32_print_register
 - or32.c, 430
- or_page
 - dyn_page, 86
- oraddr_t
 - arch.h, 284
- orig_time
 - _cuc_func, 14
- orreg_t
 - arch.h, 284
- orsim_dbcl_set
 - debug.c, 790
- orsim_dbcl_set_name
 - debug.c, 790
 - debug.h, 793
- orsim_dbg_log
 - debug.c, 790
 - debug.h, 794
- osize
 - _csm_list, 12
- out
 - gpio_device, 121
- out_file
 - generate.c, 372
- out_lines
 - generate.c, 372
- output_call
 - generate.c, 371
- output_function
 - generate.c, 372
- output_verilog
 - verilog.c, 507
 - verilog.h, 509
- p_align
 - elf32_phdr, 91
 - elf64_phdr, 100
- p_filesz
 - elf32_phdr, 91
 - elf64_phdr, 100
- p_flags
 - elf32_phdr, 91
 - elf64_phdr, 100
- p_memsz
 - elf32_phdr, 91
 - elf64_phdr, 100
- p_offset
 - elf32_phdr, 91
 - elf64_phdr, 100
- p_paddr
 - elf32_phdr, 91
 - elf64_phdr, 100
- p_type
 - elf32_phdr, 91
 - elf64_phdr, 100
- p_vaddr
 - elf32_phdr, 91

- elf64_phdr, 100
- PACKET
 - atacmd.h, 593
- packet
 - ata_device, 20
- packet_length
 - eth_device, 110
- packetlen
 - eth_device, 110
- page_mask
 - dmmu, 83
 - immu, 126
- page_offset_mask
 - dmmu, 83
 - immu, 126
- pagesize
 - dmmu, 83
 - immu, 126
- pagesize_log2
 - dmmu, 83
 - immu, 126
- pal
 - fb_state, 115
- palette
 - vga_state, 175
- param
 - op_queue, 154
 - sched_entry, 162
- param_num
 - op_queue, 154
- param_t
 - sim-config.h, 782
- param_type
 - op_queue, 154
- param_val, 155
 - addr_val, 155
 - int_val, 155
 - longlong_val, 155
 - str_val, 155
- PARAMS
 - dyn32_defs.h, 322
 - simpl32_defs.h, 435
- params
 - config_section, 56
- paramt_addr
 - sim-config.h, 782
- paramt_int
 - sim-config.h, 782
- paramt_longlong
 - sim-config.h, 782
- paramt_none
 - sim-config.h, 782
- paramt_str
 - sim-config.h, 782
- paramt_word
 - sim-config.h, 782
- parse.c
 - addprogram, 266
 - freemem, 270
 - identifyfile, 266
 - IMM_STATS, 266
 - loadcode, 267
 - MEMORY_LEN, 266
 - readfile_coff, 268
 - readfile_elf, 268
 - readsyms_coff, 269
 - rstrip, 269
 - transl_error, 270
 - transl_table, 270
 - translate, 269
- parse.h
 - loadcode, 271
 - rstrip, 272
- parse_args
 - sim-config.c, 774
 - sim-config.h, 783
- parse_baud
 - tty.c, 643
- parse_dbchs
 - debug.c, 790
 - debug.h, 794
- parse_params
 - or32.c, 430
- pattern
 - mem_config, 147
- pc
 - cpu_state, 58
- pc_delay
 - cpu_state, 58
- pcnext
 - execute.c, 369
 - execute.h, 260
- pctr
 - ata_host, 24
- pdiagi
 - ata_device, 20
- pdiago
 - ata_device, 20
- peek_into_dtlb
 - dmmu.c, 543
 - dmmu.h, 546
- peek_into_itlb
 - immu.c, 552
 - immu.h, 554
- peripheral/16450.c, 564
- peripheral/16450.h, 586
- peripheral/atacmd.h, 590
- peripheral/atadevice.c, 594

- peripheral/atadevice.h, 598
- peripheral/atadevice_cmdi.c, 603
- peripheral/atadevice_cmdi.h, 608
- peripheral/atahost.c, 613
- peripheral/atahost.h, 623
- peripheral/atahost_define.h, 629
- peripheral/channels/channel.c, 630
- peripheral/channels/channel.h, 632
- peripheral/channels/fd.c, 633
- peripheral/channels/fd.h, 635
- peripheral/channels/file.c, 636
- peripheral/channels/file.h, 638
- peripheral/channels/generic.c, 698
- peripheral/channels/generic.h, 700
- peripheral/channels/tcp.c, 639
- peripheral/channels/tcp.h, 641
- peripheral/channels/tty.c, 642
- peripheral/channels/tty.h, 644
- peripheral/channels/xterm.c, 645
- peripheral/channels/xterm.h, 647
- peripheral/crc32.c, 648
- peripheral/crc32.h, 649
- peripheral/dma-defs.h, 650
- peripheral/dma.c, 657
- peripheral/dma.h, 664
- peripheral/eth.c, 666
- peripheral/eth.h, 682
- peripheral/fb.c, 684
- peripheral/fb.h, 690
- peripheral/fields.h, 691
- peripheral/generic.c, 693
- peripheral/generic.h, 699
- peripheral/gpio.c, 701
- peripheral/gpio.h, 709
- peripheral/mc.c, 710
- peripheral/mc.h, 718
- peripheral/memory.c, 498
 - mem_reset, 499
 - memory_baseaddr, 499
 - memory_ce, 499
 - memory_delay, 499
 - memory_delayw, 499
 - memory_log, 499
 - memory_mc, 499
 - memory_name, 499
 - memory_pattern, 499
 - memory_random_seed, 499
 - memory_sec_end, 500
 - memory_sec_start, 500
 - memory_size, 500
 - memory_type, 500
 - reg_memory_sec, 501
 - simmem_read16, 501
 - simmem_read32, 502
 - simmem_read8, 502
 - simmem_read_zero16, 502
 - simmem_read_zero32, 502
 - simmem_read_zero8, 502
 - simmem_write16, 502
 - simmem_write32, 502
 - simmem_write8, 502
 - simmem_write_null16, 502
 - simmem_write_null32, 502
 - simmem_write_null8, 502
- peripheral/memory.h, 720
- peripheral/ps2kbd.c, 722
- peripheral/ps2kbd.h, 729
- peripheral/vga.c, 730
- peripheral/vga.h, 736
- PF_R
 - elf.h, 253
- PF_W
 - elf.h, 253
- PF_X
 - elf.h, 253
- pftr0
 - ata_host, 24
- pftr1
 - ata_host, 24
- pic
 - config, 52
 - fb_state, 115
- pic.c
 - clear_interrupt, 738
 - DEFAULT_DEBUG_CHANNEL, 738
 - pic_edge_trigger, 738
 - pic_enabled, 738
 - pic_ints_en, 738
 - pic_rep_int, 739
 - pic_reset, 739
 - pic_state, 740
 - pic_state_int, 740
 - reg_pic_sec, 739
 - report_interrupt, 740
- pic.h
 - clear_interrupt, 741
 - pic_ints_en, 741
 - pic_reset, 741
 - reg_pic_sec, 741
 - report_interrupt, 742
- pic/pic.c, 737
- pic/pic.h, 741
- pic_edge_trigger
 - pic.c, 738
- pic_enabled
 - pic.c, 738
- pic_ints_en
 - pic.c, 738

- pic.h, 741
- pic_lines
 - cpu_state, 58
- pic_rep_int
 - pic.c, 739
- pic_reset
 - pic.c, 739
 - pic.h, 741
- pic_state
 - pic.c, 740
- pic_state_int
 - pic.c, 740
- pics
 - vga_state, 175
- pid
 - xterm_channel, 176
- pindex
 - vga_state, 175
- pio
 - ata_device, 20
- pio_mode
 - ata_device, 20
- PIO_MODE0_T1
 - atahost.c, 615
 - atahost_define.h, 629
- pio_mode0_t1
 - ata_host, 24
- PIO_MODE0_T2
 - atahost.c, 615
 - atahost_define.h, 629
- pio_mode0_t2
 - ata_host, 24
- PIO_MODE0_T4
 - atahost.c, 615
 - atahost_define.h, 629
- pio_mode0_t4
 - ata_host, 24
- PIO_MODE0_TEOC
 - atahost.c, 615
 - atahost_define.h, 629
- pio_mode0_tec
 - ata_host, 24
- planes
 - INFOHEADER, 128
- pm
 - config, 52
- pm.c
 - pm_enabled, 743
 - pm_reset, 743
 - reg_pm_sec, 743
- pm.h
 - pm_reset, 745
 - reg_pm_sec, 745
- pm/pm.c, 743
- pm/pm.h, 745
- pm_enabled
 - pm.c, 743
- pm_reset
 - pm.c, 743
 - pm.h, 745
- poc
 - mc, 144
- port.h
 - isblank, 747
 - PRi16, 747
 - PRi8, 747
 - strndup, 747
- port/isblank.c, 746
- port/port.h, 747
- port/strndup.c, 748
- port_number
 - tcp_channel, 171
- POWERUP_IN_STANDBY_FEATURE_SET_-
SPINUP
 - atacmd.h, 593
- preloaded
 - channel.c, 631
- prep_except
 - op.c, 401
- preroll
 - cuc_timings, 66
- preunroll_bb
 - cuc.c, 455
- preunroll_loop
 - bb.c, 441
 - cuc.h, 474
- prev
 - cuc_bb, 60
 - hist_exec, 122
 - op_queue, 154
- PRIdREG
 - arch.h, 283
- print_bb_num
 - bb.c, 441
 - cuc.h, 474
- print_breakpoints
 - labels.c, 262
 - labels.h, 264
- print_config
 - sim-config.c, 775
 - sim-config.h, 783
- print_cuc_bb
 - bb.c, 442
 - cuc.h, 474
- print_cuc_insns
 - cuc.h, 474
 - load.c, 493
- print_deps

- verilog.c, 507
- print_insn_exec
 - sim-cmd.c, 758
- print_insn_v
 - verilog.c, 508
- print_insns
 - cuc.h, 475
 - insn.c, 482
- print_op_v
 - verilog.c, 508
- print_option
 - cuc.c, 458
- print_shared
 - insn.c, 482
 - insn.h, 488
- print_turn_off_dep
 - verilog.c, 508
- PRINTF
 - sim-config.h, 782
- printotherstats
 - stats.c, 275
- printout
 - mprofiler.c, 558
- printstats
 - stats.c, 275
 - stats.h, 278
- priv_dat
 - vapi_handler, 173
- PRIx16
 - port.h, 747
- PRIx8
 - port.h, 747
- PRIxADDR
 - arch.h, 283
- PRIxREG
 - arch.h, 284
- prof_acquire
 - profiler.c, 750
 - profiler.h, 754
- prof_cycles
 - profiler.c, 751
 - profiler.h, 754
- prof_fn
 - config, 52
- prof_func
 - profiler.c, 751
 - profiler.h, 754
- prof_nfuncs
 - profiler.c, 751
 - profiler.h, 754
- prof_print
 - profiler.c, 750
- prof_set
 - profiler.c, 751
- profiler.h, 754
- profile
 - config, 52
- profile.c
 - mprofile, 802
- profile.h
 - MPROF_16, 804
 - MPROF_32, 804
 - MPROF_8, 804
 - MPROF_FETCH, 804
 - MPROF_READ, 804
 - MPROF_WRITE, 804
 - mprofile, 804
- profiler.c, 749
 - cumulative, 751
 - fprof, 751
 - main_profiler, 750
 - MAX_STACK, 750
 - maxstack, 751
 - nfuncalls, 751
 - nstack, 751
 - ntotcalls, 751
 - prof_acquire, 750
 - prof_cycles, 751
 - prof_func, 751
 - prof_nfuncs, 751
 - prof_print, 750
 - prof_set, 751
 - quiet, 751
 - stack, 751
- profiler.h, 753
 - main_profiler, 753
 - MAX_FUNCS, 753
 - prof_acquire, 754
 - prof_cycles, 754
 - prof_func, 754
 - prof_nfuncs, 754
 - prof_set, 754
- protocol_clean
 - gdbcomm.c, 529
- ps2kbd.c
 - code, 728
 - kbd_baseaddr, 725
 - KBD_BAUD_RATE, 725
 - KBD_CCMD_DKI, 725
 - KBD_CCMD_EKI, 725
 - KBD_CCMD_RCB, 725
 - KBD_CCMD_ST1, 725
 - KBD_CCMD_ST2, 725
 - KBD_CCMD_WCB, 725
 - KBD_CCMDBYTE_EN, 725
 - KBD_CCMDBYTE_EN2, 725
 - KBD_CCMDBYTE_INT, 725
 - KBD_CCMDBYTE_INT2, 725

- KBD_CCMDBYTE_SYS, [725](#)
- KBD_CCMDBYTE_XLAT, [725](#)
- KBD_CTRL, [725](#)
- KBD_DATA, [725](#)
- kbd_enabled, [725](#)
- kbd_info, [725](#)
- kbd_irq, [725](#)
- kbd_job, [725](#)
- KBD_KCMD_DK, [725](#)
- KBD_KCMD_ECHO, [725](#)
- KBD_KCMD_EK, [725](#)
- KBD_KCMD_RST, [725](#)
- KBD_KCMD_SRL, [725](#)
- KBD_KRESP_ACK, [725](#)
- KBD_KRESP_ECHO, [725](#)
- KBD_KRESP_RSTOK, [725](#)
- KBD_MAX_BUF, [725](#)
- kbd_put, [726](#)
- kbd_read8, [726](#)
- kbd_reset, [726](#)
- kbd_rxfile, [726](#)
- kbd_sec_end, [726](#)
- kbd_sec_start, [727](#)
- KBD_SPACE, [725](#)
- KBD_STATUS_A2, [725](#)
- KBD_STATUS_IBF, [725](#)
- KBD_STATUS_INH, [725](#)
- KBD_STATUS_MOBF, [725](#)
- KBD_STATUS_OBF, [725](#)
- KBD_STATUS_PERR, [725](#)
- KBD_STATUS_SYS, [725](#)
- KBD_STATUS_TO, [725](#)
- kbd_write8, [727](#)
- reg_kbd_sec, [727](#)
- scan_decode, [728](#)
- scan_table, [728](#)
- shift, [728](#)
- ps2kbd.h
 - reg_kbd_sec, [729](#)
- PT_DYNAMIC
 - elf.h, [253](#)
- PT_HIPROC
 - elf.h, [253](#)
- PT_INTERP
 - elf.h, [253](#)
- PT_LOAD
 - elf.h, [253](#)
- PT_LOPROC
 - elf.h, [253](#)
- PT_NOTE
 - elf.h, [253](#)
- PT_NULL
 - elf.h, [253](#)
- PT_PHDR
 - elf.h, [253](#)
- PT_SHLIB
 - elf.h, [253](#)
- pdrig
 - gpio_device, [121](#)
- QUEUE_DEPTH
 - atadevice_cmdi.h, [611](#)
- quiet
 - profiler.c, [751](#)
- R_386_32
 - elf.h, [253](#)
- R_386_COPY
 - elf.h, [253](#)
- R_386_GLOB_DAT
 - elf.h, [253](#)
- R_386_GOT32
 - elf.h, [253](#)
- R_386_GOTOFF
 - elf.h, [253](#)
- R_386_GOTPC
 - elf.h, [253](#)
- R_386_JMP_SLOT
 - elf.h, [253](#)
- R_386_NONE
 - elf.h, [253](#)
- R_386_NUM
 - elf.h, [253](#)
- R_386_PC32
 - elf.h, [253](#)
- R_386_PLT32
 - elf.h, [253](#)
- R_386_RELATIVE
 - elf.h, [253](#)
- R_68K_16
 - elf.h, [253](#)
- R_68K_32
 - elf.h, [253](#)
- R_68K_8
 - elf.h, [253](#)
- R_68K_COPY
 - elf.h, [253](#)
- R_68K_GLOB_DAT
 - elf.h, [253](#)
- R_68K_GOT16
 - elf.h, [253](#)
- R_68K_GOT16O
 - elf.h, [253](#)
- R_68K_GOT32
 - elf.h, [253](#)
- R_68K_GOT32O
 - elf.h, [253](#)
- R_68K_GOT8

- elf.h, 253
- R_68K_GOT8O
 - elf.h, 253
- R_68K_JMP_SLOT
 - elf.h, 253
- R_68K_NONE
 - elf.h, 253
- R_68K_PC16
 - elf.h, 253
- R_68K_PC32
 - elf.h, 253
- R_68K_PC8
 - elf.h, 253
- R_68K_PLT16
 - elf.h, 253
- R_68K_PLT16O
 - elf.h, 253
- R_68K_PLT32
 - elf.h, 253
- R_68K_PLT32O
 - elf.h, 253
- R_68K_PLT8
 - elf.h, 253
- R_68K_PLT8O
 - elf.h, 253
- R_68K_RELATIVE
 - elf.h, 253
- r_addend
 - elf32_rela, 93
 - elf64_rela, 102
- r_info
 - elf32_rel, 92
 - elf32_rela, 93
 - elf64_rel, 101
 - elf64_rela, 102
- r_offset
 - elf32_rel, 92
 - elf32_rela, 93
 - elf64_rel, 101
 - elf64_rela, 102
- r_symndx
 - COFF_reloc, 45
- r_type
 - COFF_reloc, 45
- r_vaddr
 - COFF_reloc, 45
- raddr
 - stack_struct, 170
- random_seed
 - mem_config, 147
- range
 - raw_stats, 156
- range_cache
 - or32.c, 432
- RAW_RANGE
 - stats.c, 275
- raw_stats, 156
 - range, 156
 - reg, 156
 - stats.c, 276
- read
 - channel_ops, 37
- READ_BUFFER
 - atacmd.h, 593
- read_dat16
 - mem_ops, 149
- read_dat32
 - mem_ops, 149
- read_dat8
 - mem_ops, 149
- READ_DMA
 - atacmd.h, 593
- READ_DMA_QUEUED
 - atacmd.h, 593
- read_file
 - mprofiler.c, 558
- read_func
 - vapi_handler, 173
- READ_MULTIPLE
 - atacmd.h, 593
- READ_NATIVE_MAX_ADDRESS
 - atacmd.h, 593
- read_packet
 - vapi.c, 841
- read_script_file
 - sim-config.c, 775
- READ_SECTOR
 - atacmd.h, 593
- READ_SECTORS
 - atacmd.h, 593
- read_up
 - config, 52
- READ_VERIFY_SECTOR
 - atacmd.h, 593
- READ_VERIFY_SECTORS
 - atacmd.h, 593
- readfile_coff
 - parse.c, 268
- readfile_elf
 - parse.c, 268
- readfunc16
 - mem_ops, 149
- readfunc32
 - mem_ops, 149
- readfunc8
 - mem_ops, 149
- readhit
 - cachestats_entry, 34

- readmiss
 - cachestats_entry, 34
- readsyms_coff
 - parse.c, 269
- rebuild_fds
 - vapi.c, 841
- rec_i386.h
 - get_sp, 433
- rec_stack_base
 - dyn_rec.h, 351
- recalc_cnts
 - cuc.h, 475
 - timings.c, 505
- recalc_last_used_reg
 - bb.c, 442
- RECED_PAGE_ENLARGE_BY
 - dyn_rec.c, 330
- receiveing
 - dev_16450, 71
- recheck_immu
 - dyn_rec.c, 344
 - dyn_rec.h, 350
- RECOMMENDED_MWDMA_CYCLE_TIME
 - atadevice_cmdi.h, 611
- recompile_delay_insn
 - dyn_rec.c, 344
- recompile_insn
 - dyn_rec.c, 344
- recompile_page
 - dyn_rec.c, 344
 - dyn_rec.h, 350
- recv_break
 - dev_16450, 71
- reenter_int
 - sim-cmd.c, 759
- REF
 - cuc.h, 465
- ref
 - _csm_list, 12
 - _dep_list_t, 15
 - cuc_shared_item, 64
- REF_BB
 - cuc.h, 465
- REF_I
 - cuc.h, 465
- referenced
 - dma_channel, 78
- refresh
 - fb_state, 115
- refresh_count
 - fb_state, 115
- REFRESH_DIVIDER
 - fb.c, 685
- refresh_rate
 - fb_state, 115
 - vga_state, 175
- reg
 - cpu_state, 57
 - raw_stats, 156
- reg_ata_sec
 - atahost.c, 621
 - atahost.h, 627
- reg_bpb_sec
 - branch-predict.c, 180
 - branch-predict.h, 183
- REG_C
 - arch.h, 284
- reg_config_param
 - sim-config.c, 775
 - sim-config.h, 783
- reg_config_sec
 - sim-config.c, 775
 - sim-config.h, 784
- reg_config_secs
 - sim-config.c, 777
 - sim-config.h, 786
- reg_cpu_sec
 - cpu-config.c, 202
 - cpu-config.h, 204
- reg_cuc_sec
 - cuc.c, 458
 - cuc.h, 475
- reg_dc_sec
 - dcache-model.c, 187
 - dcache-model.h, 191
- reg_debug_sec
 - debug-unit.c, 517
 - debug-unit.h, 522
- reg_dep
 - bb.c, 442
 - cuc.h, 475
- reg_dep_rec
 - bb.c, 442
- reg_dma_sec
 - dma.c, 663
 - dma.h, 665
- reg_dmmu_sec
 - dmmu.c, 543
 - dmmu.h, 546
- reg_ethernet_sec
 - eth.c, 680
 - eth.h, 682
- reg_fb_sec
 - fb.c, 688
 - fb.h, 690
- reg_generic_sec
 - generic.c, 696
 - generic.h, 699

- reg_gpio_sec
 - gpio.c, 707
 - gpio.h, 709
- reg_ic_sec
 - icache-model.c, 196
 - icache-model.h, 198
- reg_immu_sec
 - immu.c, 552
 - immu.h, 554
- reg_kbd_sec
 - ps2kbd.c, 727
 - ps2kbd.h, 729
- reg_mc_sec
 - mc.c, 716
 - mc.h, 719
- reg_mem_area
 - abstract.c, 214
 - abstract.h, 230
- reg_memory_sec
 - memory.h, 721
 - peripheral/memory.c, 501
- reg_pic_sec
 - pic.c, 739
 - pic.h, 741
- reg_pm_sec
 - pm.c, 743
 - pm.h, 745
- reg_sim_reset
 - toplevel-support.c, 827
 - toplevel-support.h, 832
- reg_sim_sec
 - sim-config.c, 778
- reg_sim_stat
 - sim-cmd.c, 759
 - sim-cmd.h, 771
- reg_t
 - op_queue, 154
- reg_uart_sec
 - 16450.c, 573
 - 16450.h, 588
- reg_vapi_sec
 - vapi.c, 841
 - vapi.h, 846
- reg_vga_sec
 - vga.c, 732
 - vga.h, 736
- register_memoryarea_mask
 - abstract.c, 214
- regs
 - ata_device, 20
 - ata_host, 24
 - dev_16450, 71
 - dma_channel, 78
 - dma_controller, 81
 - eth_device, 110
- reloc, 157
 - addend, 157
 - cuc.h, 477
 - func_offset, 157
 - load.c, 494
 - name, 157
 - type, 157
- relocate_bb
 - bb.c, 442
- remove_breakpoint
 - labels.c, 262
 - labels.h, 264
- remove_dead
 - cuc.h, 476
 - insn.c, 482
- remove_dead_bb
 - bb.c, 442
 - cuc.h, 476
- remove_dslots
 - load.c, 493
- remove_nops
 - cuc.h, 476
 - insn.c, 482
- remove_trivial_regs
 - cuc.h, 476
 - insn.c, 483
- report_interrupt
 - pic.c, 740
 - pic.h, 742
- reserved1
 - BMP_HEADER, 27
- reserved2
 - BMP_HEADER, 27
- reset_cycles
 - runtime, 160
- reset_hook
 - sim_reset_hook, 165
- reset_instructions
 - runtime, 160
- RET_OPCODE
 - dyngen_i386.c, 358
- ret_spr
 - spr-dump.c, 302
- rev
 - ata_host, 24
- rfds
 - eth_device, 110
- RGPIO_AUX
 - gpio.c, 703
- RGPIO_CTRL
 - gpio.c, 703
- RGPIO_CTRL_ECLK
 - gpio.c, 703

- RGPIO_CTRL_INTE
 - gpio.c, 703
- RGPIO_CTRL_INTS
 - gpio.c, 703
- RGPIO_CTRL_NEC
 - gpio.c, 703
- RGPIO_IN
 - gpio.c, 703
- RGPIO_INTE
 - gpio.c, 703
- RGPIO_INTS
 - gpio.c, 703
- RGPIO_OE
 - gpio.c, 703
- RGPIO_OUT
 - gpio.c, 703
- RGPIO_PTRIG
 - gpio.c, 703
- RISCOP_RESET
 - debug-unit.c, 511
- RISCOP_STALL
 - debug-unit.c, 511
- roll_loop
 - bb.c, 442
- rtl
 - cuc_known_insn, 63
- rtx_sock
 - eth_device, 110
- rtx_type
 - eth_device, 110
- run_sched_out_of_line
 - dyn_rec.c, 345
 - dyn_rec.h, 350
- runtime, 158
 - cpu, 160
 - cuc, 160
 - cycle_duration, 160
 - cycles, 160
 - enabled, 160
 - end_cycles, 160
 - ext_int, 160
 - fexe_log, 160
 - filename, 160
 - fmprof, 160
 - fout, 160
 - fprof, 160
 - hazardwait, 160
 - hush, 160
 - init, 160
 - instructions, 160
 - iprompt, 160
 - iprompt_run, 160
 - loadcycles, 160
 - mdelay, 160
 - mem_cycles, 160
 - reset_cycles, 160
 - reset_instructions, 160
 - server_port, 160
 - sim, 160
 - sim-config.c, 780
 - sim-config.h, 787
 - stalled, 160
 - storecycles, 160
 - supercycles, 160
 - time_point, 160
 - vapi, 160
 - vapi_file, 160
- rx
 - eth_device, 110
- rx_buff
 - eth_device, 110
- rx_channel
 - eth_device, 110
- rxb
 - ata_host, 24
- rxbuf
 - dev_16450, 71
- rxbuf_full
 - dev_16450, 71
- rxbuf_head
 - dev_16450, 71
- rxbuf_tail
 - dev_16450, 71
- rxfd
 - eth_device, 110
- rxfile
 - eth_device, 110
 - kbd_state, 141
- rxfs
 - kbd_state, 141
- rxser
 - dev_16450, 71
- s_flags
 - COFF_scnhdr, 46
- S_FUNC
 - op.c, 394
- s_innoptr
 - COFF_scnhdr, 46
- s_name
 - COFF_scnhdr, 46
- s_nlnno
 - COFF_scnhdr, 46
- s_nreloc
 - COFF_scnhdr, 46
- S_OP_NAME
 - op.c, 394
- s_paddr

- COFF_scnhdr, 46
- s_relptr
 - COFF_scnhdr, 46
- s_scnptr
 - COFF_scnhdr, 46
- s_size
 - COFF_scnhdr, 46
- s_vaddr
 - COFF_scnhdr, 46
- same_transfers
 - cuc/memory.c, 496
- save_t_bound
 - op.c, 401
- saved_regs
 - _cuc_func, 14
- sbp_bf_fwd
 - config, 52
- sbp_bnf_fwd
 - config, 52
- sbuf_buf
 - execute.c, 369
- sbuf_count
 - execute.c, 369
- sbuf_head
 - execute.c, 369
- sbuf_len
 - config, 52
- sbuf_load
 - execute.c, 367
- sbuf_prev_cycles
 - execute.c, 369
- sbuf_store
 - execute.c, 367
- sbuf_tail
 - execute.c, 369
- sbuf_total_cyc
 - execute.c, 369
 - execute.h, 260
- sbuf_wait_cyc
 - execute.c, 369
 - execute.h, 260
- scan_decode
 - ps2kbd.c, 728
- scan_table
 - ps2kbd.c, 728
- sched.c
 - DECLARE_DEBUG_CHANNEL, 806
 - DEFAULT_DEBUG_CHANNEL, 806
 - do_scheduler, 806
 - sched_add, 806
 - sched_find_remove, 806
 - sched_guard, 806
 - SCHED_HEAP_SIZE, 806
 - sched_init, 806
 - sched_next_insn, 806
 - sched_print_jobs, 806
 - sched_reset, 806
 - SCHED_TIME_MAX, 806
 - sched_timer_job
 - tick.c, 814
 - schedule_memory
 - cuc.h, 476
 - cuc/memory.c, 496
- sched.h
 - do_scheduler, 808
 - SCHED_ADD, 807
 - sched_add, 808
 - SCHED_FIND_REMOVE, 807
 - sched_find_remove, 808
 - sched_init, 808
 - sched_next_insn, 808
 - sched_reset, 808
 - scheduler, 808
- SCHED_ADD
 - sched.h, 807
- sched_add
 - sched.c, 806
 - sched.h, 808
- sched_entry, 162
 - func, 162
 - next, 162
 - param, 162
 - time, 162
- SCHED_FIND_REMOVE
 - sched.h, 807
- sched_find_remove
 - sched.c, 806
 - sched.h, 808
- sched_guard
 - sched.c, 806
- SCHED_HEAP_SIZE
 - sched.c, 806
- sched_i386.h
 - set_sched_cycle, 434
- sched_init
 - sched.c, 806
 - sched.h, 808
- sched_next_insn
 - sched.c, 806
 - sched.h, 808
- sched_print_jobs
 - sched.c, 806
- sched_reset
 - sched.c, 806
 - sched.h, 808
- SCHED_TIME_MAX
 - sched.c, 806
- sched_timer_job
 - tick.c, 814
- schedule_memory
 - cuc.h, 476
 - cuc/memory.c, 496

- scheduler
 - sched.c, 806
 - sched.h, 808
- scheduler_struct, 163
 - free_job_queue, 163
 - job_queue, 163
- scr
 - dev_16450, 71
- SD
 - stats.c, 275
- search_csm
 - insn.c, 483
- sec_end
 - config_section, 56
- sec_start
 - config_section, 56
- sections
 - sim-config.c, 780
- sector_count
 - ata_device, 20
- sector_number
 - ata_device, 20
- sectors
 - ata_device, 20
- sectors_per_track
 - ata_device, 20
- SECURITY_DISABLE_PASSWORD
 - atacmd.h, 593
- SECURITY_ERASE_PREPARE
 - atacmd.h, 593
- SECURITY_ERASE_UNIT
 - atacmd.h, 593
- SECURITY_FREEZE_LOCK
 - atacmd.h, 593
- SECURITY_SET_PASSWORD
 - atacmd.h, 593
- SECURITY_UNLOCK
 - atacmd.h, 593
- SEEK
 - atacmd.h, 593
- selected_tim
 - cuc_bb, 60
- SELMAG
 - elf.h, 253
- send_char
 - 16450.c, 574
- server_fd
 - gdbcomm.c, 529
 - vapi.c, 844
- server_ip
 - gdbcomm.c, 529
- server_port
 - config, 52
 - gdbcomm.c, 529
- runtime, 160
- server_request
 - vapi.c, 841
- serverIP
 - vapi.c, 844
- SERVICE
 - atacmd.h, 593
- set_config
 - sim-config.c, 778
- set_config_command
 - sim-config.c, 779
 - sim-config.h, 787
- set_csc_tms
 - mc.c, 717
- set_devint_reg
 - debug-unit.c, 517, 518
- set_direct16
 - abstract.c, 214
 - abstract.h, 231
- set_direct32
 - abstract.c, 215
 - abstract.h, 231
- set_direct8
 - abstract.c, 215
 - abstract.h, 232
- set_dma_nd_i
 - dma.c, 663
 - dma.h, 665
- set_dma_req_i
 - dma.c, 663
 - dma.h, 665
- SET_FEATURES
 - atacmd.h, 593
- SET_FEATURES_REQUIRED_AFTER_-
POWER_UP
 - atadevice_cmdi.h, 611
- SET_FIELD
 - fields.h, 692
- SET_FLAG
 - fields.h, 692
- set_func_deps
 - cuc.c, 458
- set_insbrkpoint
 - trace.c, 279
 - trace.h, 281
- set_io
 - cuc.h, 477
 - insn.c, 483
- set_mask
 - dmmu, 83
 - ic, 124
 - immu, 126
- SET_MAX
 - atacmd.h, 593

- SET_MAX_ADDRESS
 - atacmd.h, 593
- SET_MAX_FREEZE_LOCK
 - atacmd.h, 593
- SET_MAX_LOCK
 - atacmd.h, 593
- SET_MAX_SET_PASSWORD
 - atacmd.h, 593
- SET_MAX_UNLOCK
 - atacmd.h, 593
- set_mem16
 - abstract.c, 216
 - abstract.h, 232
- set_mem32
 - abstract.c, 216
 - abstract.h, 233
- set_mem8
 - abstract.c, 217
 - abstract.h, 233
- set_mem_16_inv
 - abstract.c, 217
- set_mem_16_inv_direct
 - abstract.c, 218
- set_mem_32_inv
 - abstract.c, 218
- set_mem_32_inv_direct
 - abstract.c, 218
- set_mem_8_inv
 - abstract.c, 219
- set_mem_8_inv_direct
 - abstract.c, 219
- set_mem_valid
 - abstract.c, 219
 - abstract.h, 234
- SET_MULTIPLE_MODE
 - atacmd.h, 593
- SET_OV_FLAG_FN
 - execute.c, 361
- set_pc
 - common_i386.h, 319
- set_pc_delay_gpr
 - dyn_rec.c, 346
- set_program32
 - abstract.c, 219
 - abstract.h, 234
- set_program8
 - abstract.c, 219
 - abstract.h, 234
- set_sched_cycle
 - sched_i386.h, 434
- set_stall_state
 - debug-unit.c, 518
 - debug-unit.h, 523
- SET_TRANSFER_MODE_SECTOR_COUNT_-REG
 - atacmd.h, 593
- setsim_mem16
 - abstract.c, 219
 - abstract.h, 234
- setsim_mem32
 - abstract.c, 220
 - abstract.h, 235
- setsim_mem8
 - abstract.c, 220
 - abstract.h, 235
- setsim_reg
 - execute.c, 368
 - execute.h, 259
- sh_addr
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_addralign
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_entsize
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_flags
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_info
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_link
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_name
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_offset
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_size
 - elf32_shdr, 94
 - elf64_shdr, 103
- sh_type
 - elf32_shdr, 94
 - elf64_shdr, 103
- shared
 - cuc_timings, 66
- SHF_ALLOC
 - elf.h, 253
- SHF_EXECINSTR
 - elf.h, 253
- SHF_MASKPROC
 - elf.h, 253
- SHF_WRITE

- elf.h, 253
- shift
 - ps2kbd.c, 728
- shift_fprintf
 - generate.c, 372
- ship_gprs_out_t
 - dyn_rec.c, 345
- ship_t_out
 - dyn_rec.c, 346
- SHN_ABS
 - elf.h, 253
- SHN_COMMON
 - elf.h, 253
- SHN_HIPROC
 - elf.h, 253
- SHN_HIRESERVE
 - elf.h, 253
- SHN_LOPROC
 - elf.h, 253
- SHN_LORESERVE
 - elf.h, 253
- SHN_UNDEF
 - elf.h, 253
- SHT_DYNAMIC
 - elf.h, 253
- SHT_DYNSYM
 - elf.h, 253
- SHT_HASH
 - elf.h, 253
- SHT_HIPROC
 - elf.h, 253
- SHT_HIUSER
 - elf.h, 253
- SHT_LOPROC
 - elf.h, 253
- SHT_LOUSER
 - elf.h, 253
- SHT_NOBITS
 - elf.h, 253
- SHT_NOTE
 - elf.h, 253
- SHT_NULL
 - elf.h, 253
- SHT_NUM
 - elf.h, 253
- SHT_PROGBITS
 - elf.h, 253
- SHT_REL
 - elf.h, 253
- SHT_RELA
 - elf.h, 253
- SHT_SHLIB
 - elf.h, 253
- SHT_STRTAB
 - elf.h, 253
- SHT_SYMTAB
 - elf.h, 253
- sigs
 - ata_device, 20
- sigsegv_addr
 - dyn_rec.c, 346
- sigsegv_state
 - dyn_rec.c, 346
- sim
 - config, 52
 - runtime, 160
- sim-cmd.c, 755
 - check_insn_exec, 757
 - handle_sim_command, 757
 - print_insn_exec, 758
 - reenter_int, 759
 - reg_sim_stat, 759
 - sim_cmd_break, 760
 - sim_cmd_breaks, 760
 - sim_cmd_cm, 760
 - sim_cmd_cuc, 760
 - sim_cmd_de, 763
 - sim_cmd_debug, 763
 - sim_cmd_dh, 763
 - sim_cmd_dm, 763
 - sim_cmd_dv, 764
 - sim_cmd_help, 764
 - sim_cmd_hist, 764
 - sim_cmd_info, 765
 - sim_cmd_mprofile, 765
 - sim_cmd_pc, 765
 - sim_cmd_pm, 766
 - sim_cmd_pr, 766
 - sim_cmd_profile, 766
 - sim_cmd_quit, 766
 - sim_cmd_r, 767
 - sim_cmd_reset, 767
 - sim_cmd_run, 767
 - sim_cmd_set, 768
 - sim_cmd_setdbch, 768
 - sim_cmd_stall, 768
 - sim_cmd_stats, 768
 - sim_cmd_trace, 769
 - sim_cmd_unstall, 769
 - sim_commands, 770
 - sim_stats, 770
 - strip_space, 769
 - to_insn_num, 770
- sim-cmd.h, 771
 - handle_sim_command, 771
 - reg_sim_stat, 771
- sim-config.c, 772
 - base_include, 774

- config, 780
- cur_section, 780
- DEFAULT_DEBUG_CHANNEL, 774
- get_paramt_str, 774
- init_defconfig, 774
- MERROR, 774
- parse_args, 774
- print_config, 775
- read_script_file, 775
- reg_config_param, 775
- reg_config_sec, 775
- reg_config_secs, 777
- reg_sim_sec, 778
- runtime, 780
- sections, 780
- set_config, 778
- set_config_command, 779
- sim_clkcycle, 779
- sim_debug, 779
- sim_exe_log, 779
- sim_exe_log_end, 779
- sim_exe_log_fn, 779
- sim_exe_log_marker, 779
- sim_exe_log_start, 779
- sim_exe_log_type, 779
- sim_history, 779
- sim_mprof_fn, 780
- sim_mprofile, 780
- sim_prof_fn, 780
- sim_profile, 780
- sim_verbose, 780
- switch_param, 780
- WARNING, 774
- sim-config.h, 781
 - CHECK_INT_TIME, 782
 - config, 787
 - CONFIG_ERROR, 782
 - cur_section, 787
 - do_stats, 787
 - EXE_LOG_HARDWARE, 782
 - EXE_LOG_SIMPLE, 782
 - EXE_LOG_SOFTWARE, 782
 - init_defconfig, 783
 - MAX_SBUF_LEN, 782
 - param_t, 782
 - paramt_addr, 782
 - paramt_int, 782
 - paramt_longlong, 782
 - paramt_none, 782
 - paramt_str, 782
 - paramt_word, 782
 - parse_args, 783
 - print_config, 783
 - PRINTF, 782
 - reg_config_param, 783
 - reg_config_sec, 784
 - reg_config_secs, 786
 - runtime, 787
 - set_config_command, 787
 - STR_SIZE, 782
- sim_clkcycle
 - sim-config.c, 779
- sim_cmd_break
 - sim-cmd.c, 760
- sim_cmd_breaks
 - sim-cmd.c, 760
- sim_cmd_cm
 - sim-cmd.c, 760
- sim_cmd_cuc
 - sim-cmd.c, 760
- sim_cmd_de
 - sim-cmd.c, 763
- sim_cmd_debug
 - sim-cmd.c, 763
- sim_cmd_dh
 - sim-cmd.c, 763
- sim_cmd_dm
 - sim-cmd.c, 763
- sim_cmd_dv
 - sim-cmd.c, 764
- sim_cmd_help
 - sim-cmd.c, 764
- sim_cmd_hist
 - sim-cmd.c, 764
- sim_cmd_info
 - sim-cmd.c, 765
- sim_cmd_mprofile
 - sim-cmd.c, 765
- sim_cmd_pc
 - sim-cmd.c, 765
- sim_cmd_pm
 - sim-cmd.c, 766
- sim_cmd_pr
 - sim-cmd.c, 766
- sim_cmd_profile
 - sim-cmd.c, 766
- sim_cmd_quit
 - sim-cmd.c, 766
- sim_cmd_r
 - sim-cmd.c, 767
- sim_cmd_reset
 - sim-cmd.c, 767
- sim_cmd_run
 - sim-cmd.c, 767
- sim_cmd_set
 - sim-cmd.c, 768
- sim_cmd_setdbch
 - sim-cmd.c, 768

- sim_cmd_stall
 - sim-cmd.c, 768
- sim_cmd_stats
 - sim-cmd.c, 768
- sim_cmd_trace
 - sim-cmd.c, 769
- sim_cmd_unstall
 - sim-cmd.c, 769
- sim_command, 164
 - cmd_handle, 164
 - name, 164
- sim_commands
 - sim-cmd.c, 770
- sim_debug
 - sim-config.c, 779
- sim_done
 - toplevel-support.c, 828
 - toplevel-support.h, 833
- sim_exe_log
 - sim-config.c, 779
- sim_exe_log_end
 - sim-config.c, 779
- sim_exe_log_fn
 - sim-config.c, 779
- sim_exe_log_marker
 - sim-config.c, 779
- sim_exe_log_start
 - sim-config.c, 779
- sim_exe_log_type
 - sim-config.c, 779
- sim_history
 - sim-config.c, 779
- sim_init
 - toplevel-support.c, 828
 - toplevel-support.h, 833
- sim_mprof_fn
 - sim-config.c, 780
- sim_mprofile
 - sim-config.c, 780
- sim_prof_fn
 - sim-config.c, 780
- sim_profile
 - sim-config.c, 780
- sim_reset
 - toplevel-support.c, 829
 - toplevel-support.h, 834
- sim_reset_hook, 165
 - dat, 165
 - next, 165
 - reset_hook, 165
- sim_reset_hooks
 - toplevel-support.c, 830
- sim_stat, 166
 - dat, 166
 - next, 166
 - stat_func, 166
- sim_stats
 - sim-cmd.c, 770
- sim_verbose
 - sim-config.c, 780
- simgetstr
 - simprintf.c, 810
- simmem_read16
 - peripheral/memory.c, 501
- simmem_read32
 - peripheral/memory.c, 502
- simmem_read8
 - peripheral/memory.c, 502
- simmem_read_zero16
 - peripheral/memory.c, 502
- simmem_read_zero32
 - peripheral/memory.c, 502
- simmem_read_zero8
 - peripheral/memory.c, 502
- simmem_write16
 - peripheral/memory.c, 502
- simmem_write32
 - peripheral/memory.c, 502
- simmem_write8
 - peripheral/memory.c, 502
- simmem_write_null16
 - peripheral/memory.c, 502
- simmem_write_null32
 - peripheral/memory.c, 502
- simmem_write_null8
 - peripheral/memory.c, 502
- simpl32_defs.h
 - PARAMS, 435
- simplify_bb
 - bb.c, 442
- simprintf
 - simprintf.c, 810
 - simprintf.h, 812
- simprintf.c
 - DEFAULT_DEBUG_CHANNEL, 810
 - FMTLEN, 810
 - fmtstr, 810
 - simgetstr, 810
 - simprintf, 810
 - STACK_ARGS, 810
- simprintf.h
 - simprintf, 812
- size
 - _csm_list, 12
 - ata_device, 20
 - BMP_HEADER, 27
 - cuc_timing_table, 65
 - cuc_timings, 66

- dev_generic, 74
- dev_memarea, 76
- INFOHEADER, 128
- mem_config, 147
- size_mask
 - dev_memarea, 76
- size_sect
 - ata_device, 20
- sizei
 - cuc_timing_table, 65
- skew
 - dev_16450, 71
- sl_entz
 - COFF_slib, 47
- sl_pathndx
 - COFF_slib, 47
- SLEEP
 - atacmd.h, 593
- slowdown
 - kbd_state, 141
- SMART
 - atacmd.h, 593
- SMART_ATTRIBITE_AUTOSAVE
 - atacmd.h, 593
- SMART_DISABLE_OPERATIONS
 - atacmd.h, 593
- SMART_ENABLE_OPERATIONS
 - atacmd.h, 593
- SMART_EXECUTE_OFFLINE_IMMEDIATE
 - atacmd.h, 593
- SMART_READ_DATA
 - atacmd.h, 593
- SMART_READ_LOG
 - atacmd.h, 593
- SMART_RETURN_STATUS
 - atacmd.h, 593
- SMART_SAVE_ATTRIBUTE_VALUES
 - atacmd.h, 593
- SMART_WRITE_LOG
 - atacmd.h, 593
- socket_fd
 - tcp_channel, 171
- sockif
 - eth_device, 110
- source
 - dma_channel, 78
- source_mask
 - dma_channel, 78
- SPEEDY_CALL
 - op_i386.h, 412
- spr-defs.h
 - MATCHPOINTS_TO_NDP, 298
 - MAX_GRPs, 298
 - MAX_MATCHPOINTS, 300
 - MAX_SPRs, 300
 - MAX_SPRs_PER_GRP, 300
 - MAX_SPRs_PER_GRP_BITS, 300
 - MAX_WATCHPOINTS, 300
 - NOP_CNT_RESET, 300
 - NOP_EXIT, 300
 - NOP_NOP, 300
 - NOP_PRINTF, 300
 - NOP_PUTC, 300
 - NOP_REPORT, 300
 - NOP_REPORT_FIRST, 300
 - NOP_REPORT_LAST, 300
 - SPR_CPUCFGR, 300
 - SPR_CPUCFGR_CGF, 300
 - SPR_CPUCFGR_NSGF, 300
 - SPR_CPUCFGR_OB32S, 300
 - SPR_CPUCFGR_OB64S, 300
 - SPR_CPUCFGR_OF32S, 300
 - SPR_CPUCFGR_OF64S, 300
 - SPR_CPUCFGR_OV64S, 300
 - SPR_CPUCFGR_RES, 300
 - SPR_DCBFR, 300
 - SPR_DCBIR, 300
 - SPR_DCBLR, 300
 - SPR_DCBPR, 300
 - SPR_DCBWR, 300
 - SPR_DCCFGR, 300
 - SPR_DCCFGR_CBFRI, 300
 - SPR_DCCFGR_CBIRI, 300
 - SPR_DCCFGR_CBLRI, 300
 - SPR_DCCFGR_CBPRI, 300
 - SPR_DCCFGR_CBS, 300
 - SPR_DCCFGR_CBS_OFF, 300
 - SPR_DCCFGR_CBWBRI, 300
 - SPR_DCCFGR_CCRI, 300
 - SPR_DCCFGR_CWS, 300
 - SPR_DCCFGR_NCS, 300
 - SPR_DCCFGR_NCS_OFF, 300
 - SPR_DCCFGR_NCW, 300
 - SPR_DCCFGR_NCW_OFF, 300
 - SPR_DCCR, 300
 - SPR_DCCR_EW, 300
 - SPR_DCFGR, 300
 - SPR_DCFGR_NDP, 300
 - SPR_DCFGR_NDP1, 300
 - SPR_DCFGR_NDP2, 300
 - SPR_DCFGR_NDP3, 300
 - SPR_DCFGR_NDP4, 300
 - SPR_DCFGR_NDP5, 300
 - SPR_DCFGR_NDP6, 300
 - SPR_DCFGR_NDP7, 300
 - SPR_DCFGR_NDP8, 300
 - SPR_DCFGR_WPCI, 300
 - SPR_DCR, 300

SPR_DCR_BASE, 300
SPR_DCR_CC, 300
SPR_DCR_CC_EQUAL, 300
SPR_DCR_CC_GREAT, 300
SPR_DCR_CC_GREATER, 300
SPR_DCR_CC_LESS, 300
SPR_DCR_CC_LESSE, 300
SPR_DCR_CC_MASKED, 300
SPR_DCR_CC_NEQUAL, 300
SPR_DCR_CT, 300
SPR_DCR_CT_DISABLED, 300
SPR_DCR_CT_IFEA, 300
SPR_DCR_CT_LD, 300
SPR_DCR_CT_LEA, 300
SPR_DCR_CT_LSD, 300
SPR_DCR_CT_LSEA, 300
SPR_DCR_CT_SD, 300
SPR_DCR_CT_SEA, 300
SPR_DCR_DP, 300
SPR_DCR_LAST, 300
SPR_DCR_SC, 300
SPR_DMMUCFGR, 300
SPR_DMMUCFGR_CRI, 300
SPR_DMMUCFGR_HTR, 300
SPR_DMMUCFGR_NAE, 300
SPR_DMMUCFGR_NTS, 300
SPR_DMMUCFGR_NTS_OFF, 300
SPR_DMMUCFGR_NTW, 300
SPR_DMMUCFGR_NTW_OFF, 300
SPR_DMMUCFGR_PRI, 300
SPR_DMMUCFGR_TEIRI, 300
SPR_DMMUCR, 300
SPR_DMMUCR_P1S, 300
SPR_DMMUCR_P2S, 300
SPR_DMMUCR_PADDR_WIDTH, 300
SPR_DMMUCR_VADDR_WIDTH, 300
SPR_DMR1, 300
SPR_DMR1_BT, 300
SPR_DMR1_CW, 300
SPR_DMR1_CW0, 300
SPR_DMR1_CW0_AND, 300
SPR_DMR1_CW0_OR, 300
SPR_DMR1_CW1, 300
SPR_DMR1_CW1_AND, 300
SPR_DMR1_CW1_OR, 300
SPR_DMR1_CW2, 300
SPR_DMR1_CW2_AND, 300
SPR_DMR1_CW2_OR, 300
SPR_DMR1_CW3, 300
SPR_DMR1_CW3_AND, 300
SPR_DMR1_CW3_OR, 300
SPR_DMR1_CW4, 300
SPR_DMR1_CW4_AND, 300
SPR_DMR1_CW4_OR, 300
SPR_DMR1_CW5, 300
SPR_DMR1_CW5_AND, 300
SPR_DMR1_CW5_OR, 300
SPR_DMR1_CW6, 300
SPR_DMR1_CW6_AND, 300
SPR_DMR1_CW6_OR, 300
SPR_DMR1_CW7, 300
SPR_DMR1_CW7_AND, 300
SPR_DMR1_CW7_OR, 300
SPR_DMR1_CW8, 300
SPR_DMR1_CW8_AND, 300
SPR_DMR1_CW8_OR, 300
SPR_DMR1_CW9, 300
SPR_DMR1_CW9_AND, 300
SPR_DMR1_CW9_OR, 300
SPR_DMR1_RES1, 300
SPR_DMR1_RES2, 300
SPR_DMR1_ST, 300
SPR_DMR2, 300
SPR_DMR2_AWTC, 300
SPR_DMR2_AWTC_OFF, 300
SPR_DMR2_WBS, 300
SPR_DMR2_WBS_OFF, 300
SPR_DMR2_WCE0, 300
SPR_DMR2_WCE1, 300
SPR_DMR2_WGB, 300
SPR_DMR2_WGB_OFF, 300
SPR_DRR, 300
SPR_DRR_AE, 300
SPR_DRR_BUSEE, 300
SPR_DRR_DME, 300
SPR_DRR_DPFE, 300
SPR_DRR_IE, 300
SPR_DRR_IIE, 300
SPR_DRR_IME, 300
SPR_DRR_IPFE, 300
SPR_DRR_RE, 300
SPR_DRR_RSTE, 300
SPR_DRR_SCE, 300
SPR_DRR_TE, 300
SPR_DRR_TTE, 300
SPR_DSR, 300
SPR_DSR_AE, 300
SPR_DSR_BUSEE, 300
SPR_DSR_DME, 300
SPR_DSR_DPFE, 300
SPR_DSR_IE, 300
SPR_DSR_IIE, 300
SPR_DSR_IME, 300
SPR_DSR_IPFE, 300
SPR_DSR_RE, 300
SPR_DSR_RSTE, 300
SPR_DSR_SCE, 300
SPR_DSR_SSE, 300

SPR_DSR_TE, 300
 SPR_DSR_TTE, 300
 SPR_DTLBMR_BASE, 300
 SPR_DTLBMR_CID, 300
 SPR_DTLBMR_LAST, 300
 SPR_DTLBMR_LRU, 300
 SPR_DTLBMR_PL1, 300
 SPR_DTLBMR_V, 300
 SPR_DTLBMR_VPN, 300
 SPR_DTLBTR_A, 300
 SPR_DTLBTR_BASE, 300
 SPR_DTLBTR_CC, 300
 SPR_DTLBTR_CI, 300
 SPR_DTLBTR_D, 300
 SPR_DTLBTR_LAST, 300
 SPR_DTLBTR_PPN, 300
 SPR_DTLBTR_SRE, 300
 SPR_DTLBTR_SWE, 300
 SPR_DTLBTR_URE, 300
 SPR_DTLBTR_UWE, 300
 SPR_DTLBTR_WBC, 300
 SPR_DTLBTR_WOM, 300
 SPR_DVR, 300
 SPR_DWCR0, 300
 SPR_DWCR1, 300
 SPR_DWCR_COUNT, 300
 SPR_DWCR_MATCH, 300
 SPR_DWCR_MATCH_OFF, 300
 SPR_EEAR_BASE, 300
 SPR_EEAR_LAST, 300
 SPR_EPCR_BASE, 300
 SPR_EPCR_LAST, 300
 SPR_ESR_BASE, 300
 SPR_ESR_LAST, 300
 SPR_ICBIR, 300
 SPR_ICBLR, 300
 SPR_ICBPR, 300
 SPR_ICCFGR, 300
 SPR_ICCFGR_CBIRI, 300
 SPR_ICCFGR_CBLRI, 300
 SPR_ICCFGR_CBPRI, 300
 SPR_ICCFGR_CBS, 300
 SPR_ICCFGR_CBS_OFF, 300
 SPR_ICCFGR_CCRI, 300
 SPR_ICCFGR_NCS, 300
 SPR_ICCFGR_NCS_OFF, 300
 SPR_ICCFGR_NCW, 300
 SPR_ICCFGR_NCW_OFF, 300
 SPR_ICCR, 300
 SPR_ICCR_EW, 300
 SPR_ICR_BASE, 300
 SPR_ICR_LAST, 300
 SPR_IMMUCFGR, 300
 SPR_IMMUCFGR_CRI, 300
 SPR_IMMUCFGR_HTR, 300
 SPR_IMMUCFGR_NAE, 300
 SPR_IMMUCFGR_NTS, 300
 SPR_IMMUCFGR_NTS_OFF, 300
 SPR_IMMUCFGR_NTW, 300
 SPR_IMMUCFGR_NTW_OFF, 300
 SPR_IMMUCFGR_PRI, 300
 SPR_IMMUCFGR_TEIRI, 300
 SPR_IMMUCR, 300
 SPR_IMMUCR_P1S, 300
 SPR_IMMUCR_P2S, 300
 SPR_IMMUCR_PADDR_WIDTH, 300
 SPR_IMMUCR_VADDR_WIDTH, 300
 SPR_ITLBMR_BASE, 300
 SPR_ITLBMR_CID, 300
 SPR_ITLBMR_LAST, 300
 SPR_ITLBMR_LRU, 300
 SPR_ITLBMR_PL1, 300
 SPR_ITLBMR_V, 300
 SPR_ITLBMR_VPN, 300
 SPR_ITLBTR_A, 300
 SPR_ITLBTR_BASE, 300
 SPR_ITLBTR_CC, 300
 SPR_ITLBTR_CI, 300
 SPR_ITLBTR_D, 300
 SPR_ITLBTR_LAST, 300
 SPR_ITLBTR_PPN, 300
 SPR_ITLBTR_SXE, 300
 SPR_ITLBTR_UXE, 300
 SPR_ITLBTR_WBC, 300
 SPR_ITLBTR_WOM, 300
 SPR_MACHI, 300
 SPR_MACLO, 300
 SPR_NPC, 300
 SPR_PCCFGR, 300
 SPR_PCCR, 300
 SPR_PCMR, 300
 SPR_PCMR_BS, 300
 SPR_PCMR_CISM, 300
 SPR_PCMR_CIUM, 300
 SPR_PCMR_CP, 300
 SPR_PCMR_DCM, 300
 SPR_PCMR_DDS, 300
 SPR_PCMR_DTLBM, 300
 SPR_PCMR_ICM, 300
 SPR_PCMR_IF, 300
 SPR_PCMR_IFS, 300
 SPR_PCMR_ITLBM, 300
 SPR_PCMR_LA, 300
 SPR_PCMR_LSUS, 300
 SPR_PCMR_SA, 300
 SPR_PCMR_UMRA, 300
 SPR_PCMR_WPE, 300
 SPR_PICMR, 300

- SPR_PICMR_IUM, 300
- SPR_PICPR, 300
- SPR_PICPR_IPRIO, 300
- SPR_PICSR, 300
- SPR_PICSR_IS, 300
- SPR_PMR, 300
- SPR_PMR_DCGE, 300
- SPR_PMR_DME, 300
- SPR_PMR_SDF, 300
- SPR_PMR_SME, 300
- SPR_PMR_SUME, 300
- SPR_PPC, 300
- SPR_SR, 300
- SPR_SR_CE, 300
- SPR_SR_CID, 300
- SPR_SR_CY, 300
- SPR_SR_DCE, 300
- SPR_SR_DME, 300
- SPR_SR_DSX, 300
- SPR_SR_EPH, 300
- SPR_SR_F, 300
- SPR_SR_FO, 300
- SPR_SR_ICE, 300
- SPR_SR_IEE, 300
- SPR_SR_IME, 300
- SPR_SR_LEE, 300
- SPR_SR_OV, 300
- SPR_SR_OVE, 300
- SPR_SR_RES, 300
- SPR_SR_SM, 300
- SPR_SR_SUMRA, 300
- SPR_SR_TEE, 300
- SPR_TTCR, 300
- SPR_TTCR_PERIOD, 300
- SPR_TTMR, 300
- SPR_TTMR_CR, 300
- SPR_TTMR_IE, 300
- SPR_TTMR_IP, 300
- SPR_TTMR_M, 300
- SPR_TTMR_PERIOD, 300
- SPR_TTMR_RT, 300
- SPR_TTMR_SR, 300
- SPR_UPR, 300
- SPR_UPR_CUP, 300
- SPR_UPR_DCP, 300
- SPR_UPR_DMP, 300
- SPR_UPR_DUP, 300
- SPR_UPR_ICP, 300
- SPR_UPR_IMP, 300
- SPR_UPR_MP, 300
- SPR_UPR_PCUP, 300
- SPR_UPR_PICP, 300
- SPR_UPR_PMP, 300
- SPR_UPR_RES, 300
- SPR_UPR_TTP, 300
- SPR_UPR_UP, 300
- SPR_VR, 300
- SPR_VR_CFG, 300
- SPR_VR_CFG_OFF, 300
- SPR_VR_RES, 300
- SPR_VR_REV, 300
- SPR_VR_REV_OFF, 300
- SPR_VR_VER, 300
- SPR_VR_VER_OFF, 300
- SPRGROUP_D, 300
- SPRGROUP_DC, 300
- SPRGROUP_DMMU, 300
- SPRGROUP_IC, 300
- SPRGROUP_IMMU, 300
- SPRGROUP_MAC, 300
- SPRGROUP_PC, 300
- SPRGROUP_PIC, 300
- SPRGROUP_PM, 300
- SPRGROUP_SYS, 300
- SPRGROUP_TT, 300
- spr-dump.c
 - DEFAULT_DEBUG_CHANNEL, 302
 - dump_spr, 302
 - ret_spr, 302
 - spr_cpucfgr, 302
 - spr_d_group, 303
 - spr_dc_group, 303
 - spr_dccfgr, 303
 - spr_dccr, 303
 - spr_dcfgr, 304
 - spr_dmmu_group, 304
 - spr_dmmucfgr, 304
 - spr_dmmucr, 304
 - spr_dmr1, 305
 - spr_dmr2, 305
 - spr_drr, 305
 - spr_dsr, 305
 - spr_dtlbmr, 306
 - spr_dtlbtr, 306
 - spr_dwcr, 306
 - spr_groups, 306
 - spr_ic_group, 307
 - spr_iccfgr, 307
 - spr_iccr, 307
 - spr_immu_group, 307
 - spr_immucfgr, 308
 - spr_immucr, 308
 - spr_itlbmr, 308
 - spr_itlbtr, 308
 - spr_mac_group, 309
 - spr_one_val, 309
 - spr_pc_group, 309
 - spr_pcmr, 309

- spr_pic_group, 310
- spr_picmr, 310
- spr_pm_group, 310
- spr_pmr, 310
- spr_sr, 310
- spr_sys_group, 311
- spr_tt_group, 311
- spr_ttmr, 311
- spr_upr, 312
- spr_vr, 312
- spr-dump.h
 - dump_spr, 313
- spr_bit_def, 167
 - mask, 167
 - name, 167
- SPR_CPUCFGR
 - spr-defs.h, 300
- spr_cpucfgr
 - spr-dump.c, 302
- SPR_CPUCFGR_CGF
 - spr-defs.h, 300
- SPR_CPUCFGR_NSGF
 - spr-defs.h, 300
- SPR_CPUCFGR_OB32S
 - spr-defs.h, 300
- SPR_CPUCFGR_OB64S
 - spr-defs.h, 300
- SPR_CPUCFGR_OF32S
 - spr-defs.h, 300
- SPR_CPUCFGR_OF64S
 - spr-defs.h, 300
- SPR_CPUCFGR_OV64S
 - spr-defs.h, 300
- SPR_CPUCFGR_RES
 - spr-defs.h, 300
- spr_d_group
 - spr-dump.c, 303
- spr_dc_group
 - spr-dump.c, 303
- SPR_DCBFR
 - spr-defs.h, 300
- SPR_DCBIR
 - spr-defs.h, 300
- SPR_DCBLR
 - spr-defs.h, 300
- SPR_DCBPR
 - spr-defs.h, 300
- SPR_DCBWR
 - spr-defs.h, 300
- SPR_DCCFGR
 - spr-defs.h, 300
- spr_dccfgr
 - spr-dump.c, 303
- SPR_DCCFGR_CBFRI
 - spr-defs.h, 300
- SPR_DCCFGR_CBIRI
 - spr-defs.h, 300
- SPR_DCCFGR_CBLRI
 - spr-defs.h, 300
- SPR_DCCFGR_CBPRI
 - spr-defs.h, 300
- SPR_DCCFGR_CBS
 - spr-defs.h, 300
- SPR_DCCFGR_CBS_OFF
 - spr-defs.h, 300
- SPR_DCCFGR_CBWBRI
 - spr-defs.h, 300
- SPR_DCCFGR_CCRI
 - spr-defs.h, 300
- SPR_DCCFGR_CWS
 - spr-defs.h, 300
- SPR_DCCFGR_NCS
 - spr-defs.h, 300
- SPR_DCCFGR_NCS_OFF
 - spr-defs.h, 300
- SPR_DCCFGR_NCW
 - spr-defs.h, 300
- SPR_DCCFGR_NCW_OFF
 - spr-defs.h, 300
- SPR_DCCR
 - spr-defs.h, 300
- spr_dccr
 - spr-dump.c, 303
- SPR_DCCR_EW
 - spr-defs.h, 300
- SPR_DCFGR
 - spr-defs.h, 300
- spr_dcfgr
 - spr-dump.c, 304
- SPR_DCFGR_NDP
 - spr-defs.h, 300
- SPR_DCFGR_NDP1
 - spr-defs.h, 300
- SPR_DCFGR_NDP2
 - spr-defs.h, 300
- SPR_DCFGR_NDP3
 - spr-defs.h, 300
- SPR_DCFGR_NDP4
 - spr-defs.h, 300
- SPR_DCFGR_NDP5
 - spr-defs.h, 300
- SPR_DCFGR_NDP6
 - spr-defs.h, 300
- SPR_DCFGR_NDP7
 - spr-defs.h, 300
- SPR_DCFGR_NDP8
 - spr-defs.h, 300
- SPR_DCFGR_WPCI

- spr-defs.h, 300
- SPR_DCR
 - spr-defs.h, 300
- SPR_DCR_BASE
 - spr-defs.h, 300
- SPR_DCR_CC
 - spr-defs.h, 300
- SPR_DCR_CC_EQUAL
 - spr-defs.h, 300
- SPR_DCR_CC_GREAT
 - spr-defs.h, 300
- SPR_DCR_CC_GREATE
 - spr-defs.h, 300
- SPR_DCR_CC_LESS
 - spr-defs.h, 300
- SPR_DCR_CC_LESSE
 - spr-defs.h, 300
- SPR_DCR_CC_MASKED
 - spr-defs.h, 300
- SPR_DCR_CC_NEQUAL
 - spr-defs.h, 300
- SPR_DCR_CT
 - spr-defs.h, 300
- SPR_DCR_CT_DISABLED
 - spr-defs.h, 300
- SPR_DCR_CT_IFEA
 - spr-defs.h, 300
- SPR_DCR_CT_LD
 - spr-defs.h, 300
- SPR_DCR_CT_LEA
 - spr-defs.h, 300
- SPR_DCR_CT_LSD
 - spr-defs.h, 300
- SPR_DCR_CT_LSEA
 - spr-defs.h, 300
- SPR_DCR_CT_SD
 - spr-defs.h, 300
- SPR_DCR_CT_SEA
 - spr-defs.h, 300
- SPR_DCR_DP
 - spr-defs.h, 300
- SPR_DCR_LAST
 - spr-defs.h, 300
- SPR_DCR_SC
 - spr-defs.h, 300
- spr_def, 168
 - bits, 168
 - from_spr, 168
 - name, 168
 - to_spr, 168
- spr_dmmu_group
 - spr-dump.c, 304
- SPR_DMMUCFGR
 - spr-defs.h, 300
- spr_dmmucfgr
 - spr-dump.c, 304
- SPR_DMMUCFGR_CRI
 - spr-defs.h, 300
- SPR_DMMUCFGR_HTR
 - spr-defs.h, 300
- SPR_DMMUCFGR_NAE
 - spr-defs.h, 300
- SPR_DMMUCFGR_NTS
 - spr-defs.h, 300
- SPR_DMMUCFGR_NTS_OFF
 - spr-defs.h, 300
- SPR_DMMUCFGR_NTW
 - spr-defs.h, 300
- SPR_DMMUCFGR_NTW_OFF
 - spr-defs.h, 300
- SPR_DMMUCFGR_PRI
 - spr-defs.h, 300
- SPR_DMMUCFGR_TEIRI
 - spr-defs.h, 300
- SPR_DMMUCR
 - spr-defs.h, 300
- spr_dmmucr
 - spr-dump.c, 304
- SPR_DMMUCR_P1S
 - spr-defs.h, 300
- SPR_DMMUCR_P2S
 - spr-defs.h, 300
- SPR_DMMUCR_PADDR_WIDTH
 - spr-defs.h, 300
- SPR_DMMUCR_VADDR_WIDTH
 - spr-defs.h, 300
- SPR_DMR1
 - spr-defs.h, 300
- spr_dmr1
 - spr-dump.c, 305
- SPR_DMR1_BT
 - spr-defs.h, 300
- SPR_DMR1_CW
 - spr-defs.h, 300
- SPR_DMR1_CW0
 - spr-defs.h, 300
- SPR_DMR1_CW0_AND
 - spr-defs.h, 300
- SPR_DMR1_CW0_OR
 - spr-defs.h, 300
- SPR_DMR1_CW1
 - spr-defs.h, 300
- SPR_DMR1_CW1_AND
 - spr-defs.h, 300
- SPR_DMR1_CW1_OR
 - spr-defs.h, 300
- SPR_DMR1_CW2
 - spr-defs.h, 300

- SPR_DMR1_CW2_AND
spr-defs.h, 300
- SPR_DMR1_CW2_OR
spr-defs.h, 300
- SPR_DMR1_CW3
spr-defs.h, 300
- SPR_DMR1_CW3_AND
spr-defs.h, 300
- SPR_DMR1_CW3_OR
spr-defs.h, 300
- SPR_DMR1_CW4
spr-defs.h, 300
- SPR_DMR1_CW4_AND
spr-defs.h, 300
- SPR_DMR1_CW4_OR
spr-defs.h, 300
- SPR_DMR1_CW5
spr-defs.h, 300
- SPR_DMR1_CW5_AND
spr-defs.h, 300
- SPR_DMR1_CW5_OR
spr-defs.h, 300
- SPR_DMR1_CW6
spr-defs.h, 300
- SPR_DMR1_CW6_AND
spr-defs.h, 300
- SPR_DMR1_CW6_OR
spr-defs.h, 300
- SPR_DMR1_CW7
spr-defs.h, 300
- SPR_DMR1_CW7_AND
spr-defs.h, 300
- SPR_DMR1_CW7_OR
spr-defs.h, 300
- SPR_DMR1_CW8
spr-defs.h, 300
- SPR_DMR1_CW8_AND
spr-defs.h, 300
- SPR_DMR1_CW8_OR
spr-defs.h, 300
- SPR_DMR1_CW9
spr-defs.h, 300
- SPR_DMR1_CW9_AND
spr-defs.h, 300
- SPR_DMR1_CW9_OR
spr-defs.h, 300
- SPR_DMR1_RES1
spr-defs.h, 300
- SPR_DMR1_RES2
spr-defs.h, 300
- SPR_DMR1_ST
spr-defs.h, 300
- SPR_DMR2
spr-defs.h, 300
- spr_dmr2
spr-dump.c, 305
- SPR_DMR2_AWTC
spr-defs.h, 300
- SPR_DMR2_AWTC_OFF
spr-defs.h, 300
- SPR_DMR2_WBS
spr-defs.h, 300
- SPR_DMR2_WBS_OFF
spr-defs.h, 300
- SPR_DMR2_WCE0
spr-defs.h, 300
- SPR_DMR2_WCE1
spr-defs.h, 300
- SPR_DMR2_WGB
spr-defs.h, 300
- SPR_DMR2_WGB_OFF
spr-defs.h, 300
- SPR_DRR
spr-defs.h, 300
- spr_drr
spr-dump.c, 305
- SPR_DRR_AE
spr-defs.h, 300
- SPR_DRR_BUSEE
spr-defs.h, 300
- SPR_DRR_DME
spr-defs.h, 300
- SPR_DRR_DPFE
spr-defs.h, 300
- SPR_DRR_IE
spr-defs.h, 300
- SPR_DRR_IIE
spr-defs.h, 300
- SPR_DRR_IME
spr-defs.h, 300
- SPR_DRR_IPFE
spr-defs.h, 300
- SPR_DRR_RE
spr-defs.h, 300
- SPR_DRR_RSTE
spr-defs.h, 300
- SPR_DRR_SCE
spr-defs.h, 300
- SPR_DRR_TE
spr-defs.h, 300
- SPR_DRR_TTE
spr-defs.h, 300
- SPR_DSR
spr-defs.h, 300
- spr_dsr
spr-dump.c, 305
- SPR_DSR_AE
spr-defs.h, 300

- SPR_DSR_BUSEE
 - spr-defs.h, 300
- SPR_DSR_DME
 - spr-defs.h, 300
- SPR_DSR_DPFE
 - spr-defs.h, 300
- SPR_DSR_IE
 - spr-defs.h, 300
- SPR_DSR_IIE
 - spr-defs.h, 300
- SPR_DSR_IME
 - spr-defs.h, 300
- SPR_DSR_IPFE
 - spr-defs.h, 300
- SPR_DSR_RE
 - spr-defs.h, 300
- SPR_DSR_RSTE
 - spr-defs.h, 300
- SPR_DSR_SCE
 - spr-defs.h, 300
- SPR_DSR_SSE
 - spr-defs.h, 300
- SPR_DSR_TE
 - spr-defs.h, 300
- SPR_DSR_TTE
 - spr-defs.h, 300
- spr_dtlbmr
 - spr-dump.c, 306
- SPR_DTLBMR_BASE
 - spr-defs.h, 300
- SPR_DTLBMR_CID
 - spr-defs.h, 300
- SPR_DTLBMR_LAST
 - spr-defs.h, 300
- SPR_DTLBMR_LRU
 - spr-defs.h, 300
- SPR_DTLBMR_PL1
 - spr-defs.h, 300
- SPR_DTLBMR_V
 - spr-defs.h, 300
- SPR_DTLBMR_VPN
 - spr-defs.h, 300
- spr_dtlbtr
 - spr-dump.c, 306
- SPR_DTLBTR_A
 - spr-defs.h, 300
- SPR_DTLBTR_BASE
 - spr-defs.h, 300
- SPR_DTLBTR_CC
 - spr-defs.h, 300
- SPR_DTLBTR_CI
 - spr-defs.h, 300
- SPR_DTLBTR_D
 - spr-defs.h, 300
- SPR_DTLBTR_LAST
 - spr-defs.h, 300
- SPR_DTLBTR_PPN
 - spr-defs.h, 300
- SPR_DTLBTR_SRE
 - spr-defs.h, 300
- SPR_DTLBTR_SWE
 - spr-defs.h, 300
- SPR_DTLBTR_URE
 - spr-defs.h, 300
- SPR_DTLBTR_UWE
 - spr-defs.h, 300
- SPR_DTLBTR_WBC
 - spr-defs.h, 300
- SPR_DTLBTR_WOM
 - spr-defs.h, 300
- SPR_DVR
 - spr-defs.h, 300
- spr_dwcr
 - spr-dump.c, 306
- SPR_DWCRO
 - spr-defs.h, 300
- SPR_DWCRI
 - spr-defs.h, 300
- SPR_DWCR_COUNT
 - spr-defs.h, 300
- SPR_DWCR_MATCH
 - spr-defs.h, 300
- SPR_DWCR_MATCH_OFF
 - spr-defs.h, 300
- SPR_EEAR_BASE
 - spr-defs.h, 300
- SPR_EEAR_LAST
 - spr-defs.h, 300
- SPR_EPCR_BASE
 - spr-defs.h, 300
- SPR_EPCR_LAST
 - spr-defs.h, 300
- SPR_ESR_BASE
 - spr-defs.h, 300
- SPR_ESR_LAST
 - spr-defs.h, 300
- spr_groups
 - spr-dump.c, 306
- spr_ic_group
 - spr-dump.c, 307
- SPR_ICBIR
 - spr-defs.h, 300
- SPR_ICBLR
 - spr-defs.h, 300
- SPR_ICBPR
 - spr-defs.h, 300
- SPR_ICCFGR
 - spr-defs.h, 300

- spr_iccfgr
 - spr-dump.c, 307
- SPR_ICCFGR_CBIRI
 - spr-defs.h, 300
- SPR_ICCFGR_CBLRI
 - spr-defs.h, 300
- SPR_ICCFGR_CBPRI
 - spr-defs.h, 300
- SPR_ICCFGR_CBS
 - spr-defs.h, 300
- SPR_ICCFGR_CBS_OFF
 - spr-defs.h, 300
- SPR_ICCFGR_CCRI
 - spr-defs.h, 300
- SPR_ICCFGR_NCS
 - spr-defs.h, 300
- SPR_ICCFGR_NCS_OFF
 - spr-defs.h, 300
- SPR_ICCFGR_NCW
 - spr-defs.h, 300
- SPR_ICCFGR_NCW_OFF
 - spr-defs.h, 300
- SPR_ICCR
 - spr-defs.h, 300
- spr_iccr
 - spr-dump.c, 307
- SPR_ICCR_EW
 - spr-defs.h, 300
- SPR_ICR_BASE
 - spr-defs.h, 300
- SPR_ICR_LAST
 - spr-defs.h, 300
- spr_immu_group
 - spr-dump.c, 307
- SPR_IMMUCFGR
 - spr-defs.h, 300
- spr_immucfgr
 - spr-dump.c, 308
- SPR_IMMUCFGR_CRI
 - spr-defs.h, 300
- SPR_IMMUCFGR_HTR
 - spr-defs.h, 300
- SPR_IMMUCFGR_NAE
 - spr-defs.h, 300
- SPR_IMMUCFGR_NTS
 - spr-defs.h, 300
- SPR_IMMUCFGR_NTS_OFF
 - spr-defs.h, 300
- SPR_IMMUCFGR_NTW
 - spr-defs.h, 300
- SPR_IMMUCFGR_NTW_OFF
 - spr-defs.h, 300
- SPR_IMMUCFGR_PRI
 - spr-defs.h, 300
- SPR_IMMUCFGR_TEIRI
 - spr-defs.h, 300
- SPR_IMMUCR
 - spr-defs.h, 300
- spr_immucr
 - spr-dump.c, 308
- SPR_IMMUCR_P1S
 - spr-defs.h, 300
- SPR_IMMUCR_P2S
 - spr-defs.h, 300
- SPR_IMMUCR_PADDR_WIDTH
 - spr-defs.h, 300
- SPR_IMMUCR_VADDR_WIDTH
 - spr-defs.h, 300
- spr_itlbmr
 - spr-dump.c, 308
- SPR_ITLBMR_BASE
 - spr-defs.h, 300
- SPR_ITLBMR_CID
 - spr-defs.h, 300
- SPR_ITLBMR_LAST
 - spr-defs.h, 300
- SPR_ITLBMR_LRU
 - spr-defs.h, 300
- SPR_ITLBMR_PL1
 - spr-defs.h, 300
- SPR_ITLBMR_V
 - spr-defs.h, 300
- SPR_ITLBMR_VPN
 - spr-defs.h, 300
- spr_itlbr
 - spr-dump.c, 308
- SPR_ITLBTR_A
 - spr-defs.h, 300
- SPR_ITLBTR_BASE
 - spr-defs.h, 300
- SPR_ITLBTR_CC
 - spr-defs.h, 300
- SPR_ITLBTR_CI
 - spr-defs.h, 300
- SPR_ITLBTR_D
 - spr-defs.h, 300
- SPR_ITLBTR_LAST
 - spr-defs.h, 300
- SPR_ITLBTR_PPN
 - spr-defs.h, 300
- SPR_ITLBTR_SXE
 - spr-defs.h, 300
- SPR_ITLBTR_UXE
 - spr-defs.h, 300
- SPR_ITLBTR_WBC
 - spr-defs.h, 300
- SPR_ITLBTR_WOM
 - spr-defs.h, 300

- spr_mac_group
 - spr-dump.c, 309
- SPR_MACHI
 - spr-defs.h, 300
- SPR_MACLO
 - spr-defs.h, 300
- SPR_NPC
 - spr-defs.h, 300
- spr_one_val
 - spr-dump.c, 309
- spr_pc_group
 - spr-dump.c, 309
- SPR_PCCFGR
 - spr-defs.h, 300
- SPR_PCCR
 - spr-defs.h, 300
- SPR_PCMR
 - spr-defs.h, 300
- spr_pcmr
 - spr-dump.c, 309
- SPR_PCMR_BS
 - spr-defs.h, 300
- SPR_PCMR_CISM
 - spr-defs.h, 300
- SPR_PCMR_CIUM
 - spr-defs.h, 300
- SPR_PCMR_CP
 - spr-defs.h, 300
- SPR_PCMR_DCM
 - spr-defs.h, 300
- SPR_PCMR_DDS
 - spr-defs.h, 300
- SPR_PCMR_DTLBM
 - spr-defs.h, 300
- SPR_PCMR_ICM
 - spr-defs.h, 300
- SPR_PCMR_IF
 - spr-defs.h, 300
- SPR_PCMR_IFS
 - spr-defs.h, 300
- SPR_PCMR_ITLBM
 - spr-defs.h, 300
- SPR_PCMR_LA
 - spr-defs.h, 300
- SPR_PCMR_LSUS
 - spr-defs.h, 300
- SPR_PCMR_SA
 - spr-defs.h, 300
- SPR_PCMR_UMRA
 - spr-defs.h, 300
- SPR_PCMR_WPE
 - spr-defs.h, 300
- spr_pic_group
 - spr-dump.c, 310
- SPR_PICMR
 - spr-defs.h, 300
- spr_picmr
 - spr-dump.c, 310
- SPR_PICMR_IUM
 - spr-defs.h, 300
- SPR_PICPR
 - spr-defs.h, 300
- SPR_PICPR_IPRIO
 - spr-defs.h, 300
- SPR_PICSR
 - spr-defs.h, 300
- SPR_PICSR_IS
 - spr-defs.h, 300
- spr_pm_group
 - spr-dump.c, 310
- SPR_PMR
 - spr-defs.h, 300
- spr_pmr
 - spr-dump.c, 310
- SPR_PMR_DCGE
 - spr-defs.h, 300
- SPR_PMR_DME
 - spr-defs.h, 300
- SPR_PMR_SDF
 - spr-defs.h, 300
- SPR_PMR_SME
 - spr-defs.h, 300
- SPR_PMR_SUME
 - spr-defs.h, 300
- SPR_PPC
 - spr-defs.h, 300
- spr_read_ttcrr
 - tick.c, 814
 - tick.h, 817
- SPR_SR
 - spr-defs.h, 300
- spr_sr
 - spr-dump.c, 310
- SPR_SR_CE
 - spr-defs.h, 300
- SPR_SR_CID
 - spr-defs.h, 300
- SPR_SR_CY
 - spr-defs.h, 300
- SPR_SR_DCE
 - spr-defs.h, 300
- SPR_SR_DME
 - spr-defs.h, 300
- SPR_SR_DSX
 - spr-defs.h, 300
- SPR_SR_EPH
 - spr-defs.h, 300
- SPR_SR_F

- spr-defs.h, 300
- SPR_SR_FO
 - spr-defs.h, 300
- SPR_SR_ICE
 - spr-defs.h, 300
- SPR_SR_IEE
 - spr-defs.h, 300
- SPR_SR_IME
 - spr-defs.h, 300
- SPR_SR_LEE
 - spr-defs.h, 300
- SPR_SR_OV
 - spr-defs.h, 300
- SPR_SR_OVE
 - spr-defs.h, 300
- SPR_SR_RES
 - spr-defs.h, 300
- SPR_SR_SM
 - spr-defs.h, 300
- SPR_SR_SUMRA
 - spr-defs.h, 300
- SPR_SR_TEE
 - spr-defs.h, 300
- spr_sys_group
 - spr-dump.c, 311
- spr_tt_group
 - spr-dump.c, 311
- SPR_TTCR
 - spr-defs.h, 300
- SPR_TTCR_PERIOD
 - spr-defs.h, 300
- SPR_TTMR
 - spr-defs.h, 300
- spr_ttmr
 - spr-dump.c, 311
- SPR_TTMR_CR
 - spr-defs.h, 300
- SPR_TTMR_IE
 - spr-defs.h, 300
- SPR_TTMR_IP
 - spr-defs.h, 300
- SPR_TTMR_M
 - spr-defs.h, 300
- SPR_TTMR_PERIOD
 - spr-defs.h, 300
- SPR_TTMR_RT
 - spr-defs.h, 300
- SPR_TTMR_SR
 - spr-defs.h, 300
- SPR_UPR
 - spr-defs.h, 300
- spr_upr
 - spr-dump.c, 312
- SPR_UPR_CUP
 - spr-defs.h, 300
- SPR_UPR_DCP
 - spr-defs.h, 300
- SPR_UPR_DMP
 - spr-defs.h, 300
- SPR_UPR_DUP
 - spr-defs.h, 300
- SPR_UPR_ICP
 - spr-defs.h, 300
- SPR_UPR_IMP
 - spr-defs.h, 300
- SPR_UPR_MP
 - spr-defs.h, 300
- SPR_UPR_PCUP
 - spr-defs.h, 300
- SPR_UPR_PICP
 - spr-defs.h, 300
- SPR_UPR_PMP
 - spr-defs.h, 300
- SPR_UPR_RES
 - spr-defs.h, 300
- SPR_UPR_TTP
 - spr-defs.h, 300
- SPR_UPR_UP
 - spr-defs.h, 300
- SPR_VR
 - spr-defs.h, 300
- spr_vr
 - spr-dump.c, 312
- SPR_VR_CFG
 - spr-defs.h, 300
- SPR_VR_CFG_OFF
 - spr-defs.h, 300
- SPR_VR_RES
 - spr-defs.h, 300
- SPR_VR_REV
 - spr-defs.h, 300
- SPR_VR_REV_OFF
 - spr-defs.h, 300
- SPR_VR_VER
 - spr-defs.h, 300
- SPR_VR_VER_OFF
 - spr-defs.h, 300
- spr_write_ttc
 - tick.c, 814
 - tick.h, 817
- spr_write_ttmr
 - tick.c, 815
 - tick.h, 818
- SPRGROUP_D
 - spr-defs.h, 300
- SPRGROUP_DC
 - spr-defs.h, 300
- SPRGROUP_DMMU

- spr-defs.h, 300
- SPRGROUP_IC
 - spr-defs.h, 300
- SPRGROUP_IMMU
 - spr-defs.h, 300
- SPRGROUP_MAC
 - spr-defs.h, 300
- SPRGROUP_PC
 - spr-defs.h, 300
- SPRGROUP_PIC
 - spr-defs.h, 300
- SPRGROUP_PM
 - spr-defs.h, 300
- SPRGROUP_SYS
 - spr-defs.h, 300
- SPRGROUP_TT
 - spr-defs.h, 300
- sprs
 - cpu_state, 57
- sprs.c
 - audio_cnt, 316
 - DECLARE_DEBUG_CHANNEL, 315
 - DEFAULT_DEBUG_CHANNEL, 315
 - fo, 316
 - mfspr, 315
 - mtspr, 315
 - sprs_status, 315
- sprs.h
 - dump_spr, 317
 - mfspr, 317
 - mtspr, 318
 - sprs_status, 318
- sprs_status
 - sprs.c, 315
 - sprs.h, 318
- sstats
 - stats.c, 276
- sstats_entry, 169
 - cnt_dynamic, 169
 - insn, 169
- SSTATS_LEN
 - stats.c, 275
- st_info
 - elf32_sym, 95
 - elf64_sym, 104
- st_name
 - elf32_sym, 95
 - elf64_sym, 104
- st_other
 - elf32_sym, 95
 - elf64_sym, 104
- st_shndx
 - elf32_sym, 95
 - elf64_sym, 104
- st_size
 - elf32_sym, 95
 - elf64_sym, 104
- st_value
 - elf32_sym, 95
 - elf64_sym, 104
- stack
 - profiler.c, 751
- STACK_ARGS
 - simprintf.c, 810
- STACK_SIZE
 - abstract.h, 224
- stack_struct, 170
 - addr, 170
 - cycles, 170
 - name, 170
 - raddr, 170
- stalled
 - runtime, 160
- STANDBY
 - atacmd.h, 593
- STANDBY_IMMEDIATE
 - atacmd.h, 593
- start_addr
 - _cuc_func, 14
 - mprofiler.c, 558
- stat
 - ata_host, 24
 - vga_state, 175
- stat_func
 - sim_stat, 166
- state
 - ata_device, 20
 - eth_device, 110
- stats.c
 - addstats, 275
 - addfstats, 275
 - addsstats, 275
 - dc_stats, 275
 - dmmu_stats, 275
 - dstats, 275
 - DSTATS_LEN, 275
 - fstats, 275
 - FSTATS_LEN, 275
 - func_unit_str, 276
 - ic_stats, 276
 - immu_stats, 276
 - initstats, 275
 - or1k_mstats, 276
 - printotherstats, 275
 - printstats, 275
 - RAW_RANGE, 275
 - raw_stats, 276
 - SD, 275

- sstats, 276
- SSTATS_LEN, 275
- stats.h
 - addstats, 278
 - addfstats, 278
 - addsstats, 278
 - dc_stats, 278
 - dmmu_stats, 278
 - ic_stats, 278
 - immu_stats, 278
 - initstats, 278
 - or1k_mstats, 278
 - printstats, 278
- status
 - ata_device, 20
 - channel_ops, 37
 - jtr_chain_response, 131
 - jtr_failure_response, 132
 - jtr_read_block_response, 134
 - jtr_read_response, 136
 - jtr_write_block_response, 138
 - jtr_write_response, 140
- STB_GLOBAL
 - elf.h, 253
- STB_LOCAL
 - elf.h, 253
- STB_WEAK
 - elf.h, 253
- store_hitdelay
 - config, 52
- store_missdelay
 - config, 52
- storecycles
 - runtime, 160
- stores_pagefaults
 - dmmustats_entry, 84
- stores_tlbhit
 - dmmustats_entry, 84
- stores_tlbmiss
 - dmmustats_entry, 84
- STR_SIZE
 - sim-config.h, 782
- str_val
 - param_val, 155
- stream
 - ata_device, 20
- strip_space
 - sim-cmd.c, 769
- strndup
 - port.h, 747
 - strndup.c, 748
- strndup.c
 - strndup, 748
- strstrip
 - parse.c, 269
 - parse.h, 272
- STT_FILE
 - elf.h, 253
- STT_FUNC
 - elf.h, 253
- STT_NOTYPE
 - elf.h, 253
- STT_OBJECT
 - elf.h, 253
- STT_SECTION
 - elf.h, 253
- supercycles
 - runtime, 160
- superscalar
 - config, 52
- support/dbchs.h, 788
- support/debug.c, 789
- support/debug.h, 791
- support/dumpverilog.c, 795
- support/dumpverilog.h, 798
- support/misc.c, 800
- support/misc.h, 801
- support/profile.c, 802
- support/profile.h, 803
- support/sched.c, 805
- support/sched.h, 807
- support/simprintf.c, 809
- support/simprintf.h, 811
- SUPPORT_APM
 - atadevice_cmdi.h, 611
- SUPPORT_CFA
 - atadevice_cmdi.h, 611
- SUPPORT_DEVICE_RESET_CMD
 - atadevice_cmdi.h, 611
- SUPPORT_DOWNLOAD_MICROCODE
 - atadevice_cmdi.h, 611
- SUPPORT_HOST_PROTECTED_AREA
 - atadevice_cmdi.h, 611
- SUPPORT_LOOKAHEAD
 - atadevice_cmdi.h, 611
- SUPPORT_NOP_CMD
 - atadevice_cmdi.h, 611
- SUPPORT_POWER_MANAGEMENT
 - atadevice_cmdi.h, 611
- SUPPORT_POWER_UP_IN_STANDBY_MODE
 - atadevice_cmdi.h, 611
- SUPPORT_READ_BUFFER_CMD
 - atadevice_cmdi.h, 611
- SUPPORT_READ_WRITE_DMA_QUEUED
 - atadevice_cmdi.h, 611
- SUPPORT_RELEASE_INTERRUPT
 - atadevice_cmdi.h, 611
- SUPPORT_REMOVABLE_MEDIA

- atadevice_cmdi.h, 611
- SUPPORT_REMOVABLE_MEDIA_NOTIFICATION
 - atadevice_cmdi.h, 611
- SUPPORT_SECURITY_MODE
 - atadevice_cmdi.h, 611
- SUPPORT_SERVICE_INTERRUPT
 - atadevice_cmdi.h, 611
- SUPPORT_SET_MAX
 - atadevice_cmdi.h, 611
- SUPPORT_SMART
 - atadevice_cmdi.h, 611
- SUPPORT_WRITE_BUFFER_CMD
 - atadevice_cmdi.h, 611
- SUPPORT_WRITE_CACHE
 - atadevice_cmdi.h, 611
- SWAP_ENDIAN_LONG
 - coeff.h, 243
- SWAP_ENDIAN_SHORT
 - coeff.h, 243
- switch_param
 - sim-config.c, 780
- swptr
 - dma_channel, 78
- sz
 - dma_channel, 78
- T
 - op_1t.h, 402
 - op_2t.h, 404
 - op_3t.h, 406
- T0_REG
 - i386_regs.h, 373
- T1_REG
 - i386_regs.h, 373
- T2_REG
 - i386_regs.h, 373
- T_NONE
 - dyn_rec.c, 330
- tagaddr
 - dc_set, 67
- tagaddr_mask
 - ic, 124
- tags
 - ic, 124
- taken
 - bpb_entry, 28
 - branchstat, 30
- tcp.c
 - tcp_channel_ops, 640
 - tcp_init, 640
 - tcp_open, 640
 - tcp_read, 640
 - tcp_write, 640
- wait_for_tcp_connect, 640
- tcp.h
 - tcp_channel_ops, 641
- tcp_channel, 171
 - connected, 171
 - fds, 171
 - nonblocking, 171
 - port_number, 171
 - socket_fd, 171
- tcp_channel_ops
 - tcp.c, 640
 - tcp.h, 641
- tcp_init
 - tcp.c, 640
- tcp_level
 - gdbcomm.c, 529
 - vapi.c, 844
- tcp_open
 - tcp.c, 640
- tcp_read
 - tcp.c, 640
- tcp_write
 - tcp.c, 640
- temp
 - vapi_handler, 173
- TEST_FLAG
 - fields.h, 692
- text_start
 - COFF_AOUTHDR, 38
- TFLAG_DST
 - dyn_rec.c, 330
- TFLAG_SAVED
 - dyn_rec.c, 330
- TFLAG_SOURCED
 - dyn_rec.c, 330
- TFLAG_SRC
 - dyn_rec.c, 330
- tflags
 - op_queue, 154
- ti
 - or32.c, 432
- tick.c
 - cycles_start, 816
 - DEFAULT_DEBUG_CHANNEL, 814
 - sched_timer_job, 814
 - spr_read_ttc, 814
 - spr_write_ttc, 814
 - spr_write_ttmr, 815
 - tick_count, 816
 - tick_one_shot, 815
 - tick_raise_except, 815
 - tick_reset, 816
 - tick_restart, 816
- tick.h

- spr_read_tcr, 817
- spr_write_tcr, 817
- spr_write_ttmr, 818
- tick_reset, 818
- tick/tick.c, 813
- tick/tick.h, 817
- tick_count
 - tick.c, 816
- tick_one_shot
 - tick.c, 815
- tick_raise_except
 - tick.c, 815
- tick_reset
 - tick.c, 816
 - tick.h, 818
- tick_restart
 - tick.c, 816
- tim
 - cuc_bb, 60
- tim_comp
 - cuc.c, 458
- time
 - sched_entry, 162
- time_point
 - runtime, 160
- timing_table
 - timings.c, 505
- timings
 - _cuc_func, 14
- timings.c
 - analyse_timings, 504
 - bb_size, 504
 - cut_tree, 504
 - ii_size, 504
 - insn_size, 505
 - insn_time, 505
 - load_timing_table, 505
 - mark_cut, 505
 - max_bb_delay, 505
 - max_delay, 505
 - memory_delay, 505
 - new_bb_cycles, 505
 - recalc_cnts, 505
 - timing_table, 505
- timings_fn
 - config, 52
- tmp
 - _cuc_func, 14
 - cuc_bb, 60
 - cuc_insn, 62
- tmp_op
 - insn.c, 483
- tmp_opt
 - insn.c, 483
- tms
 - mc, 144
- to
 - cuc_conv, 61
- to_insn_num
 - sim-cmd.c, 770
- to_spr
 - spr_def, 168
- toplevel-mprofile.c, 819
 - main, 821
- toplevel-profile.c, 822
 - main, 824
- toplevel-support.c, 825
 - check_int, 826
 - ctrl_c, 827
 - reg_sim_reset, 827
 - sim_done, 828
 - sim_init, 828
 - sim_reset, 829
 - sim_reset_hooks, 830
- toplevel-support.h, 831
 - check_int, 831
 - ctrl_c, 832
 - reg_sim_reset, 832
 - sim_done, 833
 - sim_init, 833
 - sim_reset, 834
- toplevel.c, 836
 - main, 836
- total_size
 - dma_channel, 78
- TRACE
 - debug.h, 793
- trace.c
 - set_insnbrkpoint, 279
- trace.h
 - set_insnbrkpoint, 281
- TRACE_
 - debug.h, 793
- TRACE_ON
 - debug.h, 793
- trans_direction
 - dev_generic, 74
- trans_size
 - dev_generic, 74
- transl_error
 - parse.c, 270
- transl_table
 - parse.c, 270
- translate
 - parse.c, 269
- TRUE
 - gdbcomm.h, 530
- ts_bound

- dyn_page, 86
- tsize
 - COFF_AOUTHDR, 38
- tty.c
 - baud_table, 643
 - DEFAULT_BAUD, 643
 - DEFAULT_TTY_DEVICE, 643
 - name, 643
 - parse_baud, 643
 - tty_channel_ops, 643
 - tty_init, 643
 - tty_open, 643
 - value, 643
- tty.h
 - tty_channel_ops, 644
- tty_channel, 172
 - fds, 172
- tty_channel_ops
 - tty.c, 643
 - tty.h, 644
- tty_init
 - tty.c, 643
- tty_open
 - tty.c, 643
- tx
 - eth_device, 110
- tx_bd_num
 - eth_device, 110
- tx_buff
 - eth_device, 110
- tx_channel
 - eth_device, 110
- txb
 - ata_host, 24
- txbuf
 - dev_16450, 71
- txbuf_full
 - dev_16450, 71
- txbuf_head
 - dev_16450, 71
- txbuf_tail
 - dev_16450, 71
- txfd
 - eth_device, 110
- txfile
 - eth_device, 110
- txser
 - dev_16450, 71
- type
 - ata_device, 20
 - BMP_HEADER, 27
 - config_param, 55
 - cuc_bb, 60
 - cuc_insn, 62
 - mem_config, 147
 - mprofentry_struct, 151
 - reloc, 157
- TYPE_FILE
 - atadevice.h, 601
- TYPE_LOCAL
 - atadevice.h, 601
- TYPE_NO_CONNECT
 - atadevice.h, 601
- uart16550
 - dev_16450, 71
- uart_16550
 - 16450.c, 574
- uart_add_char
 - 16450.c, 574
- UART_ADDR_SPACE
 - 16450.c, 567
- uart_baseaddr
 - 16450.c, 574
- UART_BREAK_COUNT
 - 16450.c, 567
- uart_channel
 - 16450.c, 574
- uart_char_clock
 - 16450.c, 574
- UART_CHAR_TIMEOUT
 - 16450.c, 567
- uart_check_char
 - 16450.c, 575
- uart_check_rdi
 - 16450.c, 575
- uart_check_rlsi
 - 16450.c, 576
- uart_clear_int
 - 16450.c, 576
- UART_CLOCK_DIVIDER
 - 16450.c, 567
- UART_DLH
 - 16450.c, 567
- UART_DLL
 - 16450.c, 567
- uart_enabled
 - 16450.c, 576
- UART_FCR
 - 16450.c, 567
- UART_FCR_FIE
 - 16450.c, 567
- UART_FCR_RRXFI
 - 16450.c, 567
- UART_FCR_RTXFI
 - 16450.c, 567
- UART_FGETC_SLOWDOWN
 - 16450.c, 567

- UART_FIFO_TRIGGER
 - 16450.c, [567](#)
- UART_IER
 - 16450.c, [568](#)
- UART_IER_MSI
 - 16450.c, [570](#)
- UART_IER_RDI
 - 16450.c, [570](#)
- UART_IER_RLSI
 - 16450.c, [570](#)
- UART_IER_THRI
 - 16450.c, [570](#)
- UART_IIR
 - 16450.c, [570](#)
- UART_IIR_CTI
 - 16450.c, [570](#)
- UART_IIR_ID
 - 16450.c, [570](#)
- UART_IIR_MSI
 - 16450.c, [570](#)
- UART_IIR_NO_INT
 - 16450.c, [570](#)
- UART_IIR_RDI
 - 16450.c, [570](#)
- UART_IIR_RLSI
 - 16450.c, [570](#)
- UART_IIR_THRI
 - 16450.c, [570](#)
- uart_int_cti
 - 16450.c, [577](#)
- uart_int_msi
 - 16450.c, [577](#)
- uart_int_rdi
 - 16450.c, [577](#)
- uart_int_rlsi
 - 16450.c, [578](#)
- uart_int_thri
 - 16450.c, [578](#)
- uart_irq
 - 16450.c, [579](#)
- uart_jitter
 - 16450.c, [579](#)
- UART_LCR
 - 16450.c, [570](#)
- UART_LCR_DLAB
 - 16450.c, [570](#)
- UART_LCR_EPAR
 - 16450.c, [570](#)
- UART_LCR_PARITY
 - 16450.c, [570](#)
- UART_LCR_RESET
 - 16450.c, [570](#)
- UART_LCR_SBC
 - 16450.c, [570](#)
- UART_LCR_SPAR
 - 16450.c, [570](#)
- UART_LCR_STOP
 - 16450.c, [570](#)
- UART_LCR_WLEN5
 - 16450.c, [570](#)
- UART_LCR_WLEN6
 - 16450.c, [570](#)
- UART_LCR_WLEN7
 - 16450.c, [570](#)
- UART_LCR_WLEN8
 - 16450.c, [570](#)
- uart_loopback
 - 16450.c, [579](#)
- UART_LSR
 - 16450.c, [570](#)
- UART_LSR_BREAK
 - 16450.c, [570](#)
- UART_LSR_FRAME
 - 16450.c, [570](#)
- UART_LSR_OVRRUN
 - 16450.c, [570](#)
- UART_LSR_PARITY
 - 16450.c, [570](#)
- UART_LSR_RDRDY
 - 16450.c, [570](#)
- UART_LSR_RXERR
 - 16450.c, [570](#)
- UART_LSR_TXBUFE
 - 16450.c, [570](#)
- UART_LSR_TXSERE
 - 16450.c, [570](#)
- UART_MAX_FIFO_LEN
 - 16450.c, [570](#)
- UART_MCR
 - 16450.c, [570](#)
- UART_MCR_AUX1
 - 16450.c, [571](#)
- UART_MCR_AUX2
 - 16450.c, [571](#)
- UART_MCR_DTR
 - 16450.c, [571](#)
- UART_MCR_LOOP
 - 16450.c, [571](#)
- UART_MCR_RTS
 - 16450.c, [571](#)
- UART_MSR
 - 16450.c, [571](#)
- UART_MSR_CTS
 - 16450.c, [571](#)
- UART_MSR_DCD
 - 16450.c, [571](#)
- UART_MSR_DCTS
 - 16450.c, [571](#)

- UART_MSR_DDCD
 - 16450.c, [571](#)
- UART_MSR_DDSR
 - 16450.c, [571](#)
- UART_MSR_DSR
 - 16450.c, [571](#)
- UART_MSR_RI
 - 16450.c, [571](#)
- UART_MSR_TERI
 - 16450.c, [571](#)
- uart_newway
 - 16450.c, [579](#)
- uart_next_int
 - 16450.c, [580](#)
- uart_read_byte
 - 16450.c, [580](#)
- uart_recv_break
 - 16450.c, [580](#)
- uart_recv_break_start
 - 16450.c, [581](#)
- uart_recv_break_stop
 - 16450.c, [581](#)
- uart_recv_char
 - 16450.c, [581](#)
- uart_reset
 - 16450.c, [582](#)
 - 16450.h, [589](#)
- UART_RXBUF
 - 16450.c, [571](#)
- uart_sched_recv_check
 - 16450.c, [582](#)
- UART_SCR
 - 16450.c, [571](#)
- uart_sec_end
 - 16450.c, [582](#)
- uart_sec_start
 - 16450.c, [583](#)
- uart_send_break
 - 16450.c, [583](#)
- uart_status
 - 16450.c, [584](#)
 - 16450.h, [589](#)
- uart_tx_send
 - 16450.c, [584](#)
- UART_TXBUF
 - 16450.c, [571](#)
- UART_VALID_FCR
 - 16450.c, [571](#)
- UART_VALID_IER
 - 16450.c, [571](#)
- UART_VALID_IIR
 - 16450.c, [571](#)
- UART_VALID_LCR
 - 16450.c, [571](#)
- UART_VALID_LSR
 - 16450.c, [571](#)
- UART_VALID_MCR
 - 16450.c, [571](#)
- UART_VALID_MSR
 - 16450.c, [571](#)
- UART_VAPI_BUF_LEN
 - 16450.c, [571](#)
- uart_vapi_cmd
 - 16450.c, [584](#)
- uart_vapi_id
 - 16450.c, [585](#)
- uart_vapi_read
 - 16450.c, [585](#)
- uart_write_byte
 - 16450.c, [585](#)
- ULONGEST
 - abstract.h, [224](#)
- unmark_tree
 - insn.c, [483](#)
- unroll
 - cuc_timings, [66](#)
- unrolled
 - cuc_bb, [60](#)
- uorreg_t
 - arch.h, [284](#)
- upd_reg_from_t
 - op_support.h, [420](#)
- upd_sim_cycles
 - common_i386.h, [319](#)
- update_pc
 - execute.c, [368](#)
- used_regs
 - _cuc_func, [14](#)
- useless_x86
 - common_i386.h, [319](#)
- ustates
 - config, [52](#)
 - dmmu, [83](#)
 - ic, [124](#)
 - immu, [126](#)
- ustates_reload
 - ic, [124](#)
- val3232
 - common_i386.h, [319](#)
- val64
 - common_i386.h, [319](#)
- valid
 - dev_memarea, [76](#)
- value
 - dev_generic, [74](#)
 - tty.c, [643](#)
- vapi

- config, 52
- dev_16450, 71
- runtime, 160
- vapi.c
 - add_handler, 840
 - DEFAULT_DEBUG_CHANNEL, 840
 - fds, 844
 - find_handler, 840
 - get_server_socket, 840
 - handler_fits_id, 841
 - nfds, 844
 - nhandlers, 844
 - read_packet, 841
 - rebuild_fds, 841
 - reg_vapi_sec, 841
 - server_fd, 844
 - server_request, 841
 - serverIP, 844
 - tcp_level, 844
 - vapi_check, 841
 - vapi_done, 842
 - vapi_enabled, 842
 - vapi_hide_device_id, 842
 - vapi_init, 842
 - vapi_install_handler, 842
 - vapi_install_multi_handler, 843
 - vapi_log_enabled, 843
 - vapi_log_fn, 843
 - vapi_num_unconnected, 843
 - vapi_read_stream, 843
 - vapi_request, 843
 - vapi_send, 843
 - vapi_server_port, 844
 - vapi_write_log_file, 844
 - vapi_write_stream, 844
 - write_packet, 844
- vapi.h
 - reg_vapi_sec, 846
 - VAPI_COMMAND_END, 845
 - VAPI_COMMAND_REQUEST, 845
 - VAPI_COMMAND_SEND, 845
 - vapi_check, 846
 - VAPI_COMMAND, 845
 - vapi_done, 846
 - vapi_init, 846
 - vapi_install_handler, 847
 - vapi_install_multi_handler, 847
 - VAPI_MAX_DEVID, 845
 - vapi_num_unconnected, 847
 - vapi_send, 847
 - vapi_write_log_file, 847
- vapi/vapi.c, 839
- vapi/vapi.h, 845
- VAPI_COMMAND_END
 - vapi.h, 845
- VAPI_COMMAND_REQUEST
 - vapi.h, 845
- VAPI_COMMAND_SEND
 - vapi.h, 845
- vapi_buf
 - dev_16450, 71
- vapi_buf_head_ptr
 - dev_16450, 71
- vapi_buf_tail_ptr
 - dev_16450, 71
- vapi_check
 - vapi.c, 841
 - vapi.h, 846
- VAPI_COMMAND
 - vapi.h, 845
- vapi_done
 - vapi.c, 842
 - vapi.h, 846
- vapi_enabled
 - vapi.c, 842
- vapi_file
 - runtime, 160
- vapi_fn
 - config, 52
- vapi_handler, 173
 - base_id, 173
 - fd, 173
 - next, 173
 - num_ids, 173
 - priv_dat, 173
 - read_func, 173
 - temp, 173
- vapi_hide_device_id
 - vapi.c, 842
- vapi_id
 - config, 52
 - dev_16450, 71
 - dma_controller, 81
- vapi_init
 - vapi.c, 842
 - vapi.h, 846
- vapi_install_handler
 - vapi.c, 842
 - vapi.h, 847
- vapi_install_multi_handler
 - vapi.c, 843
 - vapi.h, 847
- vapi_log_enabled
 - vapi.c, 843
- vapi_log_fn
 - vapi.c, 843
- VAPI_MAX_DEVID
 - vapi.h, 845

- vapi_num_unconnected
 - vapi.c, 843
 - vapi.h, 847
- vapi_read_stream
 - vapi.c, 843
- vapi_request
 - vapi.c, 843
- vapi_send
 - vapi.c, 843
 - vapi.h, 847
- vapi_server_port
 - vapi.c, 844
- vapi_write_log_file
 - vapi.c, 844
 - vapi.h, 847
- vapi_write_stream
 - vapi.c, 844
- vbar
 - vga_state, 175
- vbindx
 - vga_state, 175
- verbose
 - config, 52
- verify_memoryarea
 - abstract.c, 221
 - abstract.h, 236
- verilog.c
 - branch_index, 507
 - find_lsc_index, 507
 - func_index, 507
 - GEN, 507
 - generate_main, 507
 - output_verilog, 507
 - print_deps, 507
 - print_insn_v, 508
 - print_op_v, 508
 - print_turn_off_dep, 508
- verilog.h
 - generate_main, 509
 - output_verilog, 509
- vga.c
 - reg_vga_sec, 732
 - VGA_ADDR_SPACE, 732
 - vga_baseaddr, 733
 - VGA_CLUTA, 732
 - VGA_CLUTB, 732
 - VGA_CTRL, 732
 - VGA_CTRL_CD, 732
 - VGA_CTRL_PC, 732
 - VGA_CTRL_VEN, 732
 - vga_dump_image, 733
 - vga_enabled, 733
 - vga_filename, 733
 - VGA_HTIM, 732
 - VGA_HVLEN, 732
 - vga_irq, 734
 - vga_job, 734
 - VGA_MASK, 732
 - vga_read32, 734
 - vga_refresh_rate, 734
 - vga_reset, 734
 - VGA_HVLEN, 732
 - vga_irq, 734
 - vga_job, 734
 - VGA_MASK, 732
 - vga_read32, 734
 - vga_refresh_rate, 734
 - vga_reset, 734
 - vga_sec_end, 734
 - vga_sec_start, 735
 - VGA_STAT, 732
 - VGA_VBARA, 732
 - VGA_VBARB, 732
 - VGA_VTIM, 732
 - vga_write32, 735
- vga.h
 - reg_vga_sec, 736
- VGA_ADDR_SPACE
 - vga.c, 732
- vga_baseaddr
 - vga.c, 733
- VGA_CLUTA
 - vga.c, 732
- VGA_CLUTB
 - vga.c, 732
- VGA_CTRL
 - vga.c, 732
- VGA_CTRL_CD
 - vga.c, 732
- VGA_CTRL_PC
 - vga.c, 732
- VGA_CTRL_VEN
 - vga.c, 732
- vga_dump_image
 - vga.c, 733
- vga_enabled
 - vga.c, 733
- vga_filename
 - vga.c, 733
- VGA_HTIM
 - vga.c, 732
- VGA_HVLEN
 - vga.c, 732
- vga_irq
 - vga.c, 734
- vga_job
 - vga.c, 734
- VGA_MASK
 - vga.c, 732
- vga_read32
 - vga.c, 734
- vga_refresh_rate
 - vga.c, 734
- vga_reset
 - vga.c, 734

- vga_sec_end
 - vga.c, 734
- vga_sec_start
 - vga.c, 735
- VGA_STAT
 - vga.c, 732
- vga_state, 174
 - baseaddr, 175
 - ctrl, 175
 - enabled, 175
 - filename, 175
 - hlen, 175
 - htim, 175
 - irq, 175
 - palette, 175
 - pics, 175
 - pindex, 175
 - refresh_rate, 175
 - stat, 175
 - vbar, 175
 - vbindx, 175
 - vlen, 175
 - vtim, 175
- VGA_VBARA
 - vga.c, 732
- VGA_VBARB
 - vga.c, 732
- VGA_VTIM
 - vga.c, 732
- vga_write32
 - vga.c, 735
- vlen
 - vga_state, 175
- vpn_mask
 - dmmu, 83
 - immu, 126
- vstamp
 - COFF_AOUTHDR, 38
- vtim
 - vga_state, 175
- wait_for_tcp_connect
 - tcp.c, 640
- waiting_for_dma
 - eth_device, 110
- WARN
 - debug.h, 793
- WARN_
 - debug.h, 793
- WARN_ON
 - debug.h, 793
- WARNING
 - cpu-config.c, 201
 - sim-config.c, 774
- way
 - bpb_entry, 28
 - btic_entry, 32
 - dc_set, 67
- wfds
 - eth_device, 110
- width
 - INFOHEADER, 128
- word_enabled
 - dev_generic, 74
- words_transferred
 - dma_channel, 78
- working
 - eth_device, 110
- write
 - channel_ops, 37
- WRITE_BUFFER
 - atacmd.h, 593
- write_dat16
 - mem_ops, 149
- write_dat32
 - mem_ops, 149
- write_dat8
 - mem_ops, 149
- WRITE_DMA
 - atacmd.h, 593
- WRITE_DMA_QUEUED
 - atacmd.h, 593
- WRITE_MULTIPLE
 - atacmd.h, 593
- write_packet
 - vapi.c, 844
- WRITE_SECTOR
 - atacmd.h, 593
- WRITE_SECTORS
 - atacmd.h, 593
- write_to_reg
 - generate.c, 372
- write_up
 - config, 52
- writelfunc16
 - mem_ops, 149
- writelfunc32
 - mem_ops, 149
- writelfunc8
 - mem_ops, 149
- writelit
 - cachestats_entry, 34
- writemiss
 - cachestats_entry, 34
- writeprog32
 - mem_ops, 149
- writeprog32_dat
 - mem_ops, 149

- writeprog8
 - mem_ops, 149
- writeprog8_dat
 - mem_ops, 149
- x_ary
 - COFF_auxent, 41
- x_dimen
 - COFF_auxent, 41
- x_endndx
 - COFF_auxent, 41
- x_fcn
 - COFF_auxent, 41
- x_fcnary
 - COFF_auxent, 41
- x_file
 - COFF_auxent, 41
- x_fname
 - COFF_auxent, 41
- x_fsize
 - COFF_auxent, 41
- x_inno
 - COFF_auxent, 41
- x_innoptr
 - COFF_auxent, 41
- x_insz
 - COFF_auxent, 41
- x_misc
 - COFF_auxent, 41
- x_n
 - COFF_auxent, 41
- x_nlinno
 - COFF_auxent, 41
- x_nreloc
 - COFF_auxent, 41
- x_offset
 - COFF_auxent, 41
- x_scn
 - COFF_auxent, 41
- x_scnlen
 - COFF_auxent, 41
- x_size
 - COFF_auxent, 41
- x_sym
 - COFF_auxent, 41
- x_tagndx
 - COFF_auxent, 41
- x_tv
 - COFF_auxent, 41
- x_tvfill
 - COFF_auxent, 41
- x_tvlen
 - COFF_auxent, 41
- x_tvndx
 - COFF_auxent, 41
- x_tvran
 - COFF_auxent, 41
- x_zeroes
 - COFF_auxent, 41
- xchg_insn
 - load.c, 493
- xglue
 - dyn_rec.h, 348
- xref
 - op_queue, 154
- xresolution
 - INFOHEADER, 128
- xterm.c
 - basename, 646
 - MAX_XTERM_ARGS, 646
 - xterm_channel_ops, 646
 - xterm_close, 646
 - xterm_init, 646
 - xterm_open, 646
- xterm.h
 - xterm_channel_ops, 647
- xterm_channel, 176
 - argv, 176
 - fds, 176
 - pid, 176
- xterm_channel_ops
 - xterm.c, 646
 - xterm.h, 647
- xterm_close
 - xterm.c, 646
- xterm_init
 - xterm.c, 646
- xterm_open
 - xterm.c, 646
- yresolution
 - INFOHEADER, 128